2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovations

Security OSIF: Toward Automatic Discovery and Analysis of Event Based Cyber Threat Intelligence

Ke Li[1,2], Hui Wen[1,*], Hong Li[1], Hongsong Zhu[1], Limin Sun[1]

[1]*Beijing Key Laboratory of IOT Information Security Technology, Institute of Information Engineering, CAS, Beijing, China*
[2]*University of Chinese Academy of Sciences, Beijing, China*

***Abstract*—To adapt to the rapidly evolving landscape of cyber threats, efficient collection and analysis of cyber threat intelligence (CTI) is crucial for safety staff to implement a proactive cyber defense, such as security hardening or incident responding. However, with the exponential increase in open source information, cyber threat intelligence becomes increasing hard to gather from wild open source by human efforts. Furthermore, automatically determining cyber intelligent information with respect to relevant threats reported or newsletter remains a challenge, largely due to the lack of corresponding principles or rules to analyze semantics and contextual information that present in textual representations. To overcome these limitation, this paper propose a security open source intelligence framework (OSIF) to automatically analyze unstructured text for generating event based cyber threat intelligent. It uses several technologies such as natural language process, machine learning and data mining to extract cybersecurity event related information (device, organization, location, etc.) and Common Vulnerabilities and Exposure (CVE) for threat actor profiling. Finally, we perform a comprehensive structural and conceptual evaluation of critical threats on dataset that collected from dozens of websites. And the experiments that conducted on the dataset demonstrate that our approach have a considerable performance.**

***Keywords*-cyber threat intelligent; text analytics; nature language process; information extraction;**

I. Introduction

Cyber Threat Intelligence (CTI) is considered as evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subject response to that menace or hazard [1]. It enables the organizations to gather valuable insights based on the analysis of contextual and situational risks by understanding the threats against cyber security of system or device. For instance, if and when a security incident occurs, CTI can be considered as event related information (device, CVE, location, etc.). It provide insight for incident-responder activities to implement a proactive cyber defense strategy, such as replacing the specific device at risk of attack or fixing the system vulnerabilities immediately under the CVE's guidance. However, with the exponential increase in open source information, it becomes hard to effective gather such information, which entails significant burdens for analyzing a large amount of data from wild open source.

Finding valuable cyber threat intelligence from open source data have a number of challenges. The main challenge is that few threat intelligence collection can be available on public source. Most of them come from outside management services from commercial providers, which make individuals or communities become hard to gather valuable information. Although some organizations like CleanMX [2] and PhishTank [3] share their threat intelligence on website, the information only includes a few classes (URL, IP, domain, IP and MD5), which is rather thin for cybersecurity analysis. Gartner [4] provides more valuable security information by exposing a large number of devices with vulnerabilities, but the information lacks of detailed description and demand-oriented information management for proactive cyber defense work. Traditional approaches that deploy honeyclients or sandboxes to collect cyber-attack data can gather much cyber threat intelligence [5] [6], but they need to pay extra human effort to process these unlabeled data. Other approaches based on web crawlers [7] [8] or user reports provide a proactive way to auto-detect security related information from open source, but they are not able to extract specific semantic information. An intuitive way to solve this problem is to gather security related information from web newsletters or reports to construct CTI.

Gathering and filtering security related information is the key process of the CTI discovery. In many cases, we hardly can extract unique CTI from newsletters or reports on internet, even we known there are some threat intelligence in these text representation. For instance, safety staff need to collect and track security news or reports daily for keeping the safety of system, which is high labor intensity and low effective. Furthermore, reports or newsletters are produced at a high volume and velocity on internet, which becomes increasing hard to extract threat intelligence in

* Corresponding author.

978-1-5386-9380-3/18/$31.00 ©2018 IEEE
DOI 10.1109/SmartWorld.2018.00142

these text representation. To address these problem, information selection [9] [10] and refining is necessary for handling these threat intelligence extraction issues. To the best of our knowledge, information selection problem can be solved by training a model to classify useful data, which is often used in text analysis. For information refining issues, document summarization technology [11] [12] [13] is a proper solution. It can generate informative and readable summarization for safety engineers to manage these security related information, which makes textual threat intelligence tracking and gathering effective.

Transforming information into intelligence is the key point of CTI analysis to understand security related information. Simple approach manually extracts intuitive threat intelligence from structure data, like filename or ip address, which are the basic elements of cyber threat intelligence that often presented in traffic data or security report. Once collected, these information can be automatically transformed and fed into various mechanisms [14] [15] [16] when they are formatted in accordance with a threat information standard, such as OpenIOC or Structured Threat Information eXpression (STIX) [17]. However, these threat information standards do not work well while extracting security intelligence from web text. Because the description from web text are typically informal, in natural languages, and need to be analyzed semantically before they can be converted into the threat information standard. Specifically, CTI analysis need to identify security related semantic elements in these text, such as cyber event related information (device, manufacturer, location, etc.) or Common Vulnerabilities and Exposure (CVE) [18]. To serve this purpose, defining a security event template with NLP technology is a proper way. Recent researches in NLP field propose an effective method to extract semantic elements in web text by using event template with NER (Name Entity Recognition) [19] and event trigger [20]. For instance, literature [21] presents a scalable system for real-time, automatic industry-related event extraction from social media. It detects and disambiguates highly ambiguous domain-relevant entities, such as street names, and extracts various events with their geo-locations. Other similar works [22] [23] also use NLP technology to extract semantic entity for their work in security.

Our work starts from a observation that automatically gathering security related information from web text can provide detailed threat intelligence for safety staff making better security defense strategy. And semantic analysis based threat intelligence extraction method can work in high effective and low labor cost. However, there are several problems mentioned above should be considered. To overcome these problems, our method has the following contribution:

- We propose a security open source intelligence framework (OSIF) to automatic analyze unstructured text for generating event based cyber threat intelligent.
- We present an event driven strategy for finding security related information in wild open data source and build a security event dataset for evaluation.
- We propose a data filtering and text summary method for gathering and selecting security related information for better manage these implicit threat intelligence.
- We define CTI template and present a semantic analysis based method for generating compatible, semantic rich intelligence, which addresses the emerging challenge in the effective analysis of massive open-source data.

The rest of the paper is organized as follows. Section II presents our security open source intelligence framework and related modules. In Section III, the details of security threat intelligence extraction method was established. Section IV presents the performance evaluation results. Finally we make some concluding remarks in Section V.

II. Algorithm Design and Architecture

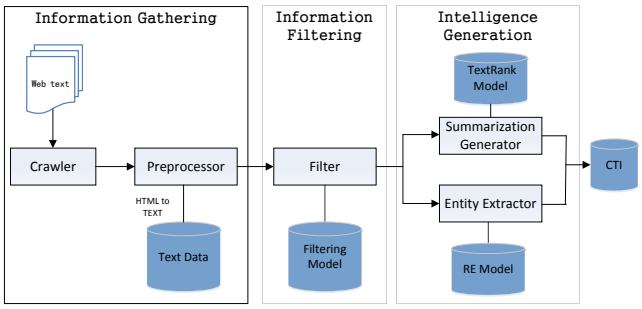

Figure 1. Security Open Source Intelligence Framework (OSIF)

In this section, we present the framework to construct CTI which contains cyber security terms (victimized devices, manufacturers and impacted locations, etc), CVE, and summarizations of articles.

As we know, *Entity Extraction* and *Automatic Summarization* can used to extract terms and generate summarization respectively. However, it is hard to extract information from unstructured natural language text, especially when high accuracy is required. In addition, the generic information extraction techniques cannot be directly applied due to the unique features of the issues. To solve the problem, we observe that the common terms are presented in a predictable way within such articles. Based on the observation, we construct a set of regexes for locating sentences likely containing technical terms. Then, the terms are extracted by using *Named Entity Recognition* tools and manually built regular expression rules. The summarizations of articles are generated with utilization of the *TextRank*. Below we introduce the proposed framework.

Architecture. Figure 1 illustrates the architecture of CTI generation. The architecture includes *Information Gathering*,

Information Filtering and *Intelligence Generation*. The Information Gathering contains a *crawler* and a *preprocessor*. The crawler is designed to gather cybersecurity events related texts from credible cybersecurity websites. The preprocessor then converts the collected texts into human readable articles and removes the irrelevant information (e.g., advertisements and website logos). Although such websites are selected carefully, there are also a few of collected articles likely containing no CTI information. Therefore, a filter is trained to inspect filter out articles without CTI information based on NLP techniques and a support vector machine. The Intelligence Generation contains a *Entity Extractor* and a *Summarization Generator*. The entity extractor breaks each article into sentences, and uses the specifically built regexes to locate sentences likely containing the entities. Then the entities in each located sentences are extracted by using NLP tools. The summarization generator utilizes the TextRank to generate the summarization of each article. Finally, the summarization and entities are sorted by the timestamps of articles to generate the CTI about security event chains.

III. Detailed Algorithm Implementation

In this section, we present the details of modules including information gathering, information filtering and intelligence generation. Each modules contains several processes.

A. Information Gathering and Filtering

In order to achieve automatic security information extraction from unstructured natural language text, the first step is to scrape the cybersecurity events related articles from the Internet and preprocess the content of the collected articles. In this subsection, we introduce the crawler, preprocessor and filter.

Crawler. The crawler is designed to collect cyber events related articles, and the target websites of the crawler are selected manually. When the target cyber events or key words are given, the crawler will leverage the search application programming interface (API) of those websites to determine and collect the corresponding articles.

Preprocessor. The preprocessor firstly converts the HTML formatted data into human-readable articles. But it is observed that there are a few extra content in the articles, such as the introduction to authors, links to correlative articles and even advertisements. If those content is not removed, it will affect the performance of the filter, even the following information extraction. In order to achieve the of content filtering, some specific nodes in DOM tree of HTML are removed when converting the HTML into text.

Filter. Once the articles are scraped and preprocessed, it is inspected by the filter whether the content of those articles contains CTI information. Although the crawler is designed to be directional as mentioned early, there are a few of articles whose content is not tightly related to cybersecurity events. Such articles seem to offer few information valuable to cybersecurity and therefore need to be filtered out. To filter out those articles, the filter trains a classifier based on natural language feature.

The feature is described as follows. In each article containing CTI information, the theme focusing on the cyber security is reflected by the key words. It is intuitive that the distribution of key words in articles containing CTI information is different from those in articles without CTI information. And this feature is leveraged to separate the articles. In the NLP community, the *term frequency–inverse document frequency* (*TF–IDF*) is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus [24]. With utilization of TF-IDF, we extract the top 10 words in each article, together with the corresponding score. We firstly construct a set of key words by combining top 10 terms of all articles, and then compute each TF-IDF score of key words for each article. The vector of TF-IDF scores is used as the feature for classification.

In order to train the classification model, we manually label 131 articles (containing 72 articles with CTI information and 49 without CTI) and separate those articles into the training set and the test set. Then using the above feature, a SVM is trained on the training set and evaluated on the test set.

B. Intelligence Generation

Although the *Preprocessor* and *Filter* can select articles which are tightly related to cyber security events, it is challenging for safety staff to read all articles and extract valuable information artificially. In order to make the CTI more concise, valuable and readable, we use some NLP technologies to extract entities and generate summarization of articles. The entities focus on the CVEs, victimized devices, device manufacturers and location distribution, which are related to each cyber security event. Those information make it possible for safety staff to promptly inspect and protect their devices. In addition, the summarizations generated from articles clearly reveal the development of cyber events. Below we firstly introduce the CTI template, and then present how the entity extractor and summarization generator work.

CTI Template. In order to concentrate on the threat intelligence related to semantic elements and content, we define the generic CTI template for each article. The CTI template contains 2 parts of threat intelligence, including threat intelligence related entities and summarizations of articles. The entities consist of the CVEs, victimized devices, device manufacturers and impacted locations. By analyzing the detailed information about the CVEs, security researchers could propose approaches to prevent cyber attacks. Meanwhile, correlation analysis of CVEs could help to find the changes of the exploited vulnerability, even speculation on the adjustment of attackers' strategies. The victimized devices and device manufacturers indicate the vulnerability

of such devices, and the administrators could be reminded to harden the security of responding devices. To find the target of the attacks, the distribution of impacted locations need to be presented. However, there are many other valuable information contained in such articles and the entities are not sufficient for constructing the CTI. It is known that the summarization can help the safety staff to comprehensively understand the core content of the articles and extract more complicated information. Therefore, it is significant to extract summarization of each article for CTI generation.

Entity Extractor. It is obviously that regular expression can be used to extract CVEs accurately since their formats are similar. But it is challenging to extract victimized devices, device manufacturers and locations from sentences. As mentioned, the problem of identifying entities in CTI template is the NER in the NLP community. There are many previous works focusing on the NER and many mature tools have been created, such as *Stanford CORENLP*. However, it is insufficient to directly apply the existing NLP techniques for NER in this work since that the NER techniques are highly domain-specific — the method designed for one domain can not work well in the other domain generally. In order to identify the entities accurately, we combine a set of regular expression (regex) and existing NLP tools. The procedure of identifying the entities is described as follows.

Each article is firstly broken into sentences and we use manually designed regexes to locate sentences which contain the entities. It is observed that the sentences containing entities may have a similar grammatical structure. For example, the sentence containing impacted locations could have the anchor "be located at" and the context of the anchor have similar parts-of-speech (pos) tagger and dependency structure. We employ the *Stanford dependency parser* to get dependency structure of each sentence and pos tagger, and then use them to generate regexes manually. In addition, for the purpose of entity extraction, the located sentences must contain the entity candidates, and this rule (containing entity candidates) is added to the regexes to locate sentences.

Once sentences are located, the corresponding entities are extracted from those sentences using NLP tools. The Stanford NER is applied to recognize named entities in implementation. But true negatives may be introduced by the Stanford NER, for example, the word "Dahua" (a manufacturer of IP camera) can not be recognized as a manufacturer. To improve the accuracy, we leverage the observation that the such entities are accompanied with common terms, for example, the manufacturer "Dahua" are usually followed by "Technology" or "Company". In addition, it is found by testing the Stanford NER that the context may impact the result of the NER, namely the same word in different context may get different result of NER. To eliminate this effect, we use the recognized entities to construct a entity set. And we use the entity set to recognize more entities in the located sentences where it is hard to identify entities by using existing NER. Further, the same entities could be presented in different ways, such as "DAHUA", "Dahua Technology" or "China Dahua" which all refer to "Dahua". To perform *Entity Linking*, we remove the common terms like "Technology" and match the remaining string of entities.

Algorithm 1 Security Information Extraction for CTI

Input: a set of articles D, containing a labeled set and an unlabeled set;
Output: time-sorted CTIs of cyber events;

```
for each article d in D do
    remove the stop words;
    compute TF-IDF of each word in d;
    generate the feature of each article;
end for
Train a SVM over the labeled set and filter out irrelevant articles;
for each relevant article d do
    for each sentence s in d do
        use the pre-defined regexes to determine whether s contains entities;
        if s contains entities then
            extract entities;
        end if
    end for
end for
for each relevant article do
    generate summarization using TextRank;
end for
Sort the extracted information and generate the CTI.
```

Summarization Generator. As mentioned earlier, to clearly reveal the development of cyber security events, we need to automatically extract summarization of each article. Essentially, summarization extraction is the *Automatic Summarization* problem in NLP. The solutions to the problem have been extensively studied, such as Kleinbergs HITS algorithm, Google's PageRank and TextRank. The TextRank is essentially a graph-based ranking algorithm which uses global information from entire graph to determine the importance of each vertex. The Summarization Generator utilizes the TextRank to extract the summarization of each article in implementation. To apply the graph-based ranking algorithm, the first step is to covert natural language text into a graph. For the purpose of sentence extraction, each sentence is added to the graph as a vertex. To draw edges between vertexes, the similarity between two sentences is measured as a function of context overlap. The similarity is defined as the number of common tokens between the lexical representations in two sentences, divided by a normalization factor. Based on the graphs which categorize natural language texts, we extract the most representative sentences. The basic idea is that the higher number of vertexes linking

to a vertex indicates the higher importance of the vertex in general. After scores of vertexes are computed, the entire sentences are ranked in reversed order and the most important sentences can be extracted as summarization. By sorting the summarizations of articles with the timestamps, the development of cyber security events can be revealed clearly.

IV. Evaluation

In this section, we evaluate the security OSIF on the dataset where the aritcles are scraped from cyber security websites. The experimental result demonstrates that the OSIF has a good performance on information filtering and information extraction, and the intelligence analysis brings a new sight into cyber security.

A. Dataset

Firstly, we present the dataset used in the study. The dataset consists of cyber security related articles, which are collected from several specific websites by the crawler. The websites are cyber security web portals which are manually selected to ensure the reliability, such as *http://www.freebuf.com/* and *https://www.easyaq.com/*. When target events is given, the crawler uses the search API of each portal to determine and scrap the corresponding articles. In the study, the cyber events include "Mirai", "Stuxnet", "Ghoul", "Flame", "Duqu", "WannaCry", etc. And there are 610 collected articles, as shown in table I.

Table I
Security Event Datasets

Dataset	Detailed Information	
Labeled Dataset	Articles with CTI	Articles without CTI
	49	71
Unlabeled Dataset	490	
Filtered Datset	Labeled Articles	Unlabeled Articles
	49	145

B. The Performance of Information Filtering

In order to train a SVM to filter out the irrelevant articles, the dataset is divided into two parts, a labeled set and an unlabeled set. The labeled set contains 71 articles without CTI information and 49 articles with CTI information, while the unlabeled set contains 490 articles. Then, the labeled set is divided into training set (80 articles) and test set (50 articles) randomly. The SVM is trained on the training set and the evaluation over test set shows that the trained model achieves a precision of 85% and a recall of 91%. After filtered by the trained SVM, there are 194 articles tightly related to the cyber events above. The table I shows the detailed information of the datasets.

C. The Performance of Information Extraction

The entity extractor in OSIF is designed to automatically extract entities from unstructured texts, therefore we employ the measurement of precision and recall for entity extraction to validate the performance of entity extraction. In order to measure the precision and recall, we randomly pick up dozens of articles from filtered articles and manually extract the entities in each article. Particularly, the 44 articles are inspected, and the extracted entities include 39 geo-locations, 9 CVEs, 25 types of victimized devices and 35 device manufacturers.

To compare the proposed approach with the state-of-the-art alternatives, the Stanford NER is used as baseline for the entity extraction. In this experiment, the Stanford NER is run over the 44 articles which are manually extract entities. The result of the study demonstrates that the proposed method is more effective than the Stanford NER. The proposed method achieves an average precision of 85.2% and an average recall of 76.6%, while the Stanford NER achieves an average precision of 85.2% and an average recall of 73.4%. The results is presented in the table II.

Table II
Precision and Recall Comparison of the Proposed Method

Method	Type	Location	Device	Org
Our method	Precision	76.9%	92%	85.7%
	Recall	75%	80%	75%
Stanford NER	Precision	76.9%	92%	85.7%
	Recall	66.7%	80%	73.1%

D. The Performance of Intelligence Analysis

The CTIs automatically constructed from the technological articles bring us new findings of the cyber security events and other related threat information. By analyzing the time-sorted CTIs, we gain a comprehensive sight into development of the events, open-source intelligence, even the evolution of attackers' strategies. For example, it could be found what time the events happened, and detailed information of malware analysis. Further, by analyzing the entities, we find which devices are most likely to be infected and which locations are most likely to be attacked. The intelligence analysis are elaborated as follows.

1) Devices and Locations: In order to find which devices could be infected, we looked into the distribution of victimized devices across the articles. As shown in the Figure 3, the percentages of articles containing each victimized device are displayed. And the IP camera, router and DVR are the devices which have high possibility to be intruded. There are several reasons for this phenomenon, such as the default username and passwords in those devices. Meanwhile, to summarize the impacted locations, we analyze the number of articles in which the corresponding location is mentioned.

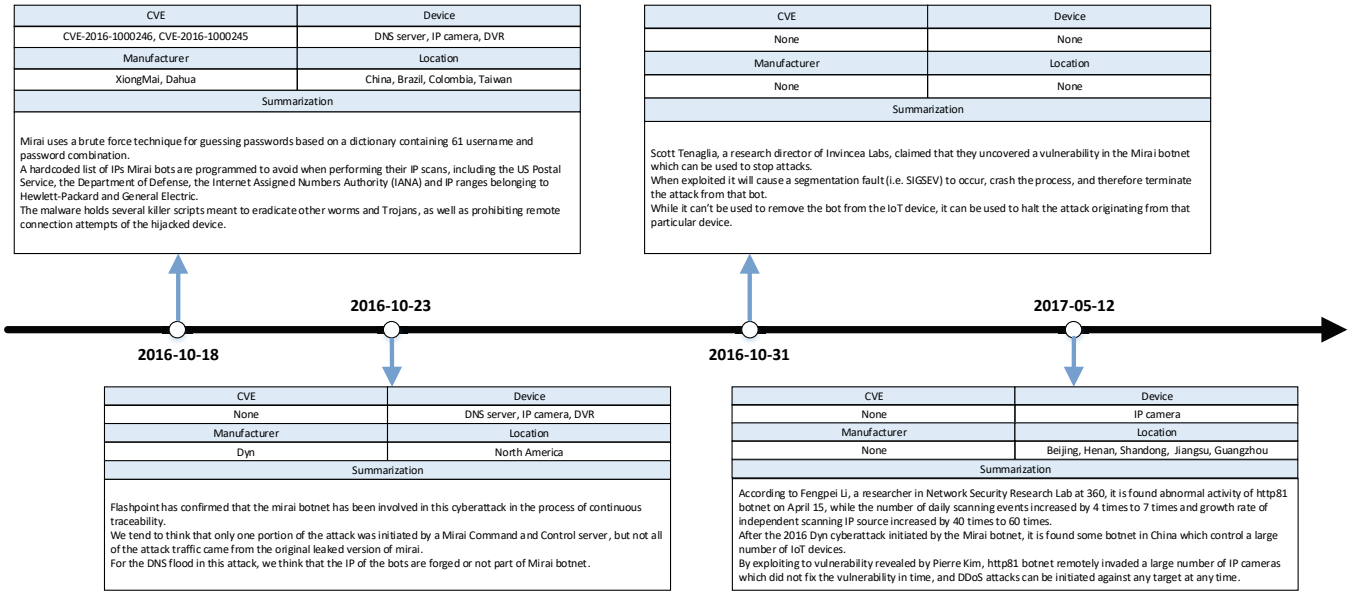

Figure 2. Threat intelligence of Mirai Event Chain

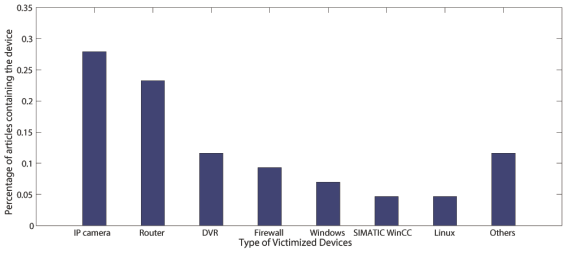

Figure 3. The distribution of Victimized Devices

The top 3 impacted countries are the United States (17 articles), Iran (8 articles) and Ukraine (7 articles).

2) Development of Events: Through analyzing the time-sorted summarization of technological articles, it presents a new insight into the development of the cyber events. For example, the figure 2 shows the partial event chain of the *Mirai*. As we know, Mirai is malware which can turn networked *Internet of Things* devices into remotely controlled bots. The Mirai botnet has been used in the most disruptive distributed denial of services (DDoS) attacks, the 2016 Dyn cyberattack happened on October 21, 2016. According to the extracted summarizations before October 20, 2016, the features of malware had been analyzed and Mirai botnet had been found. The infected devices were DNS servers, IP cameras and DVRs. And those devices were mainly located in China, Brazilian, Colombia and China Taiwan. After the 2016 Dyn cyberattack happened, some security practitioners found an interesting observation that the Mirai botnet was not the only attacker. On October 31, 2016, the *Invincea Labs* found a bug in Mirai which could be used to stop the Mirai botnet from launching DDoS attacks. On May 12, 2017, the *360 Security* found the botnet *http81* and the involved malware was similar to the Mirai.

V. Conclusion

In this paper, we propose the security OSIF, a framework for event based CTI discovery and analytics from unstructured natural language text. The proposed method is designed to gather event based articles on the Internet, filter out the collected data, and generate the pre-defined CTI. With utilization of some technologies in the NLP community, the OSIF can efficiently extract information in the well-defined CTI template. The evaluation on the dataset demonstrates that our method has good performance on data filtering and entity extractor, and the analysis about the event chain in generated CTI present some security related information. This work reveals the significance of event based cybersecurity information extraction.

VI. Acknowledgement

This work was supported by the National Key R&D Program of China (2017YFC0820701), National Natural Science Foundation of China (61702503), and the Fundamental Theory and Cutting Edge Technology Research Program of Institute of Information Engineering, CAS and SKLOIS (Y7Z0311104).

References

[1] R. Mcmillan, "Definition: Threat intelligence," 2013.

[2] M. Akiyama, T. Yagi, K. Aoki, T. Hariu, and Y. Kadobayashi, "Active credential leakage for observing web-based attack cycle," in *International Workshop on Recent Advances in Intrusion Detection*, 2013, pp. 223–243.

[3] L. M. Surhone, M. T. Timpledon, S. F. Marseken, OpenDNS, and O. (web browser), *Phishtank*, 2010.

[4] F. Caldwell, "Risk intelligence: applying km to information risk management," *Vine*, vol. 38, no. 2, pp. 163–166, 2008.

[5] M. Balduzzi, M. Balduzzi, and D. Balzarotti, "Automatic extraction of indicators of compromise for web applications," in *International Conference on World Wide Web*, 2016, pp. 333–343.

[6] Y. M. Wang, D. Beck, X. Jiang, and R. Roussev, "Automated web patrol with strider honeymonkeys," *Ndss*, 2005.

[7] D. Canali, M. Cova, G. Vigna, and C. Kruegel, "Prophiler : A fast filter for the large-scale detection of malicious web pages categories and subject descriptors," pp. 197–206, 2011.

[8] P. M. Comparetti and L. Invernizzi, "Evilseed: A guided approach to finding malicious web pages," in *IEEE Symposium on Security and Privacy*, 2012, pp. 428–442.

[9] R. C. Taylor, A. Baldwin, and C. H. Seal, "Information selection," 2007.

[10] L. Khan and F. Luo, "Ontology construction for information selection," in *IEEE International Conference on TOOLS with Artificial Intelligence*, 2002, p. 122.

[11] J. Niu, H. Chen, Q. Zhao, L. Su, and M. Atiquzzaman, "Multi-document abstractive summarization using chunk-graph and recurrent neural network."

[12] K. Filippova, "Multi-sentence compression: finding shortest paths in word graphs," in *COLING 2010, International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, 2010, pp. 322–330.

[13] S. Banerjee, P. Mitra, and K. Sugiyama, "Multi-document abstractive summarization using ilp based multi-sentence compression," in *International Conference on Artificial Intelligence*, 2015, pp. 1208–1214.

[14] E. W. Burger, M. D. Goodman, P. Kampanakis, and K. A. Zhu, "Taxonomy model for cyber threat intelligence information exchange technologies," in *ACM Workshop on Information Sharing and Collaborative Security*, 2014, pp. 51–60.

[15] J. Bortniker, "Malware analysis for cyber-threat intelligence," 2016.

[16] E. Asgarli and E. Burger, "Semantic ontologies for cyber threat sharing standards," in *Technologies for Homeland Security*, 2016, pp. 1–6.

[17] S. Barnum, "Standardizing cyber threat intelligence information with the structured threat information expression (stix)," *Mitre Corporation*, 2014.

[18] L. M. Surhone, M. T. Tennoe, S. F. Henssonow, and M. Corporation, *Common Vulnerabilities and Exposures*. Betascript Publishing, 2010.

[19] I. Budi, "Association rules mining for name entity recognition," in *International Conference on Web Information Systems Engineering*, 2003, p. 325.

[20] R. T. Love, A. M. Revel, J. A. Fabien, and R. C. Burbidge, "Event trigger for scheduling information in wireless communication networks," 2006.

[21] L. Hennig, P. Thomas, R. Ai, J. Kirschnick, H. Wang, J. Pannier, N. Zimmermann, S. Schmeier, F. Xu, and J. Ostwald, "Real-time discovery and geospatial visualization of mobility and industry events from large-scale, heterogeneous data streams," in *Acl-2016 System Demonstrations*, 2016, pp. 37–42.

[22] P. Anantharam, P. Barnaghi, K. Thirunarayan, and A. Sheth, "Extracting city traffic events from social streams," *Acm Transactions on Intelligent Systems Technology*, vol. 6, no. 4, pp. 1–27, 2015.

[23] X. Liao, K. Yuan, Z. Li, Z. Li, L. Xing, and R. Beyah, "Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence," in *ACM Sigsac Conference on Computer and Communications Security*, 2016, pp. 755–766.

[24] A. Rajaraman and J. D. Ullman, *Mining of massive datasets /*. Cambridge University Press,, 2012.