iMCircle: Automatic Mining of Indicators of Compromise from the Web

Panpan Zhang[1,2], Jing Ya[1,2], Tingwen Liu[1], Quangang Li[1], Jinqiao Shi[1,3], Zhaojun Gu[4]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
[3]Beijing University of Posts and Telecommunications, Beijing 100876, China
[4]Information Security Evaluation Center of Civil Aviation, Civil Aviation University of China, Tianjin 300300, China
{zhangpanpan, yajing, liutingwen, liquangang, shijinqiao}@iie.ac.cn, 15620968007@163.com

***Abstract*—With the rapidly evolving landscape of cyber threats, Indicators of Compromise (IOCs) are aggressively exchanged as forensic artifacts to help security professionals quickly identify and response cyber threats. Previous related studies mostly focus on extracting and generating IOCs from some fixed-point monitoring data sources, which are passive and time-consuming. In this paper, we present iMCircle, an innovation system that automatically mines IOCs from the Web by checking suspicious indicators with the help of open-source threat information. Based on the initial input of several suspicious indicators, iMCircle first collects their relevant public threat information from the Web and generates IOCs by checking whether those indicators are threat indicators in the target threat field. Second, it actively extracts new indicators from the search results as new inputs and checks them as described above. In that way, the system works in a circle and generates IOCs continuously. Running this system for almost two months in the real world, it has the appreciable performances on the active checking of suspicious indicators and the automatic generation of IOCs.**

***Index Terms*—cyber threats, indicators of compromise, automatic mining, open-source threat information, search engine**

I. INTRODUCTION

Cyber threats have been rapidly increasing in both volume and sophistication [1]. This attracts more and more researchers towards Cyber Threat Intelligence (CTI) as it helps to quickly identify security threats and make the corresponding preventions. CTI is proof-based knowledge that analyzes and explains the details of the existing or evolving cybersecurity threats [2]. To conveniently share CTI, most organizations exchange CTI in the form of Indicators of Compromise (IOCs). IOCs are forensic artifacts that are used as signs that a system has been compromised by an attack or that a system has been infected with a particular malicious software [3], such as domains/IPs of APT, MD5 hashes of attack files, virus signatures.

Given IOCs' importance, there have been many studies on extracting and generating IOCs [3]–[6]. These studies utilized the similar processing strategy: they manually or automatically extracted and generated IOCs from the information gathered by monitoring several public data feeds, like security technical blogs and white papers. Taking [5] as an example, they presented iACE, a fully-automated approach for IOCs extraction. They monitored 45 public blogs and crawled 71,000 articles to obtain relevant threat information. Note that they must gather information in advance and format data from different sources. Altogether, the whole process is passive and relies entirely on the release of information from the fixed-point monitoring data sources. It is very time-consuming and complicated to generate IOCs in this manner.

In this paper, we present iMCircle (IOCs Mining Circle), an innovation system that automatically mines IOCs from the Web by checking suspicious indicators with the help of open-source threat information. This system takes some suspicious indicators (e.g. domains and IP addresses) as input. It first checks if those input indicators are real threat indicators by actively collecting and analyzing their relevant open-source threat information from the Web. According to the checking results, iMCircle generates available IOCs. Then iMCircle automatically extracts new indicators from IOCs' related webpages as new inputs, repeating the checking process above to generate more IOCs. To the best of our knowledge, this is the first work for the automatic generation of IOCs by checking suspicious indicators using relevant public threat information.

To actively collect open-source information, iMCircle needs some input indicators to guide the data collection. We select some suspicious domains as the initial input in our implementation. In fact, there are multiple selections of types and sources for the input indicators, we described in details later in the discussion section. Based on those indicators, we actively gather their relevant open-source threat information from the Web using search engines. The Web can be regarded as a source that contains almost all public data sources. Search engine, such as Google and Bing, is used to find correlated public information quickly. We take the input indicators as the target, and combine with a specific threat field to directly retrieve related information on the Web. This method is efficient and covers much relevant information.

Consider the retrieval results shown in Fig. 1. "APT attack" is the target threat field, "update-java.net" is the input indicator, they together form a query of the search engine. The returned results are highly relevant to the query and semantic-rich. Note that all related results are gained at one time and all in the uniform format. Each search result item also has the same composition, including a title, a URL and a snippet. These results contain relevant potential information that indicates whether the indicator in the target threat field is a real threat indicator. Thus, we directly leverage that

978-1-7281-2999-0/19/$31.00 ©2019 IEEE

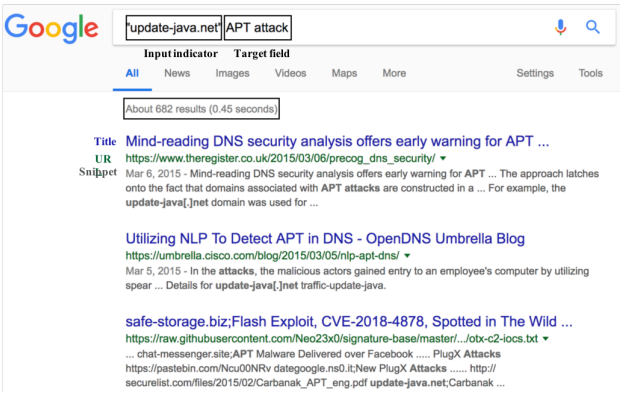

Fig. 1. Retrieval results of an input indicator

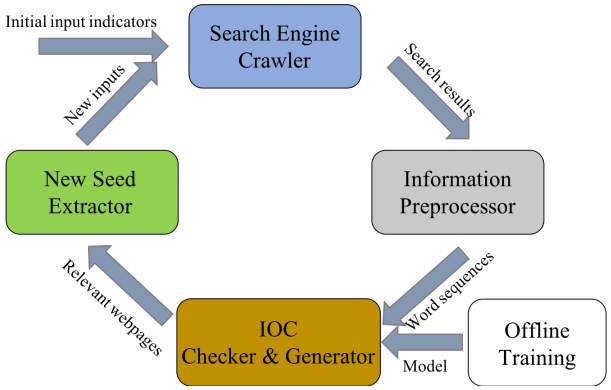

Fig. 2. The architecture of iMCircle

information to check the suspicious indicator. Obviously the cost of processing multiple open-source data in our system is much less than the methods of prior fixed-point monitoring work. In this paper, we reduce the checking problem to a typical binary classification problem that can be solved well by some advanced classification algorithms. According to the checking results, we generate available IOCs using the real threat indicators.

After checking the initial input indicators, we extract new seeds from the search results of the generated IOCs as new inputs to circulate our system and generate more IOCs. This is because that attackers always use multiple similar or related indicators in an attack, resulting in malicious threat indicators usually do not appear separately in one blog. In our system, we leverage the regular expression of domains to extract new indicators from the relevant web content of IOCs' related search results. When getting the new indicators, we check them again as described above to generate more IOCs. In that way, our system works in a circle and mines IOCs automatically. Running our system in the real world, the final evaluation shows that iMCircle has an effective performance on mining IOCs from the Web.

The contributions of this paper are as follows:

- We present iMCircle, the first system that automatically mines IOCs from the Web by checking suspicious indicators using open-source threat information.
- We propose an approach to mine IOCs continuously and automatically, by actively extracting new input indicators from the search results of the checked threat indicators.
- We implement our system and run it in the real world for about two months. In the end, iMCircle achieves the good performances on the active checking of suspicious indicators and the automatic generation of IOCs.

II. IMCIRCLE: DESIGN OVERVIEW

In this section, we introduce the design of our system and explicate how it works through an example.

Architecture. Fig. 2 illustrates the architecture of iMCircle, which consists of four main components, including a search engine crawler, an information preprocessor, an IOC checker and generator, and a new seed extractor.

The search engine crawler is used for the active collection of relevant open-source threat information from the Web. Given some suspicious indicators, the search engine crawler leverages custom queries composed of the target threat field and indicators to gather the related public information efficiently. Different from monitoring several public sources to get threat information, this manner of data collection gets as much information as possible, as well as covers almost all public data sources. The search results are stored in the form of webpages and output as the input of the next component.

After getting the related search results, the information preprocessor parses their HTML files to extract the corresponding text content. Then it exploits Natural Language Processing (NLP) techniques to serialize those results and outputs their word sequences, for capturing the potential semantic features. The separation of data collection and data preprocessing helps improve the overall efficiency of iMCircle, which neither affects the speed of data collection nor interferes with data processing offline.

With the help of the trained model output by the offline training in the right bottom of Fig. 2, the IOC checker and generator first checks whether the suspicious indicators in the target threat field are real threat indicators by using the lexical features captured from word sequences. Then it generates available IOCs using those real threat indicators and outputs them to the IOCs database, for daily accumulation. As of this step, iMCircle has completed the IOCs generation by checking suspicious indicators.

After checking the initial input indicators, the new seed extractor leverages those real threat indicators and their relevant webpages in the search results to extract new indicators. Those extracted indicators are output as the new inputs to our search engine crawler. This results in that our system works in a circle as shown in Fig. 2 and automatically generates more IOCs, until the new seed extractor does not extract new indicators.

An example. Here, we use a specific example in reality to demonstrate how iMCircle works. For some suspicious domains found in daily network traffic, we want to use them to automatically generate IOCs about APT attacks. This problem can be solved well by iMCircle. Firstly, it takes "APT attack" as the target threat field and combines indicators with this

target field as custom queries, as shown in Fig. 1. Then it directly uses those queries in search engines, to actively collect related open-source threat information on the Web. After getting the retrieval results, iMCircle utilizes the information preprocessor to extract text content and serialize them. These sequences have rich semantic features, that helps guide the checking of suspicious indicators. Based on those features and the trained model, iMCircle easily identifies APT domains through the IOC checker and generator. These domains can be treated as IOCs about APT attacks, and saved in the IOCs database. In addition, iMCircle leverages the new seed extractor to seek new indicators as input, for automatically mining more IOCs.

III. APPROACH

In this section, we elaborate on the technical implementation of each component according to their functions.

A. Relevant Information Collection

In this paper, we actively collected open-source threat information from the Web using search engines, different from obtaining the related threat information passively by monitoring several public sources. As shown in Fig. 2, some suspicious indicators must be given as initial input indicators to initiate the active collection. In our implementation, one target threat field is also needs for forming custom queries to quickly locate the relevant threat information on the Internet.

As the Web is an information aggregator from all kinds of public sources, we leveraged it to directly collect as much threat information as possible at one time. Our approach is easy to operate and low cost compared to previous data collection methods that monitor a list of data sources. Moreover, it maximizes public information coverage, and avoids missing valuable information due to incomplete monitoring sources.

In order to quickly locate relevant threat information of indicators on the Web, we used search engines and custom queries composed by indicators and the target field to govern the retrieval of open-source information. Search engine is an efficient tool that helps minimize the time required to find information and the amount of information which must be consulted. To choose a good search engine, we compared the search capability of three search engines, including Baidu, Bing and Google. By analyzing the search results of multiple same queries, we found that Google performed best, with more accurate and comprehensive search results. Thus we chose Google as the search engine used in our data collection, to conveniently obtain most open-source threat information.

With the help of Google search engine, we could get all relevant information for one query at one time. The retrieval results covered almost all public available data sources and are sorted by relevance. In our implementation, we crawled the search pages and saved them in the form of HTML files.

B. Search Results Preprocessing

After obtaining the retrieval results, we first analyzed their HTML files and parsed the related text contents. Then we leveraged advanced NLP techniques to process them to capture the potential features.

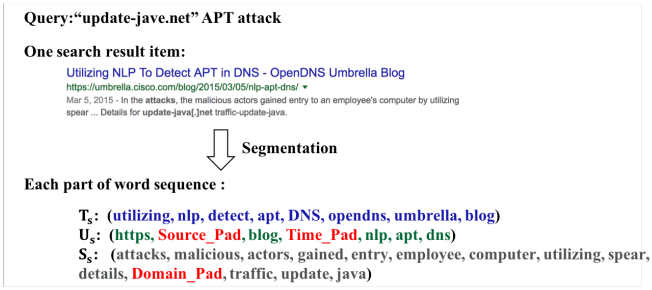

Fig. 3. The consequence of the segmentation

As shown in Fig. 1, each search result item contains three parts, including a title, a URL and a snippet. These parts illustrate the contents of this search result item from different perspectives. For the title, it outlines the content of the corresponding webpage. For the URL, it indicates the source and the webpage type of this search result. For the snippet, it offers a concise but high-quality description of this webpage. In our work, we extracted the content of these three parts based on their tags in HTML files and leveraged them to represent each result item. In order to excavate as much semantic information as possible from those short texts, we used NLP technique to process each part that is pre-converted to lowercase. Here, we elaborate on the process of each part.

- **Title processing**. As a typical short text, the title is highly summary, and describes the main content of the search result item. To avoid losing valuable information, we directly segmented it into words using delimiters such as space, comma and dash. This word sequence completely retains the original semantic information, noted as T_s.
- **URL processing**. The URL is a special short text, since it has a particular hierarchical structure. According to its characteristics, we first divided the URL into three subparts, a protocol, a hostname and a path. To uniform the segmentation, we used a meaningful word of "Source_Pad" to replace the hostname, reducing the additional influence of different hostnames. For the path, we leveraged the delimiter of "\" to split it into words. Finally, we connected words of each part to generate a whole word sequence, noted as U_s.
- **Snippet processing**. The snippet is exactly the contextual text associated with the queried indicator and the target field, with fewer noise data than that on the related webpage. As mentioned earlier, the snippet contains both the queried indicator and the target field. In our work, we firstly replaced the queried indicator in each result with a formatted expression of "Domain_Pad". Then we segmented the snippet into words using the method used in the title and removed stop words using a stop word list. The final word sequence was noted as S_s.

As shown in Fig. 3, we obtained three word sequences from each search result item. At last, we combined these three parts together to obtain a whole word sequence of one search result.

C. Checking Suspicious Indicators and Generating IOCs

In this paper, we are the first to automatically generate IOCs by checking if the suspicious indicators are real threat indicators in the target threat field. In our implementation, we reduced this checking problem into a binary classification problem and used machine learning algorithms to handle it.

To sufficiently capture the potential information in retrieval results, we leveraged the bag of words to vectorize all word sequences, to retain their rich semantic features. Using these features, we selected the SVM algorithm and the Decision Tree (DT) algorithm to train a classification model, guiding the checking of indicators. In order to check suspicious threat indicators in a timely and cost-effective manner, we trained our models using a labeled dataset in advance.

With the help of the trained checking model, we efficiently checked suspicious threat indicators in our system. Taking the processed relevant public search results of them as input, the trained model completed the identification of whether the suspicious indicators in the target field are really threat indicators or not. Since the process of preprocessing suspect indicators' related information is independent of the model training process, we can deal with these two processes in parallel to support the high performance of iMCircle. What's more, we can conveniently replace the existing model with new models trained by more data in the future, to improve the accuracy of checking suspicious threat indicators.

After checking the suspicious threat indicators, we got two kinds of indicators, namely, threat indicators and non-threat indicators. We regarded threat indicators as the real IOCs, since we already checked them using their relevant open-source threat information. In our implementation, we directly output threat indicators with the target field as IOCs to the IOCs database for accumulation.

D. Extracting New Seeds for iMCircle

As of this step, our system has completed the active generation of IOCs by checking the initial inputs of suspicious threat indicators. To reduce human participation and generate more IOCs automatically, we presented to extract new seeds from the relevant public information of the identified threat indicators as new inputs to circulate the system. The reason for this is that each web content in the related search results often contains many correlative threat indicators simultaneously, resulting from cyber attackers always uses many means to achieve the attack purpose.

According to the checked threat indicators, we leveraged their relevant search results to automatically mine more related threat indicators from their webpages. To be specific, using the URL in each search result item of the threat indicators, we first crawled the corresponding webpage. Then we leveraged some regexes of common indicators to extract new indicators from the web content. Taking domains, one type of IOCs as example, using the regex of `([A-Za-z0-9]+(?:[\-|\.][A-Za-z0-9]+)*(?:\[\.\]|\.)(?:DTLD))` (the DTLD symbolizes the common tld used in domains), iMCircle extracted all indicators in this type from the web contents.

Before outputting these indicators to the search engine crawler, we also utilized some custom rules to filter invalid extracted indicators, such as illegal domains and unregistered domains. After getting the final set of indicators, we leveraged them as the new inputs to iMCircle, for generating more IOCs by repeating the whole process above, until there are no new indicators found in the relevant web contents.

IV. EVALUATION

To demonstrate the performance of iMCircle, we implemented and evaluated it using the indicators of domains on the target field of APT attack in our work. In this section, we first introduce the details of the experimental setup. Then we analyze the results of our experiments to evaluate the effectiveness of our prototype system.

A. Dataset

To the best of our knowledge, this is the first work to actively check suspicious indicators in the target field using open-source threat information. There is no existing dataset available. Thus, given the target field of APT attacks, we selected known domains as facts to construct a labeled dataset for training the offline classification models.

We manually collected 6400 threat domains from 400 APT attack reports (named APT domains), published by famous security companies, such as FireEye and Kaspersky Lab. These APT domains are threat indicators and have been used in the process of APT attacks, noted as positive indicators. We also collected 10,000 normal domains from Alexa which is a famous website to provide popular normal traffic data. They are noted as negative indicators. There is no overlap between these two kinds of indicators. Then we utilized the search API provided by Google and the custom query {"domain name" APT attack}, e.g., {"update-jave.net" APT attack} in Fig. 1, to actively collect relevant public threat information on the Web. After retrieving the relevant threat information on the Web, we saved the search results to fill in the dataset. In the end, we acquired a dataset that contains 15243 indicators, including 5848 positive indicators and 9395 negative indicators. Note that the number of samples is less than the expected, owing to crawling restriction and the removing of empty results.

B. Results

Accuracy and coverage. We first evaluated the classification models with the labeled dataset using a 10-fold cross validation. In our implementation, we trained two classification models of the SVM model and the DT model. The experimental results are shown in Table I. For checking whether the suspicious indicators are real threat indicators in the target threat field, the SVM model achieved a precision of 96.70% and a recall of 84.89% while the DT model had a precision of 88.45% and a recall of 91.47%. On the whole, the SVM model had a better performance with a 90.46% F1 score, which was used in iMCircle.

TABLE I
THE RESULTS OF THE TWO MODELS

Models	Precision	Recall	Accuracy	F1 score
DT	88.45%	**91.47%**	92.13%	89.93%
SVM	**96.70%**	84.89%	**93.11%**	**90.46%**

TABLE II
STATISTICS OF PUBLIC SOURCES OVER ALL SEARCH RESULTS OF APT DOMAIN

Rank	Source	#results
1	github.com	3616
2	threatminer.org	2771
3	seebug.org	1134
4	raw.githubusercontent.com	1122
5	docplayer.net	929
6	securelist.com	886
7	actortrackr.com	738
8	bitbit.net	619
9	kasperskycontenthub.com	530
10	archive.is	514

TABLE III
STATISTICS FOR DIFFERENT INPUTS OF IMCIRCLE

	#Indicators	#Checked IOCs	#Verified IOCs
Initial input	15243	5135	4966
New input	15549	2100	1910

In addition, we verified the high coverage of public sources in our system. Specifically, we analyzed the search results of all APT domains. The URL in each retrieval result hints at the source of the result. We regarded the hostname in each URL as the source. From the statistical result, we found that all APT domains involved 2941 public sources. The top 10 open sources are listed in Table II. We found that the first source is "github.com", an emerging public source that many people like to share threat information on. And some of the rest are usually used in the existing monitoring approaches, such as "threatminer.org", "securelist.com" and "actortrackr.com". Except these popular public sources, our results also included lots of niche public sources. Those niche public sources provide some valuable information ignored by common popular sources.

Performance. To understand the performance of iMCircle, we measured its ability to automatically generate IOCs from the Web. We ran our system in the real world.

According to the verified IOCs from the initial input indicators, iMCircle automatically mined 15549 new indicators and generated 2100 new IOCs. The statistics for different inputs of iMCircle are listed in Table III. Note that the "#Checked IOCs" is the number of the threat indicators classified by the SVM model and the "#Verified IOCs" is the number of actual threat indicators verified manually. We evaluated our system on the new generated IOCs using the human verification. We first submitted the 2100 new checked IOCs to VirusTotal, a public platform to assist people for identifying threat indicators. From VirusTotal, we got that 1671 indicators are malicious indicators among them. Then we manually verified the remain 429 indicators with some extra information and experience, and found 239 threat indicators again. All in all, we obtained 1910 real threat indicators from the new input, with a accuracy of 91%. This evaluation indicated our iMCircle had a appreciable performance on the automatic generation of IOCs.

Given its effectiveness of checking suspicious indicators and generating available IOCs, our system can be used to help cybersecurity professionals to implement manual judgments after maliciousness detections. Although there are many kinds of research approaches on network traffic analysis on multiple dimensions in recent years, cybersecurity professionals still must conduct manual judgment to the results after using such detection approaches in reality. iMCircle can be used to mine open-source CTI for the highly-suspicious domains/IPs which are labeled as "maliciousness" by existed detection models, and provide reference information for them. In that way, our work can reduce the workload of the manual judgment.

V. DISCUSSION

iMCircle is designed to automatically generate IOCs by actively checking suspicious indicators with the help of open-source threat information from the Web. As mentioned earlier, this system needs several suspicious indicators as the initial input to guide the active data collection. In addition to obtain the uncertain indicators from the daily network traffic, there also are several ways to get the input, such as reputation systems for malicious domains detection (e.g., Notos [7]), and open-source IOC parse tools (e.g. ThreatMinder). Notos system outputs reputation scores for domains indicative of whether the domain is malicious or legitimate. Domains with lower reputation scores are more likely to be malicious, but still need further analysis. ThreatMinder provides an IOC parse tool to extract IOCs contained in threat information (such as APT reports and security blogs). It may also extract legitimate indicators which happen to appear in threat information. Besides, with different types of input indicators and target fields, iMCircle can generate different types of IOCs, e.g. IPs and the botnet attack. More combinations is needed to fully mine IOCs from the Web in the future work.

We actively check suspicious threat indicators with open-source threat information, and generate IOCs in the target threat field. Different from the existing passive IOC generation process, we fully utilize as more information as possible from the Web. In that way, we expand the coverage of available public data feeds, and improve the accuracy of the extracted IOCs. In our work, we totally found 2941 distinct data sources directly collected from the Web. This implies that if we want to collect the same data by fixed-point monitoring, we need to monitor all of those data sources. Note that our system completely dependents on the open-source threat information from the Web, and we do not consider the suspicious indicators which are not open on the Web.

VI. RELATED WORK

A. Cyber Threat Intelligence

CTI is vital for security community and organizations to defend against the fast-evolving threat landscape. As mentioned earlier, CTI is always collected and exchanged in the form of IOCs. Public threat reports and platforms such as ThreatExchange, Alienvault OTX, ThreatMiner and IOC Bucket, are

valuable sources for threat intelligence sharing. And there have been many efforts to provide standardized machine-readable format to facilitate CTI sharing, e.g., OpenIOC [8], STIX [9] and TAXII [10]. Besides, much research work [5], [6], [11]–[13] has been proposed to extract and generate open-source CTI. They presented systems or approaches to generate machine-recognizable CTI from different monitored public sources. There also has some similar work to produce other types of CTI from other sources. For example, [14] presented a system to give the end user cybersecurity intelligence alerts using publicly available data from Twitter. Our work is different from those work. They mostly passively collected threat information by monitoring several public sources. We actively gathered relevant threat information from the Web by directly using search engines, that is more efficient.

B. Security Work Supported by Search Engines

Search engines are the information retrieval systems designed to easily find lots of relevant information with the input queries from the Web. The retrieval results are processed data with high quality. There are many work used search engines to help solve security problems. [15] performed semantic inconsistency search based on search engines to detect the promotional injections on sponsored top-level domains with explicit semantic meanings. [16] utilized search engine visibility (whether the site is indexed) as a factor to determine the maliciousness of a site. [17] uses search engines with the specific keywords to find all relevant information and detect shadowed domains automatically. [18] used the indexed pages of search engines to discover trending terms used in underground markets. In this paper, we used search engines to help check suspicious threat indicators are real threat indicators in the target field. With the help of search engines, we efficiently collected high-quality threat information from the Web to guide our checking.

VII. CONCLUSION

In this paper, we present iMCircle, a novel system that automatically generates IOCs from the Web by checking suspicious indicators using relevant open-source threat information. It can not only identify threat indicators accurately but also generate IOCs actively and continuously. In our system, we directly leverage Google search engine to collect related public information of indicators on the Web. This is more convenient and efficient than that using passive monitoring strategy to collect threat intelligence. For the checking problem, we reduce it to a binary classification problem and utilize a flexible classification model to accurately check the suspicious indicators. According to the checked results, iMCircle generates and accumulates IOCs in a circle. Running this system for almost two months in the real world, it has the appreciable performances on the active checking of suspicious indicators with the F1 score of 90.46% and the automatic generation of IOCs with the accuracy of 91%.

In the future, we will make efforts to improve the performance of the offline models and attempt more types of indictaors mining, for pursuing a fully-automatic and comprehensive gathering of cyber threat intelligence.

ACKNOWLEDGMENT

This work is supported by Open Project Foundation of Information Security Evaluation Center of Civil Aviation, Civil Aviation University of China(No.CAAC-ISECCA-201801).

REFERENCES

[1] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, vol. 1, no. 1, p. 80, 2011.

[2] J. Friedman and M. Bouchard, *Definitive Guide to Cyber Threat Intelligence: Using Knowledge about Adversaries to Win the War against Targeted Attacks*. CyberEdge Group, 2015.

[3] O. Catakoglu, M. Balduzzi, and D. Balzarotti, "Automatic extraction of indicators of compromise for web applications," in *Proceedings of the 25th International Conference on World Wide*. International World Wide Web Conferences Steering Committee, 2016, pp. 333–343.

[4] J. Andress, "Working with indicators of compromise," *Journal Information Systems Security Association (ISSA)*, vol. 5, 2015.

[5] X. Liao, K. Yuan, X. Wang, Z. Li, L. Xing *et al.*, "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.

[6] A. Panwar, "iGen: Toward automatic generation and analysis of indicators of compromise (iocs) using convolutional neural network," Master's thesis, Arizona State University, 2017.

[7] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a dynamic reputation system for dns." in *USENIX security symposium*, 2010, pp. 273–290.

[8] (2016) OpenIOC. [Online]. Available: https://www.darknet.org.uk/2016/06/openioc-sharing-threat-intelligence/

[9] (2018) Structured threat information expression (stix). [Online]. Available: https://oasis-open.github.io/cti-documentation/

[10] (2018) Trusted automated exchange of intelligence information (taxii). [Online]. Available: https://oasis-open.github.io/cti-documentation/

[11] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu, "TTPDrill: Automatic and accurate extraction of threat actions from unstructured text of CTI sources," in *Proceedings of the 33rd Annual Computer Security Applications Conference*. ACM, 2017, pp. 103–115.

[12] 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 Symposium*, 2015, pp. 1041–1056.

[13] I. Deliu, C. Leichter, and K. Franke, "Extracting cyber threat intelligence from hacker forums: Support vector machines versus convolutional neural networks," in *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3648–3656.

[14] S. Mittal, P. K. Das, V. Mulwad, A. Joshi, and T. Finin, "Cybertwitter: Using twitter to generate alerts for cybersecurity threats and vulnerabilities," in *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE Press, 2016, pp. 860–867.

[15] X. Liao, K. Yuan, X. Wang, Z. Pei, H. Yang, J. Chen, H. Duan, K. Du *et al.*, "Seeking nonsense, looking for trouble: Efficient promotional-infection detection through semantic inconsistency search," in *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE, 2016, pp. 707–723.

[16] J. Zhang, X. Hu, J. Jang, T. Wang, G. Gu, and M. Stoecklin, "Hunting for invisibility: Characterizing and detecting malicious web infrastructures through server visibility analysis," in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*. IEEE, 2016, pp. 1–9.

[17] D. Liu, Z. Li, K. Du, H. Wang, B. Liu, and H. Duan, "Don't let one rotten apple spoil the whole barrel: Towards automated detection of shadowed domains," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 537–552.

[18] H. Yang, X. Ma, K. Du, Z. Li, H. Duan, X. Su, G. Liu *et al.*, "How to learn klingon without a dictionary: Detection and measurement of black keywords used by the underground economy," in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017, pp. 751–769.