2019 IEEE International Conference on Big Data (Big Data)

IoCMiner: Automatic Extraction of Indicators of Compromise from Twitter

Amirreza Niakanlahiji
University of Illinois at Springfield
aniak2@uis.edu

Lida Safarnejad
University of North Carolina at Charlotte
lsafarne@uncc.edu

Reginald Harper
University of North Carolina at Charlotte
rharpe10@uncc.edu

Bei-Tseng Chu
University of North Carolina at Charlotte
billchu@uncc.edu

***Abstract*—In recent years, cyber attacks have consistently grown in terms of volume, sophistication, coordination, and pervasiveness. Such attacks impose billions of dollars loss to companies and government entities annually. Sharing cyber threat intelligence (CTI) about ongoing attacks can significantly improve the current situation as many cyber attackers tend to reuse or share their network infrastructure, techniques, tactics, and procedures across multiple attacks. Therefore, many security professionals devote their time and effort on hunting cyber threats and sharing such valuable information with the public through public data sharing platforms such as social media and text sharing websites. However, due to the sheer volume of information that is being shared on such platforms; finding CTI information is tantamount to looking for a needle in a haystack. In this paper, we present a new scalable framework, IoCMiner, to automatically extract CTI, in special Indicators of Compromise, from Twitter. It utilizes a combination of graph theory, machine learning, and text mining technique to achieve its goal. IoCMiner relies on a reputation model to discover credible twitterers who publish CTI, and only tracks the tweet stream of such Twitter handles. Moreover, it employs a CTI classifier to further filter out non-CTI tweets from the observed data streams. Finally, IoCs uses a set of regular expression rules to extract IoCs from the identifies tweets. Through experimentation, we show the usefulness of IoCMiner in finding fresh IoCs from Twitter. In the course of four weeks, IoCMiner identified more than 1,200 IoCs, including malicious URLs. Only 10% of the URLs were already listed in public blacklist databases at the time of extraction. The number of URLs that appeared in blacklists increased to 26% after one week.**

***Index Terms*—Indicators of Compromise, Cyber Threat Intelligence, Topical Expert Model, Social Media Mining, Twitter**

I. Introduction

Sharing information about recent cybersecurity incidents can considerably reduce the existing knowledge gap between defenders and attackers as the techniques, tactics, and procedures used in one cyber attack can be reused to attack other organizations with similar environments. In addition, the system and network infrastructures used by attackers to target a victim are commonly reused in other attacks. In recent years, many security companies have emerged that are specialized in collecting and characterizing cyber incidents and sharing extracted intelligence with other companies or the public to prevent such reuses. For example, abuse.ch have several trackers for tracking command and control (C&C) servers of famous botnets such as Zeus. Network defenders can consume their Zeus tracker feed to block network traffics destined to Zeus C&C servers; thus neutralizing Zeus bots.

Despite the achievements of such cyber threat intelligence companies, they still suffer from the following two problems. First, their coverage of cyber threats is far less than ideal, which means clients need to aggregate from many of such companies to cover a good percentage of ongoing threats. Second, there is a significant delay between the time of receiving threat signals to the time of identifying and publishing the threat reports. To address these two problems, we introduce IoCMiner, a framework to extract cyber threat intelligence, in particular IoCs, from public information-sharing platforms, such as social media, discussion forums, and text sharing websites, where a large number of individuals and companies share their findings of ongoing cyber attacks. It relies on concepts and techniques borrowed from graph theory, text mining, and machine learning.

IoCMiner is a lightweight online framework with no dependency on external systems. As a result, it can scale well with the amount of information published on popular data-sharing platforms. It is online (*i.e.*, it processes the input data in near real-time) as the threat landscape is continually evolving; which, on average, cause threat information to expire shortly after being reported by cybersecurity professionals on these platforms. Due to the sheer amount of published information on data sharing platforms, instead of directly examining published information, IoCMiner continuously attempts to identify reliable cyber-threat intelligence sources on the target platforms. Only contents published by such sources is analyzed to extract cyber-threat intelligence.

The current prototype of IoCMiner, accessible on GitHub [10], supports Twitter and Pastebin; however, it can be extended to support other data-sharing platforms. Due to the nearly ubiquitous adoption of social media platforms such as Twitter, IoCMiner can significantly improve the coverage problem and complement data available through traditional channels. Our experimentation with IoCMiner on Twitter also

978-1-7281-0858-2/19/$31.00 ©2019 IEEE

confirms that a large volume of fresh threat intelligence information is shared on this platform by cybersecurity professionals.

II. Background

In this section, we define the terms and concepts that are frequently referred throughout this paper and are essential for understanding the contribution of the presented work.

A. Cyber Threat Intelligence

In this work, one of our primary goals is to extract cyber threat intelligence from publicly available data sources including social media and cyber threat repositories. Rob McMillan [11], a Gartner analyst, defined threat intelligence 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's response to that menace or hazard". This definition can be used to define cyber threat intelligence (CTI) as threat intelligence related to computers, networks, and in general information technology (IT) systems.

B. Indicator of Compromise (IoC)

Indicators of Compromises (IoCs) are network or system artifacts that are observed during a cyber attack. IoCs can be categorized in different ways; a common way to do so is based on the granularity of data represented by IoCs [4]. In this categorization, IoCs are divided into three groups: atomic, computed, and behavioral IoCs. Atomic IoCs, such as ip addresses, domain names, registry keys, and process names, represent network or system artifacts being observed during a cyber attack. Computed IoCs are the ones that are calculated from data observed during the attack such as hash values of malware instances. The behavioral IoCs are the ones that are a combination of the other IoCs such as a malicious docx file X with a hash value of Y is hosted on server Z, upon opening the docx, a malware W will be executed on the victim machine.

III. Problem Statement

This paper addresses the problem of extracting cyber threat intelligence, in particular indicators of compromise (IoCs), from data shared on information sharing web platforms. To be more specific, given the stream of data published on such a platform, we want to: 1) recognize reliable cyber threat-related data, 2) extract atomic IoCs from them, and 3) aggregate the resulted atomic IoCs to obtain a more comprehensive picture of the associated threat.

Without loss of generality, we focus on: 1) Twitter, a microblogging social network, as it is one of the most famous social networks with more than 330 million active users sharing over 500 million tweets on a variety of topics per day, and 2) Pastebin, a text sharing platform, with more than 95 million text documents.

Due to the immense number of tweets published every day, and the existence of a significant imbalance between the number of security and non-security related tweets, any viable online solution must adopt a strategy to avoid processing each and every tweet published on the platform. As mentioned earlier more than 500 million tweets are tweeted every day, which means on average 3500 tweets per minute are streamed to subscribed applications. It is worth noting that Twitter only publishes one percent of the whole tweets through its streaming API. Without any filtration, a system must be able to process 3,500 tweets per minute. Almost all of these tweets are not related to cyber threats; thus most of the resources and time will be wasted on tweets that are not relevant to its goal. More importantly, because of this significant imbalance between threat and not-threat tweets, even a minuscule inaccuracy in differentiating between the two can make the system useless as it may render too many false positives.

To address these issues, instead of examining all tweets, IoCMiner monitors tweets posted by a set of users who have shown interest in tracking cyber threats and publishing their IoCs. In this way, IoCMiner receives a significantly lower number of tweets required to examine per minute; thus can perform more process-intensive operations. Moreover, the ratio of tweets containing IoCs increases drastically. This enables me to employ a classifier to identify IoCs without getting too many false positives or negatives. Most of the related research work on social media solely focus on finding influential users, the ones who have significant effects on other users' behavior. Despite its importance and relevance to this work, the primary goal of this work is not to identify such users. Instead, the focus is on identifying tweets that contain valuable information about ongoing cyber attacks. This information may or may not be originated or disseminated by influential users on social networks.

IV. IoCMiner Architecture

Figure 1 depicts the overall architecture of IoCMiner, which is consist of two separate subsystems, namely the CTI Expert Finder (CTIEFinder) and CTI Extraction subsystems. CTIEFinder, periodically, examines potential users on Twitter to identify CTI experts who consistently publish high-quality information about ongoing attacks. CTI Extraction subsystem continuously monitors tweet stream containing tweets posted or shared by identified CTI experts and extract useful cyber threat information.

To discern cyber threat intelligence experts, who are willing to share their knowledge, from other users, CTIFinder analyzes users' tweeting history and measures the amount of cyber threat information that each of them has already shared with the public. The underlying assumption is that the probability of publishing IoCs by users with good track records is significantly higher than others. However, one must also ensure that the identified users are credible sources of information. To evaluate the credibility of such users, CTIEFinder considers a range of features extracted from their tweet history, their relationships with other users, and also the lists that they are a member of. The underlying assumption is that credible users tend to follow credible users and also only like or

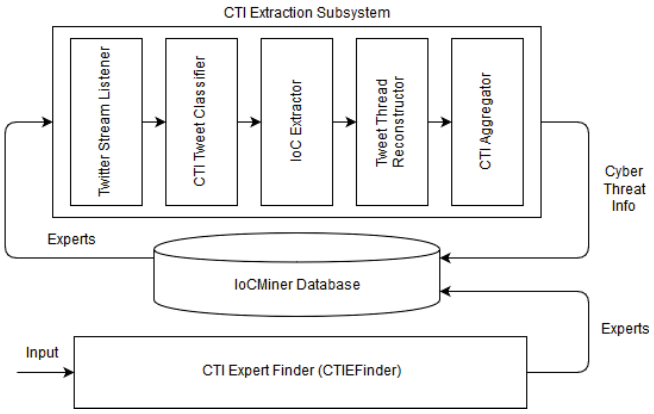

Fig. 1. IoCMiner Architecture

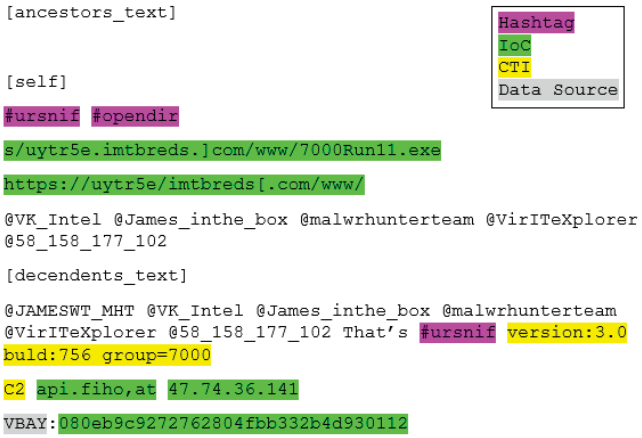

Fig. 2. Example of a CTI tweet thread

retweet tweets that seem to be valid. Moreover, users who create lists about a specific topic are knowledgeable about the topic and tend to only add topical experts who publish useful information to their lists.

After identifying CTI experts, the next step is to monitor their tweet activities and analyze them to extract useful CTI information. CTI extraction subsystem continuously collects live tweets posted by the experts. Since not all of the collected tweets are related to cyberattacks, CTI extraction subsystem employs a custom classifier, called CTI Tweet Classifier, to separate cyber threat tweets from the non-CTI ones. Tweets are then passed to IoC Extractor to mark IoCs within the CTI-labeled tweets. IoC Extractor uses a set of regular expression rules to recognize IoCs. All tweets are then passed to Tweet Thread Reconstructor which constructs tweets threads by analyzing their `in_reply_to_status_id` fields; a sample tweet thread can be seen in Figure 2. Finally, CTI Aggregator links CTI tweet threads, the ones containing at least one IoC, with each other based on shared IoCs and hashtags.

V. Identifying Cyber-Threat Intelligence Expert

In this section, we explain the internal of the Cyber Threat Intelligence Expert Finder, CTIEFinder, subsystem. This subsystem is employed by IoCMiner to discover cyber threat intelligence experts who consistently publishes IoCs related to ongoing threats on Twitter. It exploits the relationships between users and lists on Twitter to identify potential CTI experts. It, then, further prune the list of candidates by examining their tweet histories.

On Twitter, users can create lists to categorize users into groups based on some criteria. For example, a user can create a list for tracking cybersecurity news and add cybersecurity journalists to this list. Logically, users only add twitter handles that publish useful information related to the topic of their interest. As mentioned by other researchers [5], these user-defined lists are valuable resources for identifying topical experts. IoCMiner also relies on user-defined lists to find cyber-threat intelligence experts. It exploits the relationship between users and lists to identify potential topical experts and then further analyzes their tweet histories to ensure their expertise.

Each user-defined list has a number of members, who are added by the list owner. In addition, a user can be a member of multiple lists. The many-to-many relationships between lists and users can be modeled by a bipartite graph as shown in Figure 3. It worth noting that not all lists related to a specific topic has the same quality or specificity. In general, the more selective lists are the better ones. CTIEFinder relies on several metrics to measure the quality of user-defined lists in a specific topic; in this work cyber-threat intelligence. These metrics measure the relevancy, popularity, comprehensiveness of a list in addition to the credibility of its owner.

- Relevancy score - is a composite indicator that measures the degree of which a given list is relevant to a topic of interest, in this work cyber threat intelligence. As mentioned by other researchers [5], the name and description of a list are valuable semantic cues that can be exploited to uncover its topic. Formula 1 is defined to measure the relevancy of a list to CTI topic based on its name and description.
  Let $Lists$ be a set containing all the lists to be examined, and $list$ be one of these lists. In addition, Let t_{list} represents the

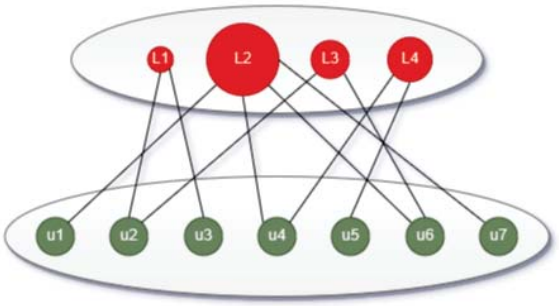

Fig. 3. Relationship between users and lists is modeled as a weighted bipartite graph

TABLE I
KEYPHRASE SETS IN IOCMINER. NOTE .? MEANS ZERO OR ONE CHARACTER

Specific Keywords	ioc
	malware
	indicator.?of.?compromise
	threat.?hunt
	phishing.?hunt
	phish.?hunt
	threat.?int
	threat.?research
	ransomware
	mal.?doc
Generic Keywords	info.?sec
	cyber.?sec
	security

concatenation of $list$ name and description. Moreover, let G be a list containing sets of keyphrases. In current IoCMiner prototype, we manually defined two sets of keyphrases: specific and generic keyphrase sets (as defined in Table I). Specific keyphrases are the ones that are closely related to the topic of interest (*i.e.*, CTI). Generic keyphrases are the words that are related to the domain in which the topic resides (*i.e.*, cybersecurity).

Let $relevancy\_nd$ be a function to compute the relevancy score of $list$ based on its name and description. It is defined as:

$$relevancy\_nd(list) = \sum_{i=0}^{|G|} w_{G_i} * |match(t_{list}, G_i)| \quad (1)$$

Where $match$ is a function that takes t_{list} and G_i, and returns the number of times keyphrases in G_i appeared in t_{list}. Moreover, w_{G_i} is the weight assigned to G_i; adjustable by the IoCMiner operator.

It is worth noting that the name and description of a list are very short and they may not entirely reflect the content of the list. To further check the relevancy of a list to the topic of interest, we define $relevency\_hist$ function to measure the relevancy of a list based on its published statuses (*i.e.*, tweet history). In this function, the textual contents of the last N tweets posted in the input list are examined to see whether they contain any IoC. It is also crucial to consider the number of times a specific IoC appeared in various lists as the ones reported in many lists are less attractive than the ones reported in a few lists.

Let $relevency\_hist$ be a function to calculate the relevancy of $list$ to the topic of interest. $relevency\_hist$ is defined as:

$$relevency\_hist(list) = \sum_{i \in IoC_{list}} \frac{1}{|\{l \in Lists, i \in IoC_l\}|} \quad (2)$$

Where IoC_{list} is a set of all IoCs appeared in $list$ and $\{l \in Lists, i \in IoC_l\}$ is a set of all lists containing IoC i.

- Popularity score - measures how much a list received attention from the community. Users can subscribe to lists that they like in order to see their timeliness. Our intuition is that better quality lists attract more subscribers. To calculate this metric, we consider the number of subscribers of a list:

$$popularity\_score(list) = subscriber\_count(list) \quad (3)$$

- Completeness - measures the coverage of lists in the terms of the number of experts they are enlisted.

$$member\_score(list) = \frac{member\_count(list)}{log_2(member\_count(list))} \quad (4)$$

- Owner credibility - measures how credible a user is. To calculate this metric, we consider the number of follower and friends of the list owner. It is defined as:

$$cred\_score(list) = log_2\left(\frac{|owner\_followers(list)| + |owner\_friends(list)|}{|owner\_friends(list)|}\right) \quad (5)$$

A multiplicative scoring approach is used to combine the described metrics to calculate the overall score for a list. To be more specific, the overall score of a list is computed with the following formula:

$$overall\_score(list) = \prod_{k=0}^{|Scores(list)|} \left(\frac{score_k(list)}{avg(\{score_k(l) : l \in Lists\})}\right)^{w_{Score_k}} \quad (6)$$

Where $Scores(list) = \{relevancy\_nd(list), relevency\_hist(list), popularity\_score(list), member\_score(list), cred\_score(list)\}$, $score_k(list)$ is kth item in $Scores$ $(list)$, and w_{score_k} is the weight for the kth score in Scores list. By changing w_{score_k}, an analyst can increase or decrease the importance of score k in $Scores(list)$.

In formula 6, all the metric scores are first normalized by dividing them with the average value of these metrics. By doing so, the measurement unit for the metrics are not important and one can compare the metrics. Then, they are multiplied with each other to get the final score for each list. It is worth noting that in this formula, metric scores below the average will be in the range [0, 1) and scores above the average will be between 1 and positive infinity. As a result, scores below average have an adverse on the magnitude of the final list score, and scores above average have a positive

contribution. Resulted list scores are, then, used as the weight of lists in the bipartite graph.

After ranking the lists, CTIEFinder takes the top N lists. The expectation is that users in these lists are topical experts. However, the overall score does not directly show the expertise level of each individual user in the list; it only shows the collective effort of its users. After finding such lists, the goal is to select the most valuable experts among the members of the lists. Intuitively, users who are listed in many lists with high weights are considered better in the eyes of the community. CTIEFinder uses formula 7 to compute the credibility of users based on the lists that they were added to, which indicate the amount of belief that the list owners have in these users.

$$user\_list\_score(u_i) = \sum_{l \in neighbor\_lists(u_i)} overall\_score(l) \tag{7}$$

Where $neighbor\_lists(u_i)$ are the nodes (*i.e.*, list) that are directly connected with u_i in the bipartite graph. In other words, it represents the lists that u_i is a member of.

CTIEFinder, then, select top $5 * k$ users based on the calculated $user\_list\_score$ scores. However, without examining the tweet activities of users, one cannot be confident that the users will publish IoCs in the future. Both the quantity and quality of information disseminated by users are important. In terms of quantity, we want to identify experts that are highly active in posting information regarding a topic domain. In terms of quality, we also want to get fresh and accurate data. The focus is to identify those experts that generate new credible data.

The expectation is that the publishing history of a user is a good indicative of their behavior in the near future. If they published lots of IoCs in the past, they will do in the future. As a result, CTIFinder counts the number of relevant posts that a user published in the past. In the current prototype of CTIFinder, the most recent 400 tweets of a user are considered. In general, recent history is a better predictive of the future than far back history. Thus, the importance of counts must decay as one goes far back in history. To this end, CTIFinder uses a polynomial function to assign weights for each day in the history as shown in formula 8.

$$user\_ioc\_score(u_i) = \sum_{d=0}^{365} \frac{ioc\_count(u_i, d)}{(d+1)^{\frac{1}{3}}} \tag{8}$$

ioc_count functions returns the numbers of IoCs published by u_i in d days before today. The final score is then calculated by multiplying $user\_list\_score(u_i)$ with $user\_ioc\_score(u_i)$. The top k users will then be selected by CTIEFinder as CTI experts who publish cyber threat information, in special IoCs.

VI. Classifying Tweet Streams

In this section, we describe the CTI Tweet Classifier module, which is responsible for tagging tweets with CTI or Non-CTI labels, in IoCMiner. The input of this module is a sequence of tweets collected by Tweet Stream Listener. By using this classifier, IoCMiner can narrow down the IoC extraction to only CTI tweets, which increases the degree of accuracy. Classifying tweets also helps to improve the output of CTI aggregator module as it only considers tweet threads containing CTI tweets and attempt to join them.

CTI Tweet Classifier, first, tokenize the input tweets, turning them into bags of words. Next, each bag is filtered by removing stop words and then stemming the words. It, then, counts the number of each word in a tweet and make a vector from these counts. The resulted vector is ultimately used as features to help classification. Finally, it performs machine classification on the vectorized tweets to identify the one that contains IoCs. CTI Tweet Classifier utilizes RandomForest algorithm to recognize tweets containing IoCs. To be more specific, it constructs a classifier by training the Random Forest classifier on a set of already-labeled tweets. The features in this classifier are the words that constitute the tweet contents, number of hashtags, and the number of mentions in the training dataset.

VII. Evaluation

In this section, we evaluate the effectiveness of IoCMiner in extracting fresh IoCs from tweet streams. We narrow down the focus to atomic IOCs and primarily on the quality of data that can be obtained from tweets published on Twitter. To evaluate IoCMiner, we seek to answer the following research questions: first, whether security professionals commonly publish IOCs on Twitter. Second, whether the atomic IOCs extracted from tweets are fresh; this is important due to the ever-evolving threat landscape. Third, whether data published on Twitter is first-hand; in other words, whether twitterers publish first-hand data, unpublished by traditional channels, or they are just rehashing the existing knowledge known to the cybersecurity community. The input to IoCMiner is a list of manually-selected security professionals that publish IOCs on Twitter. The output is a growing list of IOCs published on Twitter. In the rest of this section, we describe how the current implementation of IoCMiner works. The source code of IoCMiner prototype used for conducting the evaluation in this section can be accessed on Github [10].

A. Observation list

To bootstrap IoCMiner, one must input a seed list of exemplar cybersecurity experts who publish IoCs on Twitter. We created a seed list by manually searching different keywords related to cyber threats, including malware names such as Emotet and GandCrab, to identify security professionals who publish cyber threat-related data. I, further, considered the number of followers, the company that they work for, and the number of security-related tweets that they had published. In this way, 62 well-known threat intelligence experts were identified and added to the seed list; which then was fed to IoCMiner as input for experimentation.

For each seed user, IoCMiner fetches the metadata information, such as name, description, member count, subscriber count, and owner info, of all lists that the user has been added

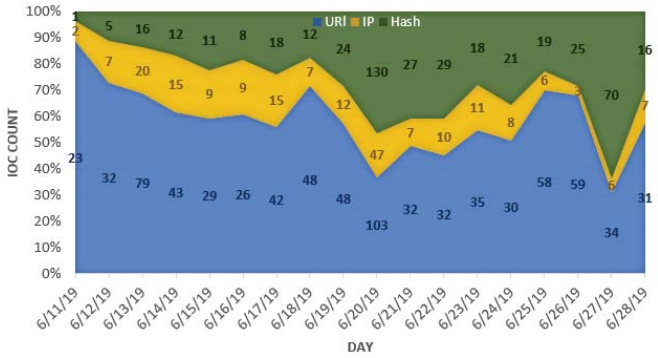

Fig. 4. Extracted IoCs, namely URLs, IP addresses, and hashes (total IoCs: 2261)

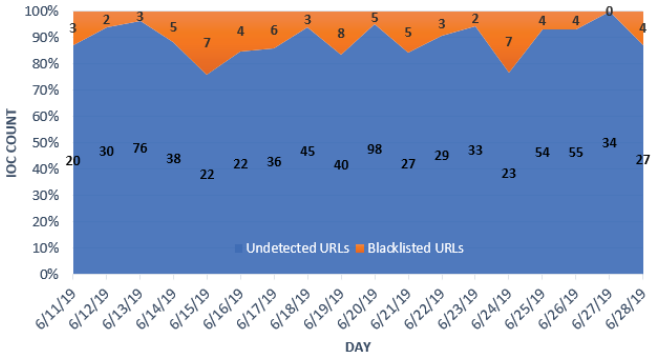

Fig. 5. Malicious URL collected over three weeks by IOMiner. 116 out of 1208 URLs were blacklisted by Google SB

to; this process resulted in retrieving metadata information for 5,851 lists in our experiment. Based on collected information, IoCMiner ranks the retrieved lists and picked the top 1,000 ones based on their metadata. For each of these selected lists, IoCMiner further retrieves 1,000 recent tweets and user information of all of its members. In this way, IoCMiner selected 118,847 users. Next, it adjusted the scores of each list based on the new information and the formula presented in section V. IoCMiner, then, computes the score for each of the users in these lists based on the list scores that they were a member of and, then, selected the 5,000 users with the highest scores. Next, for each of these users, it retrieved 400 recent tweets and used that information to readjust the scores for the top 5,000 users. The top 1,000 users then constituted the observation list. In addition to the IoCMiner selected top 1000 cyber threat intelligence experts, we selected 1,000 users randomly from the remaining users (117,847) and added them to the observation list to represent the baseline.

B. IOC extractor

We have simplified the problem of extracting IoCs to extracting malicious URLs, IP addresses, and hashes reported by security professional on Twitter. The current implementation of IOC extractor module in IoCMiner relies on a set of regular expression rules to identify such IoCs. We observed that security professionals do not post malicious URLs and IP addresses in a well-formatted form to prevent unwary users from accidentally clicking these links and infecting themselves. For example, instead of starting URLs with http, they may start malicious URLs with hxxp or / (slash). During our experimentation, we identified several such patterns and created regular expression rules to match them. Figure 4 shows the number of IoCs harvested by IoCMiner between June 11th to July 8th. During this period, 1208 of the IoCs were URLs. To test the freshness of these URLs, we checked them with Google Safe Browsing List (SBL) on the time of extraction. On average, less than 10 percents of extracted URLs marked as malicious by Google SBL as shown in Figure 5.

We rescanned the undetected URLs every day for one week after their extractions to see when these URLs will be added

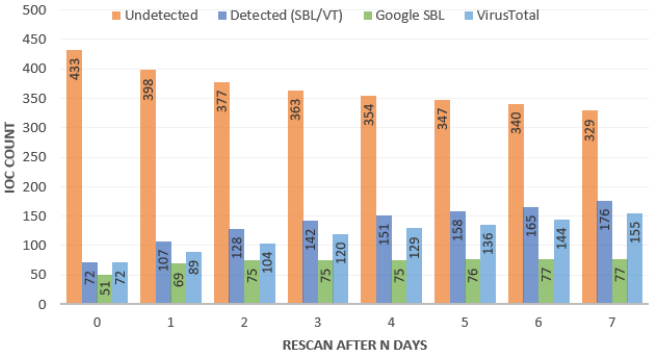

Fig. 6. Daily rescanning of URLs harvested between June 11th and July 8th with Google SBL and VirusTotal for one week after collection

to SBL. Furthermore, we checked these URLs with VirusTotal after seven days to see whether they were identified by any of 60 blacklists on this platform and if yes, when was the scan time. Figure 6 shows the results of rescanning of IoCs detected between June 11th and July 8th. About 10 percents of the URLs were detected by either Google SBL or VirusTotal on the same day of their discovery. After one week from discovery time, the percent of detected URLs has increased to 26 percent. This result suggests that the data extracted from Twitter is fresh and can complement existing blacklists such as Google SBL.

Several researchers, such as [6] and [13], have mentioned that despite the usefulness of considering hashtags for determining the topic of tweets, one should not rely on them as a large percentage of tweets do not contain any tags. Although several researchers, such as [6] and [13], have reported low usage of hashtags by Twitter users, in our preliminary experimentation, we observed that more than 52% of tweets containing IoCs have at least one tag. This suggests that tags can be used to determine the topics of tweets in the CTI domain.

C. CTI Tweet Classifier

To evaluate CTI Tweet Classifier, 75 accounts that are known to share cybersecurity were monitored for a week in

April 2018 and their tweets were collected. 2,300 tweets were collected in this way. It is worth noting that not all of the collected tweets were security-related as the users also tweeted about other matters such as their personal life. The collected tweets were, then, manually labeled as having IoCs or not having IoCs. During this process, the tweets with content not entirely in English were also discarded. To evaluate the classifier, first, it is trained with a set of 200 random tweets and tested on a group of 143 random tweets, all from the same week. The machine classification of this group yielded an accuracy of 97.2%. In the second experiment, the classifier is trained on 200 tweets, but this time, we ensured that 20% of the tweets were labeled as containing an IoC. When testing the 143 tweets once more, the accuracy was, again, around 97%.

VIII. Related Work

In recent years, many research works have been published on identifying credible sources of information and also influential users in terms of propagation of information on social media such as Twitter. Moreover, a few research works are proposed to extract cybersecurity-related information from these platforms. In this section, we briefly review some of the most notable works in this area.

Weng *et al.* [13] observed that 72.4 percent of Twitter users follow more than 80 percent of their followers which they attributed this reciprocity to the phenomenon of homophily seen in many other social networks [8]. This phenomenon suggests that people follow their followers because of the similarity in their topics of interest. Based on this observation, they proposed TwitterRank, which is an extension of PageRank algorithm to measure the influence of users on Twitter. It considers both the topical similarity of users and link structure between them to rank influential users on a specific topic.

Montangero *et al.* [9] proposed a method to identify the most influential twitterers on a specific topic, where the topic is denoted by a hashtag. They proposed the following three indicators to measure the influence of a user on a given topic: followers influence, retweet influence, and favorite influence. To identify influential users on a topic, first, for each candidate user (*i.e.*, the one that tweeted about the topic), they compute these indicators. Then, for each indicator, they create a list of k user with the highest scores. The users appeared in these lists are divided into three group: I. Highly influential users, those who appeared in all the three lists II. Influential users, those who appeared in 2 of them, and III. Potential influential, those who only appeared in one of the top k lists.

Castillo *et al.* [3] proposed a method to determine whether a set of tweets related to a topic are credible. The method relies on a number of features extracted from users' posting, and reposting behaviors, from the content of the tweets, and from external references mentioned in the tweets. The results show that the method can discern between credible and not credible tweets with the precision and recall in the range of 70% to 80%. They concluded that credible news is propagated through users that have previously posted a large number of tweets, originate at a single or a few users in the network, and have many retweets.

Alrubaian *et al.* , in [1], proposed a new approach to determine the credibility of information sources on a specific topic in Twitter, which can be utilized to detect malicious users conducting various malicious activities such as propagation of false or derogatory information on this network. In addition to considering the popularity of twitterers, they consider how sentimental the users are regarding a particular topic to calculate their credibility. To do so, they calculate the ratio of positive tweets to all tweets published by a user. In their experiments, the could achieve 93.4% accuracy at locating users who can be considered credible on a predefined topic.

Ghosh *et al.* , in [5], proposed Cognos, a crowdsourcing search engine for identifying experts on Twitter. To find domain experts, it relies on features extracted from the name and description of Twitter lists. To be more specific, in Cognos, the keywords in the title and description of lists are assigned to their members. These assigned keywords to users are then used to find a topical expert. In Cognos, all lists are treated as equal regardless of their quality and specificity. In our work, we consider other features to assign weights to lists based on their quality and trustworthiness. Moreover, not only we consider user-list relationships, but also we incorporate several user-specific features to identify topical experts more accurately.

In [2], authors proposed a binary classifier based on Support Vector Machine (SVM) to classify spammers and non-spammers on twitter. The proposed classifier relies on 1) 39 features extracted from the textual content of the user's tweets such as the average number of words of each tweet and number of hashtags on each tweet, and 2) 23 features that capture the user behavior in terms of the posting frequency, influence, and social interactions on the Twitter network. The proposed classifier achieved 70% true positive and 96% true negative on a large dataset containing 54 million users.

Sabottke *et al.* [12] examined data published on Twitter for the possibilities of early detection of exploits that are being used in the wild (i.e. before detailed information about a vulnerability officially announced by its vendor). Based on their experimentation, they introduce a set of techniques utilizing supervised machine learning for detecting such exploits.

Zou *et al.* [14] proposed an artificial neural network model of sequence labeling to automatically recognize IoCs from cybersecurity reports. Husari *et al.* [7] presented TTPDrill, which is a system that uses an information-theoretic approach to identify TTPs in cyber threat reports. The main difference between these works and our presented system is that IoCMiner employs a reputation model in conjunction with a classifier to extract IoCs from Twitter, in which the majority of shared information is unrelated to cybersecurity or may not be trustworthy.

IX. Conclusion

In this paper, we presented IoCMiner, which is a scalable framework to extract indicators of compromise from tweet stream posted by users on Twitter. Instead of examining all

published information on this platform to locate indicators of compromise, IoCMiner attempts to first identify credible sources of information that regularly publish cyber threat information, and then analyze information posted by such users to extract the IoCs about ongoing cyber attacks. Through evaluation, we showed that a large percentage of the indicators of compromises discovered by IoCMiner (about 90% of URLs) is not reported by traditional threat intelligence data sources, such as Google safe browsing list and blacklists in VirusTotal, at the time discovery. However, the majority of such unknown IoCs are become discovered and listed over time. In our experiment, the number of not blacklisted URLs reduced to 74% after one week.

REFERENCES

[1] M. Alrubaian, M. Al-Qurishi, M. Al-Rakhami, M. M. Hassan, and A. Alamri, "Reputation-based credibility analysis of twitter social network users: Reputation-Based credibility analysis of twitter social network users," *Concurr. Comput.*, vol. 29, no. 7, p. e3873, Apr. 2017.

[2] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on twitter," in *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, vol. 6. pdfs.semanticscholar.org, 2010, p. 12.

[3] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th international conference on World wide web*. ACM, Mar. 2011, pp. 675–684.

[4] S. D. Forensics and I. R. Blog, "Security intelligence: Attacking the cyber kill chain," https://digital-forensics.sans.org/blog/2009/10/14/security-intelligence-attacking-the-kill-chain, 2009.

[5] S. Ghosh, N. Sharma, F. Benevenuto, N. Ganguly, and K. Gummadi, "Cognos: crowdsourcing search for topic experts in microblogs," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012, pp. 575–590.

[6] A. Hamzehei, S. Jiang, D. Koutra, R. Wong, and F. Chen, "Topic-based social influence measurement for social networks," *Australasian Journal of Information Systems*, vol. 21, no. 0, Nov. 2017.

[7] G. Husari, X. Niu, B. Chu, and E. Al-Shaer, "Using entropy and mutual information to extract threat actions from cyber threat intelligence," in *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, Nov 2018, pp. 1–6.

[8] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociol.*, vol. 27, no. 1, pp. 415–444, Aug. 2001.

[9] M. Montangero and M. Furini, "TRank: Ranking twitter users according to specific topics," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, 2015.

[10] A. Niakanlahiji, L. Safarnejad, R. Harper, and B.-T. Chu, "Iocminer: A framework to automaticly extracting indicators of compromise (iocs) from twitter," https://github.com/aniakanl/IoCMiner, 2019.

[11] I. Rob McMillan Gartner, "Definition: Threat intelligence," https://www.gartner.com/doc/2487216/definition-threat-intelligence, 2013.

[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*. usenix.org, 2015, pp. 1041–1056.

[13] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "TwitterRank: Finding topic-sensitive influential twitterers," in *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, ser. WSDM '10. New York, NY, USA: ACM, 2010, pp. 261–270.

[14] S. Zhou, Z. Long, L. Tan, and H. Guo, "Automatic identification of indicators of compromise using neural-based sequence labelling," *arXiv preprint arXiv:1810.10156*, 2018.