TIMiner: Automatically Extracting and Analyzing Categorized Cyber Threat Intelligence from Social Data

Jun Zhao[1,2], Qiben Yan[3,*], Jianxin Li[1,2,*], Minglai Shao[1,2], Zuti He[1,2], Bo Li[1,2,*]

[1] School of Computer Science and Engineering, Beihang University, Beijing, China
[2] Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China
[3] Computer Science and Engineering, Michigan State University, East Lansing, Michigan, USA
{zhaojun,lijx,shaoml,hezuti,libo,}@act.buaa.edu.cn qyan@msu.edu

Abstract

Security organizations increasingly rely on Cyber Threat Intelligence (CTI) sharing to enhance resilience against cyber threats. However, its effectiveness remains dubious due to two major limitations: first, the existing approaches fail to identify the unseen types of *Indicator of compromise* (IOC); second, they are incapable of automatically generating categorized CTIs with domain tags (e.g., finance, government), which makes CTI sharing ineffective. To combat the challenges, this paper proposes TIMiner, a novel automated framework for CTI extraction and sharing based on social media data. Particularly, an efficient domain recognizer based on convolutional neural network is first implemented to identify CTIs' targeted domain. Then, an indicator of compromise (IOC) extraction approach based on word embedding and syntactic dependence is proposed, which provides the ability to identify unseen types of IOCs. Finally, the extracted IOC and its domain tag are integrated to generate a categorized CTI with specific-domain. TIMiner is capable of generating CTIs with domain tags automatically. With the categorized CTIs, *Threat-Index* is presented to quantify the severity of the threats toward different domains. Experimental results confirm that the proposed CTI domain recognizer and IOC extraction achieve superior performance with the accuracy exceeding 84% and 94%, respectively. Moreover, TIMiner stimulates new insights on the evolution of cyber attacks across multiple domains.

Index Terms

Cyber threat intelligence, IOC, threat index, social media, cyber security.

1 Introduction

Recently, cyber criminals are becoming increasingly sophisticated, and are capable of exploiting zero-day vulnerability and advanced persistent threat (APT) [1], [2]. Evildoers consistently permeate and attack cyber systems to steal sensitive information, take control of the target system, and collect ransom.

Traditional safeguards, such as firewall, signature registry, and intrusion detection system (IDS), hardly prevent these novel attacks [3], [4]. For example, the WannaCry ransomware that was launched on May 2017 spread across 150 countries and infected more than 230,000 computers within one day [5]. To protect systems from such destruction, security experts have proposed *Cyber Threat Intelligence (CTI)* that consists of the *indicator of compromise (IOC)* to release an early warning when a system encounters suspicious threats [6]. CTI consists, e.g., of reasoning, context, mechanism, indicators, implications, and actionable advice about an existing or evolving cyber attack that can be used to create preventive measures in advance [7]. CTI allows subscribers to expand their visibility into the fast-growing threat landscape, and enable early identification and prevention of a cyber threat.

Recently, social media (e.g., Blogs (AlienVault blog, FireEye blog, etc), Vendor bulletins (Microsoft, Cisco, etc), Hack Forums (https://hackforums.net)) have become an effective medium for exchanging and spreading cybersecurity information, on which cybersecurity experts are rushing to share their discoveries [8]. An increasing number of threat-related posts have been published on social media, which often reveal new vulnerabilities, malware, or attack tactics, providing one of the main raw materials for generating cyber threat intelligence [9]. Security vendors have been increasingly extracting IOCs (e.g., malicious IP, malicious URL, malware, etc.) from these first-hand threat descriptions to generate CTIs so as to proactively empower system protection. Take WannaCry [5] as an example, if security guards can capture the threat intelligence that Wannacry permeates port 445 to attack systems in the first place, the malicious intrusion can be easily blocked by locking port 445, which is the most direct and effective way of combating the WannaCry ransomware.

Early CTI extraction requires extensive manual inspection of the threat description, which becomes rather time-consuming given the enormous volume of threat-related descriptions. To facilitate the automatic generation and sharing of cyber threat intelligence, many CTI standards and frameworks are established, such as *IODEF* [10], *STIX* [11], *TAXII* [12], *OpenIOC* [13], and *CyBox* [14]. And most of existing IOC extraction tools follow the OpenIOC standard to extract particular types of IOCs (e.g., malicious IP, malware, file Hash, etc), such as *CleanMX*[1], *PhishTank*[2], *IOC Finder*[3], and *Gartner peer insight*[4], etc. Nevertheless, the existing IOC extraction methods present the first limitation. ***Limitation 1:*** *Most of existing approaches are incapable of identifying unknown types of IOCs, making their effectiveness is doubtful.*

Recently, numerous CTI platforms have emerged, and they indiscriminately share identified threats with subscribers in different domains. However, the threat information is usually quite generic, not shaped to particular domains, making it is ineffective for most domains [1]. In this paper, it is investigated that well-known CTI frameworks (e.g., *IODEF* [10], *STIX* [11], *TAXII* [12], *OpenIOC* [13], and *CyBox* [14]) and platforms (e.g., *IBM X-Force*[5], *Threat crowd*[6], *Opencti.io*[7], *Gartner peer insight*[8], *AlienVault*[9], etc), most of which do not offer domain tagging capabilities. As for the *Gartner peer insight* and *AlienVault*, we carefully analyzed their domain tagging capabilities and found that the domain tags need to be provided **manually** when submitting a new CTI, which becomes rather time-consuming given an enormous volume of threat descriptions. Consequently, the existing frameworks and platforms pose the second limitation. ***Limitation 2:*** *The majority of CTI platforms do not offer the capability of domain tagging for CTI, and they tend to indiscriminately share*

[1] http://list.clean-mx.com

[2] https://www.phishtank.com

[3] https://www.fireeye.com/services/freeware/ioc-finder.html

[4] https://www.gartner.com/reviews/market/security-threat-intelligence-services

[5] https://exchange.xforce.ibmcloud.com/

[6] https://www.threatcrowd.org/

[7] https://demo.opencti.io/

[8] https://www.gartner.com/reviews/market/security-threat-intelligence-services

[9] https://otx.alienvault.com/

CTIs with organizations, most of which are irrelevant with their domains. As a result, extensive manual efforts are required to extract relevant CTIs.

Actually, the optimal threat mitigation period is within 8 hours once a vulnerability or exploit is exposed [15]. With the explosive growth of uncategorized CTIs (without domain tags), if the critical response time is spent on selecting relevant CTIs, the best mitigation time may be missed. Another real-world fact is that most CTI subscribers have limited budget for purchasing cyber threat intelligence and they concentrate on CTIs relevant to their specific domains [16]. To combat the challenges, it is urgent need an automated CTI generation framework, which is capable of identifying unknown types of IOCs, and provides the domain tagging capability for CTIs to ensure them can be personalized sharing to relevant subscribers. In this paper, the task of domain tagging for CTI is formalized as below.

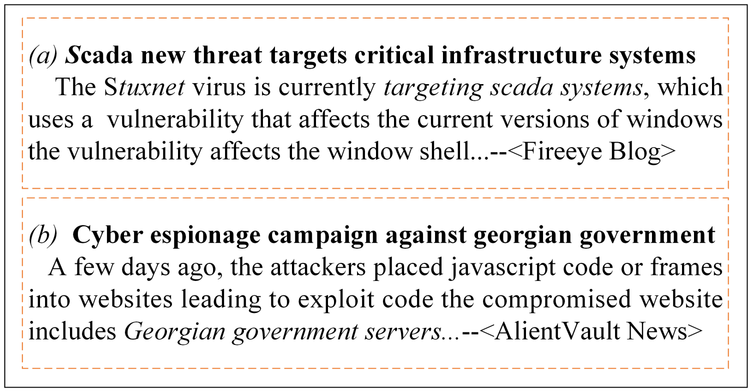
Definition 1. (*Domain tagging for CTI*). Given threat description collection $T = \{t_1, t_2, \cdots, t_n\}$, and domain tag set $D = \{d_1, d_2, \cdots, d_n\}$, the task of domain tagging for CTI is: (i) to assign an appropriate domain tag d_n to a particular threat description t_i based on its semantic characteristics; (ii) to extract IOCs from the threat description t_i leveraging the proposed IOC extraction method; (iii) to merge the domain tag d_n of t_i and extracted IOC from t_i to generate the categorized CTI with domain tag.

In the definition, t_i is a threat description collected from social media sources in TABLE 8, and d_n is the corresponding domain tag for t_i. In this paper, five domains that are most severely threatened are highlighted, including finance, government, education, Internet-of-Things (IoT), and Industrial Control System (ICS).

In fact, categorized CTIs with domain tags can bring the following advantages: first, the categorized CTIs can enable personalized sharing, reducing the burden on subscribers to filter out information that is irrelevant to them; second, the categorized CTIs allow subscribers to focus on the threat information in their own domains and deepen their insight of most relevant threats; third, categorized CTIs make it easier for security experts to demystify the evolutionary trend of different threats in particular domains.

Challenges: Actually, it is a challenging task to label the domain tags for CTIs due to the unclear boundaries between different domains of CTIs. For example, examples (a) and (b) in Fig. 1 may be considered as belonging to the same domain by most people, whereas CTI providers should be able to classify (a) as in ICS domain and (b) as in governmental domain. In fact, it is difficult to distinguish the subtle characteristics of threat descriptions in different domains. Thus, a more intelligent approach is needed, which can learn the more discriminative features between different domains to address the problem of domain tagging for CTIs to enable personalized CTI sharing.

This paper proposes TIMiner, a novel method to automatically extract and evaluate CTIs that contain domain tags. TIMiner includes a convolutional neural network (CNN) based recognizer that automatically identifies domains where CTIs belong to, and a hierarchical IOC extraction method with seamless fusion of word embedding and syntactic dependency, which could identify unseen types of IOCs. TIMiner merges IOCs with their corresponding domain tag to form a comprehensive *domain-specific CTI*. To the best of our knowledge, this is the first study to generate *domain-specific CTIs* that spark numerous novel insights. The main contributions of this paper are summarized as follows:



(a) ***S*****cada new threat targets critical infrastructure systems**

The S*tuxnet* virus is currently *targeting scada systems*, which uses a vulnerability that affects the current versions of windows the vulnerability affects the window shell...--<Fireeye Blog>

(b) **Cyber espionage campaign against georgian government**

A few days ago, the attackers placed javascript code or frames into websites leading to exploit code the compromised website includes *Georgian government servers...*--<AlientVault News>

Fig. 1: Illustration of the challenges of identifying the domain of threat text. (a) depicts *Stuxnet* virus that attacked industrial control system, (b) describes an attack specific to Georgia government.

- Developing an automated CNN based domain recognizer to assign CTI to a corresponding domain that it impacts. More specifically, this paper collects and analyzes more than 50,000 security texts describing threat events, and focuses on five domains that are most seriously threatened, including finance, government, education, IoT (Internet-of-Things), and ICS (Industrial Control System). The experimental result demonstrates that accuracy of the proposed approach exceeds 84%.
- An automated IOC extraction method based on word embedding and syntactic dependency is designed to extract IOCs from threat description texts, which not only guarantees the high accuracy of predefined IOC extraction, but also identifies and extracts unseen types IOCs. Experimental results verify that the proposed method achieves 94% and 92% accuracy and recall, respectively. To date, more than 1,280,000 IOCs have been extracted from unstructured security-related texts.
- This work presents *Threat-Index*, a novel safety assessment criteria, to evaluate the security status of different domains. *Threat-Index* captures the differences of the threat impacts across multiple domains, and quantifies the threat severity for each domain. We analyze the threat trends in multiple industries, and explore the attack characteristics and tactics that hackers undertake to disturb each domain.
- More than 118,000 texts/posts from January 2002 to November 2018 have been analyzed, based on which we gain deep insights into the threat evolution in each domain. The most intriguing insights are summarized below: (i) **DDoS**: all five industries suffer from DDoS attacks, but the attack implementations vary significantly across multiple domains. For instance, an increasing number of botnets are constructed for IoT DDoS, and attackers utilize traffic amplification for financial DDoS attacks. (ii) **Phishing**: phishing attacks often adapt to different forms according to the value of the attack target. For an increasing target value, the phishing patterns evolve from *email phishing* to *speared phishing*, to ultimately the most convoluted *watering hole phishing*. (iii) **Ransomware**: an emergence of a ransomware is often followed by its variants immediately, while some of them will eventually evolve into crypto-mining viruses.

The remainder of this paper is organized as follows: related work is reviewed in section II. Section III describes the overview of our proposed framework. In Section IV, illustrating the proposed method, focusing on how to build domain recognizer and generate domain-specific cyber threat intelligence. In Section V, introduce Threat-Index to quantitatively measure the threat severity targets each domain, and present several protective recommendations. In Section VI, threat evolution in different domains are investigated. Finally, conclude the work in section VII.

2 Related Work

CTI has been regarded as an effective way to proactively withstand novel unseen network attacks [17], [18]. Recently, It has been attracting attention from industry and academia, most security researchers and communities focus on the efficient extraction of IOC (*Indicator of Compromise*) from social media that describes attack events. Initially, IOCs are extracted from famous security knowledge bases, but they only cover a small types of IOCs, it is very difficult to leverage the thin intelligence to defend against attacks. The explosive growth of threat-related social posts provides a steady stream of raw materials for generating CTIs. OpenIOC framework defines more 600 common IOC entities to guide IOC extraction. AlientVault OTX [10], iACE [19] follow OpenIOC suggestion to capture IOCs from threat-related texts. Catakoglu *et al.* [20] developed a system to extact IOCs from web pages. Sabottke *et al.* [21] established a tool to detect potential vulnerability from tweets. Jamalpur *et al.* [22] utilized dynamic analysis to detect malware in a Cuckoo sandbox environment. Ebrahimi *et al.* [23] applied deep convonlutional neural network to capture malicious conversations in social media. Isuf Deliu *et al.* [24] explored machine learning method to rapidly sift specific IOCs in hacker forums. However, the existing methods and tools only recognize and extract predefined types of IOCs. Furthermore, there is a lack of solutions to associate such uncategorized IOCs with relevant organizations. These limitations have weakened the applicability and effectiveness of CTI sharing for cyber defense.

Recent works focus on formulating the taxonomy of cyber threat intelligence. Ahrend *et al.* [25] divide CTI into formal and informal practices to uncover and utilize tacit knowledge between collaborators. Hugh *et al.* [26] categorize CTIs into strategic and operational ones. Ray [27] partitioned IOCs into three distinct categories: network, host-based, and email IOCs. To the best of our knowledge, there is no method or framework for generating domain-specific cyber threat intelligence and delivering them to relevant organizations.

Recently, numerous CTI platforms and products have emerged, they share the threats (e.g., new malwares, spreading viruses, latest vulnerabilities, etc) with subscribers in different domains. However, the information is usually quite generic, not shaped to specific domains, which makes it ineffective [1]. One study [16] argued that CTI community should standardize data labeling to ensure security experts can then assess whether the data fits their needs. Moreover, according to the survey [28], 66% of respondents complain that the uncategorized CTIs are insufficient in perceiving suspicious cyber threats. Globally, domain-specific information sharing is required, and the need is growing. For example, the financial sector (*FS-ISAC*)[11], the retail sector (*R-CISC*)[12],

[10] https://otx.alienvault.com/
[11] http://www.fsisac.com
[12] http://r-cisc.org

the electricity sector (*E-ISAC*)[13], and the recently established automotive sector (*AUTO-ISAC*)[14], these sectors generally share intelligence in a manual and supervised manner [29]. The vertical domain-specific information sharing platforms can ensure that the most relevant information of the domain is shared between organizations. However, in the CTI field, all the popular CTI platforms and standards including *IODEF*, *STIX*, *TAXII*, *OpenIOC* and CyBox *can neither automatically generate the domain-specific CTIs that contains domain labels, nor share CTIs with relevant organizations that are interested in them*. Therefore, it is of great significance to generate and share CTI with domain tag (domain-specific CTI). In this paper, TIMiner, a novel CTI extraction and sharing framework, is proposed. TIMiner can generate CTIs with domain tags and allows CTIs can be personalized sharing to relevant subscribers.

3 Framework Overview

TIMiner consists of five major components as shown in Fig. 2, the details of which are presented below.

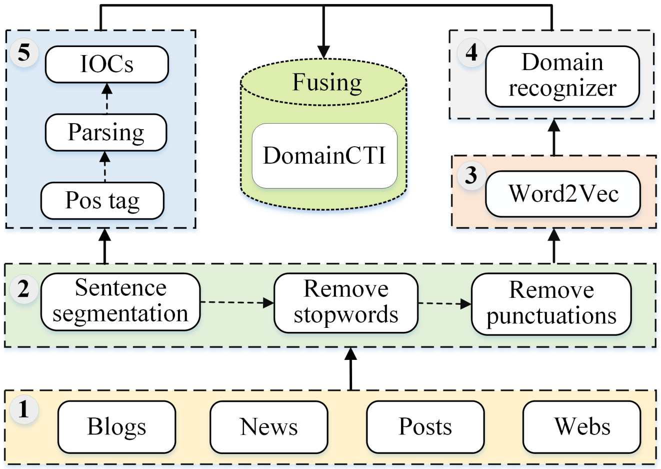


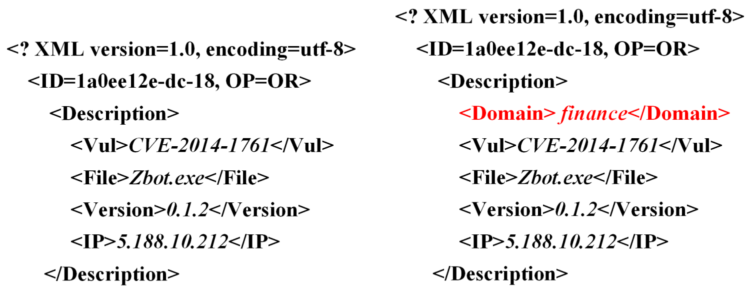
Fig. 2: The architecture of TIMiner: (1) Data collection module to collect security-related social texts automatically. (2) Prepossessing module focuses on segmenting sentences, removing stop words and punctuation. (3) Word-embedding module expresses the preprocessed texts into a low dimensional vector space. (4) leveraging the word-embedding of each threat text as input to train a domain recognizer to classify the threat intelligence into corresponding domains. (5) extract the IOC from threat texts (step 2), and embed its domain tag (step 4) to generate a complete domain-specific cyber threat intelligence.

- **Threat-related data collection**. *TI_spider*, an automated data collection system, is developed, which collects threat-related data from different social media including blogs, hacking forum posts, security news, security vendor bulletins, etc. Specifically, *TI_spider* consists of 75 independent distributed crawlers, each of which monitors and collects a

[13] http://www.eisac.com
[14] http://www.automotiveisac.com

specific data source in TABLE 8. Each crawler utilizes breadth-first search to collect threat descriptions, which starts the collection from a homepage describing threat events until no new link can be invoked. For each link, the HTML source codes are first crawled, and then to extract threat event data leveraging Xpath (*XML Path language*).

- **Data preprocessing**. The data preprocessing removes all punctuations, stopwords, and markup characters using Stanford *CoreNLP*[15]. Data preprocessing not only reduces the dimension of each text, but also mitigates the noisy features in word embedding.
- **Word embedding**. Word embedding converts natural language texts into the latent vector space. In this paper, a word2vec model [30] specific to representing threat descriptions is trained, which can effectively capture the interdependent relationships over words. The embedding_dim is to 200, which means that each word in threat descriptions is represented by a 200-dimension vector.
- **Recognition of CTI's domain**. Recognizing the domain of CTI is the necessary precursor for constructing domain-specific CTIs. The framework of CTI domain recognizer is presented in Fig. 4, in which leveraging 256 filters with kernel=5 to learn the local features of each threat description, and then splicing the pooled feature vectors into a fully connected layer. Finally, utilizing soft-max activation function to calculate the probability of each domain tag of CTI.
- **Domain-specific CTI generation**. This module generates domain-specific CTIs with domain tags. First, an IOC extraction tool based on word embedding and syntactic dependency is developed to extract IOCs, which can effectively identify unknown IOCs that are not recorded in OpenIOC [13]. Then, combining the IOC and its domain tag to generate a categorized *domain-specific CTI*, an example of which is illustrated in Fig. 3 (b).



```
<? XML version=1.0, encoding=utf-8>
  <ID=1a0ee12e-dc-18, OP=OR>
    <Description>
      <Vul>CVE-2014-1761</Vul>
      <File>Zbot.exe</File>
      <Version>0.1.2</Version>
      <IP>5.188.10.212</IP>
    </Description>
```

(a) Traditional CTI.

```
<? XML version=1.0, encoding=utf-8>
  <ID=1a0ee12e-dc-18, OP=OR>
    <Description>
      <Domain> finance</Domain>
      <Vul>CVE-2014-1761</Vul>
      <File>Zbot.exe</File>
      <Version>0.1.2</Version>
      <IP>5.188.10.212</IP>
    </Description>
```

(b) *Domain-specific CTI.*

Fig. 3: (a) and (b) are extracted from the same cyber threat description depicting financial threat event. Comparing with (a), (b) can be personalized sharing to finance-related organizations since it is clearly labeled as ”finance” domain.

4 Proposed Method

This section illustrates the design of TIMiner. First, introducing the convolutional neural network (CNN) based recognizer to identify which domain a cyber threat intelligence belongs to.

[15] https://stanfordnlp.github.io/CoreNLP/

Then, describing the proposed hierarchical IOC extraction method, which can not only accurately extract predefined IOCs but also effectively capture unknown IOCs that not enrolled in OpenIOC [13].

4.1 CTI's Domain Identification

4.1.1 Domain Recognizer

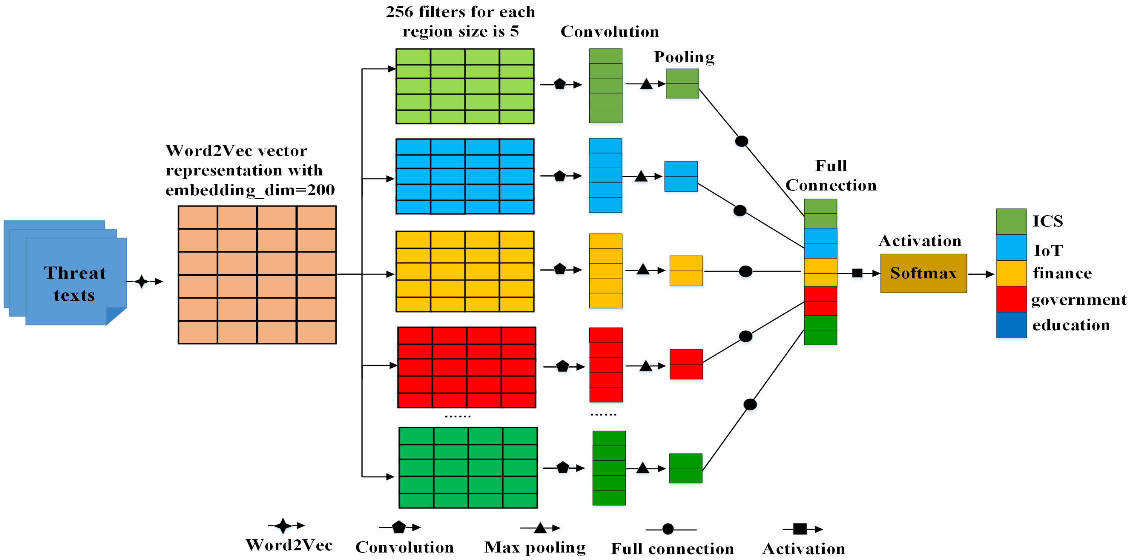


Fig. 4: The overview of CTI domain recognizer

This paper implements a CTI domain recognizer based on a variant of CNN model [31], the architecture of which is presented in Fig. 4. The main process is illustrated in Algorithm 1.

Algorithm 1 Constructing CTI Recognizer

```
Require: Threat event descriptions $\mathcal{T} = \{t_1, t_2, \cdots, t_n\}$.
Ensure: Domain Tag $\hat{y}_i$.
for each $t_i \in \mathcal{T}$ do
    words ← $preprocessing$ ($t_i$)
    word_vector ← $word2vec$ (words)
    for each epoch do
        local_features ← $convolution$ (word_vector)
        max_features ← $maxpooling$ (local_features)
        merge_feature ← $connecting$ (max_features)
        $\hat{y}_i \leftarrow max(softmax($merge_feature$))$
        $L(y_i, \hat{y}_i) \leftarrow - \bigtriangledown \sum_{i \in N} y_i \cdot log\hat{y}_i$
    end for
end for
```

Word representation is one of the most fundamental task in natural language processing (NLP). One-hot encoding and distributed word representation are popular approaches used in text classification and sentiment analysis, however, they often result in inferior word embedding as ignoring the interactive relation among words. In this paper, a word2vec model [30] specific to threat description embedding is trained, which takes a large corpus of threat descriptions as its input and produces a low-dimensional vector, with each unique word in the corpus being assigned a corresponding vector in the latent space. Formally, a word embedding **E**: *word*$\rightarrow \mathbb{R}^n$ is a parameterized function that maps words in natural language to latent vector space. For instance, word "attacker" is embedded in a vector:

$$Embedding\ (``attacker") = (-3.399, -4.462, 3.136, ...)$$

The convolution operation applies a filter $w \in \mathbb{R}^{h \times d}$ to a window of h words to generate a new feature marked as f. Then, the max pooling operation runs over the feature map and takes the maximum $F = max\{f\}$, which captures the most important feature with the highest value for each feature map. In addition, word2vec arranges the vector space so that the words with similar contexts in the corpus are located in close proximity with one another, which allows our model to capture the interdependent relationships between words. With the learned word embedding of each threat description, the convolutional operation can be conducted to learn the features of CTIs in different domains.

$$\hat{y} = max(softmax(\sigma(X \cdot W + b))) \tag{1}$$

where $X = [x_1, x_2, \cdots, x_i]$ is the word embedding of each threat description, $W = [w_1, w_2, \cdots, w_i]$ denotes the weights of words for identifying the domain of a threat description, b is a bias vector that captures all other factors which influence $\hat{y}$ other than the X, and $\sigma(\cdot)$ represents an activation function, such as $relu$.

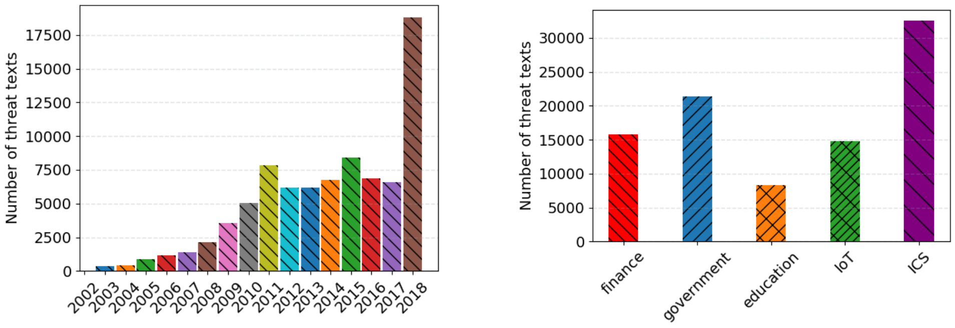
Domain recognizer adopts cross-entropy as the loss function, and leverages stochastic gradient descent to minimize the loss function $L(y_i, \hat{y_i})$.

$$L(y_i, \hat{y_i}) = -\sum_{i \in N} y_i \cdot log\hat{y_i}, \tag{2}$$

where y_i is the real domain tag of threat text i, and $\hat{y_i}$ is the corresponding predicted domain tag.

4.1.2 Performance Evaluation

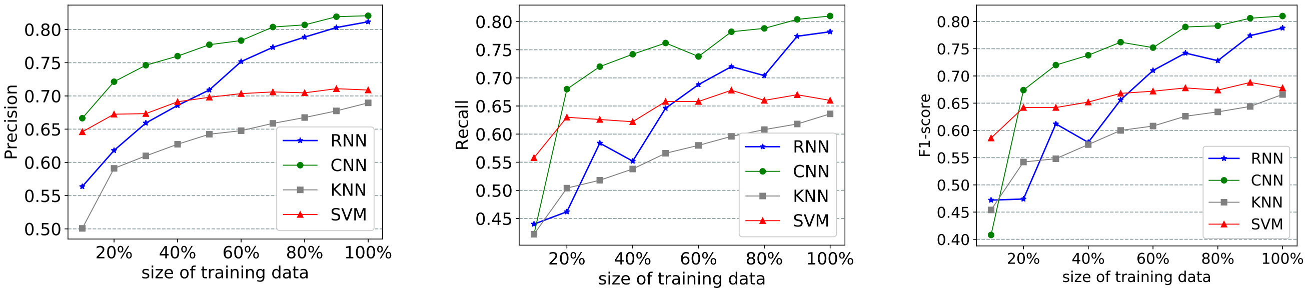
Datasets. *TI-spider*, an automated data collection tool, is developed to persistently collect threat description data that portrays cyber threat events. *TI-spider* monitors 75 threat-related data sources including security blogs (*AlienVault, FireEye, Webroot, etc*), security vendor bulletin (*Microsoft, Cisco, Kaspersky, etc*) and the posts published in hacking forums (*Webroot, HackerForum, etc*). The data sources are listed in TABLE 8 in Appendix. So far, *TI-spider* has collected more than 118,000 threat-related descriptions over the past 16 years from January 2002 to November 2018. The overall threat text statistics is demonstrated in Fig. 5 (a), and Fig. 5 (b) depicts the distribution of the domain-specific documents. Actually, in order to train and evaluate our proposed method, five cybersecurity researchers (three PhDs and two Masters) spent efforts (about fortnight) to manually label the collected data. Particularly, the five researchers independently labeled the collected threat descriptions leveraging the domain tags including education, finance, government,



(a) The number of threat texts per year.

(b) Threat text distribution statistics.

Fig. 5: Statistics of collected security-related texts.



(a) Precision of different methods

(b) Recall of different methods

(c) F1-score of different methods

Fig. 6: Performance of different recognition methods.

IoT, and ICS. To ensure the accuracy of data labeling, we test the consistency of the tags labeled by five researchers for each piece of data and remove the data with ambiguous tags. In other words, the final dataset consists of data consistently labeled by the five researchers, which constitutes a valid source of ground truth. As a result, we generate a final dataset with 15,000 labelled threat descriptions equally covering five domains. For the labeled data, 70% of them are used as training data to train our proposed model, another 20% of them to evaluate the model, and the rest for testing the model.

To evaluate the performance of CTI domain recognizer, the proposed method is compared against three popular classification algorithms including support vector machine (SVM), K-nearest neighbors (KNN), and recurrent neural network (RNN). The model parameters are fine-tuned after training 3000 epochs, and the optimal parameters are recorded in TABLE 1.

TABLE 1: The major parameters of CTI domain recognizer.

Parameter	Value
Embeddding_dim	200
Sequence_length	1000
Number_class	5
Number_filters	256
Vocab_size	56170
Hidden_dim	128
Dropout_rate	0.5
Learning_rate	0.001
Batch_size	64

Where, *Embedding_dim* represents the dimension of vector for expressing each word, *Sequence_length* stipulates that each text is represented by 1,000 words, thus, each text can be represented as a $seq\_length \times embedding\_dim$ matrix. In our model, each threat description is represented by a $1,000 \times 200$ matrix. The threat descriptions with less than 1,000 words are padded with "0". *Num_filters* denotes the number of convolutional filters, *vocab_size* is the total number of words that can be covered by the model, and *hidden_dim* indicates the number of neurons in hidden layer.

Results. As shown in Fig. 6 (a), KNN and SVM achieve 68% and 71% of recognition precision, respectively. A deeper inspection into the training data exposes that the boundaries of threat descriptions illustrating attacks in different domains are unclear. For example, for two attack events targeting IoT and ICS domains, the descriptions of "*sykipot* virus will hijack windows smart devices" and "*Stuxnet* is targeting *SCADA* systems" produce word vectors that resemble each other as computed by word2vec [30]. As a result, KNN and SVM fail to detect such subtle differences, resulting in an unsatisfactory precision.

In contrast, RNN and CNN achieve a much higher recognition precision as shown in Fig. 6(a). The performance of CNN outperforms that of RNN with a classification precision of 84% utilizing all training data. Generally speaking, RNN performs better than CNN for the tasks of translation and question-and-answer (Q&A), which should integrate contextual information in a complex text or a dialogue [32], while CNN often excels in tasks that do not require a long-term memory [31].

There are two major reasons to choose CNN over RNN for constructing the CTI domain recognizer. First, the experimental results confirm that CNN achieves the best recognition result with a simpler architecture than that of RNN. Second, CNN occupies significantly less computing resources than RNN. The execution time of the four comparing approaches on 15,000 samples is presented in Fig. 7. Specifically, with the same running environment (i.e., Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz, 16 GB RAM, 4 Cores), the model training time of RNN (1,260 minutes) is more than 21 times that of CNN (57 minutes).

4.2 Domain-specific CTI Generation

This section aims to address the challenge of extracting IOC from threat descriptions. Existing studies have been extracting useful information from technology blogs and web applications [6],

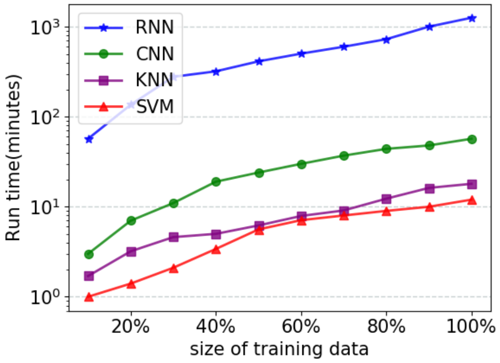


Fig. 7: Performance comparison of execution time with four methods.

[19], [33]. However, they cannot identify unknown types of IOCs that are not enrolled in OpenIOC list, and they do not provide the capability of domain tagging for CTIs. As mentioned before, CTI subscribers wish to receive valuable CTIs related to their domains. This section illustrates the detailed design of our proposed domain-specific CTIs, which consists of two major components as described below.

4.2.1 Identifying IOC Candidates

In this section, a hierarchical IOC extraction method is presented. Different from existing work, the proposed IOC extraction method can effectively recognize the unknown types of IOCs . The process of recognizing IOCs can be divided into three steps.

(step i) Regular expression matching. For the IOCs such as hash code and malicious DNS, it is difficult for traditional natural language processing tools (e.g., *NLTK*, *LTP*) to recognize them. Fortunately, most of them present certain structures, such as malicious IP (*xxx.xxx.21.30*), vulnerability number (*CVE-xxxx-xxxx*), which can be effectively identified by regular expression. Some regular expression samples for recognizing IOCs are demonstrated in TABLE 2.

TABLE 2: Regular expression samples of recognizing IOC.

IOC TYPE	**Regular Expression**
CVE	CVE-[0-9]{4}-[0-9]{4,6}
MD5	[a-f 0-9]{32}|[A-F 0-9]{32}
SHA1	[a-f 0-9]{40}|[A-F 0-9]{40}
Email	[a-z][_a-z0-9_]+[a-z0-9]+.[a-z]
Register	[HKLM|HKCU]\\\\[A-F 0-9]{40}
IP	$\backslash d\{1,3\}.\backslash d\{1,3\}.\backslash d\{1,3\}.\backslash d\{1,3\}$

(step ii) Deep recognition. Named Entity Recognition (*NER*) has been extensively studied in the NLP community. However, the existing *NER* tools (e.g., *CoreNLP, NLTK, PyLTP*) cannot be directly

applied for identifying IOCs, since they have been regarded as brittle and highly field-specific, and the models designed for one field hardly works on another fields. BiLSTM+CRF model [34], on the other hand, can leverage both past and future features by virtue of a bidirectional LSTM component, thereby producing a higher precision in text chunking and *NER*. As a result, this paper implements an efficient tool based on BiLSTM+CRF to recognize IOCs that cannot be matched using regular expressions.

(step iii) Novel IOC expansion. Combining *regular expression matching (step i)* and *deep recognition (step ii)* based IOC extraction methods, it is able to extract all types of IOC registered in OpenIOC. However, the effectiveness of such method is questionable as there are an increasing number of unknown or novel IOCs. Therefore, this step concentrates on identifying the unknown IOCs. For example, for such words as “Maze” ,“AnteFrigus” and “PureLocker”, it is hard to imagine that they would be closely linked to “WannaCry”, a destructive ransomware. As a result, we need a word embedding method, which allows similar words to be closer to each other and find unknown words with similar meanings when we search for a word in its embedded vector space. Recently, Google research team released word2vec [30], an effective word representation method, which goes beyond simple syntactic regularities and allows simple algebraic operation in embedded vector space, e.g., *“queen”-“woman’+“man”=“king”*.

Inspired by word2vec, a word embedding model for threat intelligence is developed to identify unseen IOCs. The word embedding model converts words into the latent vector space to compare the similarities over words. Particularly, all preprocessed threat texts without stopwords and punctuations are first aggregated in the words set, and transformed into a latent vector space. Then, selecting the *Top 5* most similar words to each IOC identified by step (i) and step (ii) to be its IOC extensions, which greatly increases the coverage of IOCs. For example, the word vector of *“Maze”* ,*“AnteFrigus”*,*“Buran”*, *“PureLocker”*, and *“Dharma”* are most similar to that of *“WannaCry”*, thus they can be regarded as the extension of *“WannaCry”*. Thus, for each threat description, it is capable of obtaining an IOC collection that consists of all suspicious IOCs, denoted as $IOC_{candidate}$.

TABLE 3: The performance comparison of different threat entity recognition methods.

NER Tool	Precision	Recall	F1-score
Alien OTX	0.72	0.74	0.73
Stanford NER	0.68	0.47	0.56
NLTK NER	0.65	0.52	0.58
iACE	0.92	0.87	0.89
Hierarchical IOC	**0.94**	**0.92**	**0.93**

Compared with other recognition tools such as *Stanford NER*, *NLTK NER*, *AlientVault OTX* and *iACE* [19], our proposed IOC extraction approach demonstrates better performance in terms of precision and coverage. As shown in TABLE 3, *Stanford NER* and *NLTK NER* perform the worse when dealing with threat intelligence. *Alien OTX* mainly leverages regular matching to extract IOCs, and its precision is low. *iACE* can effectively detect type-specific IOC from technological contents. However, it cannot identify novel types of IOCs that are not enrolled in OpenIOC [13], and thus, it achieves a lower recall than our proposed method.

TABLE 4: The original trigger set of different threat events.

Category	Trigger verbs
DDoS	scan, attack, invade, access, amplify, destroy, block, jam, cripple
Phishing	phishing, cheat, send, entice, trust, inform, notice, steal, filch, capture, catch
Ransomware	ransom, encrypt, lock, access, close, interdict, demand, claim, pay
APT	monitor, detect, probe, exploit, pretend, disguise, hide, conceal
Malware	download, install, exploit, damage, affect, break

4.2.2 Extracting Domain-specific CTI

In order to reduce the false positive (i.e, a legal entity is considered as an IOC) of IOCs extraction, This paper implements an unsupervised syntactic-dependence based IOC extraction method. More specifically, most of trigger verbs (e.g., attack, permeate, invade, block, etc.) describing threatening actions often appear in intrusion descriptions, and IOCs are often syntactically dependent on them. For instance, in the description: "*WannaCry attacked Korea's telecommunication system in May 2017*", the verb "*attacked*" can be regarded as a trigger verb that describes a threatening action, which subsequently forms a subject-predicate relationship with "WannaCry". To extract the entities most relevant to the attack event, we only need to detect the suspicious IOCs with an explicit syntactic dependency (e.g., subject-predicate, verb-object, etc.) to the trigger verbs, which is the most efficient and direct method to reduce the false positive of IOC extraction. Particularly, the most intuitive verbs that describe the threat events are inserted in the *VerbSet*. Then, utilizing the learned word representation in *step iii* to automatically supplement the *VerbSet* by comparing the similarity of word vectors. The original set of trigger verbs describing different types of threat is listed in TABLE 4.

The ultimate goal of this paper is to generate domain-specific CTI with domain tags. Given a threat description set $T = \{t_1, t_2, ..., t_i\}(1 \leq i \leq n)$, threat trigger verbs for t_i $VerbSet = \{v_1, v_2, ..., v_i\}(1 \leq i \leq n)$, and candidate entity set $IOC_{candidate} = \{ioc_1, ioc_2, ..., ioc_i\}(1 \leq i \leq n)$. For each domain-specific threat text t_i, extract ioc_i that has explicit semantic relationship with v_i, and integrate all ioc in text t_i and the domain label of t_i (derived from Algorithm 1) to form a complete *domain-specific CTI*. The complete CTI extraction process is demonstrated in Algorithm 2.

Compared with traditional CTI, domain-specific CTI not only mitigates the false positive of IOC extraction, but also empowers the platforms to share CTIs with relevant organizations and eliminates the burden of security officers in manually filtering unrelated threat intelligence. In addition, domain-specific CTI can assist security organization in deriving new insights about threat trend across different domains, which will be described in the following section.

Algorithm 2 Extracting *domain-specific CTI*

```
Require: Threat Descriptions $\mathcal{T} = \{t_1, t_2, \cdots, t_i\}$.
         Domain Tags $\mathcal{D} = \{education, government, finance, IoT, ICS\}$.
Ensure: Domain-specific CTI $\mathcal{C}_i$.
for each $t_i \in \mathcal{T}$ do
    $t_{D_i}$ ←labeling $t_i \subset D_i$ using CTI domain recognizer in Algorithm 1.
    for each $t_{D_i}$ do
        VerbSet ← scanning trigger verbs.
        $IOC_{candidate}$ ← detecting suspicious IOCs using hierarchical IOC.
        for $v_i$ in VerbSet do
            for $ioc_i$ in $IOC_{candidate}$ do
                if $ioc_i$ and $v_i$ have syntactic dependencies.
                $RealIOC_i$ ← inserting $ioc_i$
            end for
        end for
        $\mathcal{C}_i$ ← Integrating $RealIOC_i$ and $D_i$ as domain-specific CTI.
    end for
end for
```

5 Threat Index

With the learned domain-specific CTI, it can evaluate the threat impact severity caused by different types of attack in each domain. Threat-Index, a novel metric that quantitatively measures the threat severity from the perspective of security-related social opinion, is proposed. By examining the threat descriptions, it is discovered that cyber attacks that cause catastrophic damage to a domain often exploit multiple vulnerabilities, most of which are labeled as high-risk vulnerabilities by *CVE Details*[16]. On the contrary, intrusions using a single and light-risk vulnerability hardly cause fatal damage to a company. This fact enlightens us to quantitatively evaluate the risks of different threats towards each domain. Threat-Index follows three empirical intuitions: (i) the more frequently the domain is attacked, the greater the threat it faces; (ii) the more vulnerabilities exploited in the attack, the greater the harm is towards the system; (iii) the higher the severity level of vulnerabilities is, the more significant their impacts are towards the industry. As a result, the threats can be quantified by exploring the frequency of attacks, the number of exploited vulnerabilities, and the compromised level of vulnerabilities in each domain.

Definition 2 *(Threat-Index).* Given a threat description collection with domain tags $T = \{t_{d_1}, t_{d_2}, \cdots, t_{d_i}\}$ $(1 \leq i \leq n)$, attack types $A = \{a_1, a_2, ..., a_j\}$ $(1 \leq j \leq n)$, and the domain tags $D = \{d_1, d_2, ..., d_k\}$ $(1 \leq k \leq n)$, Threat-Index quantifies the threat impact of attack type a_i towards a domain d_i by analyzing the CTIs in threat description t_i, which can be further divided into *Impact severity index* and *Domain-normalized impact severity index*.

Specifically, in Definition 2, A represents five attack types concerned in this paper, including *DDoS, Malware, Phishing, Ransom* and *APT*. D consists of five domains: ICS, IoT, finance, education

[16] https://www.cvedetails.com

and government. Each threat description text t_{d_k} corresponds to an attack type a_j and its impacted domain d_k.

Definition 3 *(Impact severity index).* Given a threat description sequence $T = \{t_{d_1}, t_{d_2}, ..., t_{d_i}\}$ $(1 \le i \le n)$, an attack type set $A = \{a_1, a_2, ..., a_j\}$ $(1 \le j \le n)$, a domain tag set $D = \{d_1, d_2, ..., d_k\}$ $(1 \le k \le n)$, and a vulnerability set $C = \{c_1, c_2, ..., c_m\}$ $(1 \le m \le n)$. *Impact severity index* computes the threat severity for each domain d_k under attack type a_j as follows:

$$H_{a_j,d_k} = \alpha \sum_{a_j} t_{d_k} + (1-\alpha) \sum_{c_m \in t_{d_k}} R_{c_m} \tag{3}$$

where α is risk weight coefficient, t_{d_k} is threat description depicting domain d_k being attacked by attack a_j, $\sum\limits_{a_j} t_{d_k}$ represents the total frequency of domain d_k being targeted by attack a_j, $\sum R_{c_m}$ calculates total risk score of vulnerabilities included in t_{d_k}, R_{c_m} is the risk score of vulnerability c_m assessed by *CVE details*, and each t_{d_k} contains some vulnerabilities c_m ($R_{c_m} = 0$ means that the attack does not use any vulnerability registered in *CVE details*).

Compared with attack frequency, the risk score of vulnerabilities exploited in a threat can better reflect the severity of attacks, thus α is set to 0.4. *Impact severity index* concentrates on quantifying the impact severity of a particular type of attacks on different domains. Take the **APT** attack in TABLE 5 as an example, its impact on IoT, ICS, education, finance and governmental domain is 0.05, 7.57, 0.58, 2.82, and 9.89 (*see the first row in TABLE 5*) respectively, which reveals the fact that APT attack has the most serous impact on governmental domain.

TABLE 5: Impact severity index of different domains.

Domain / Type	IoT	ICS	education	finance	government
APT	0.05	7.57	0.58	2.82	9.89
DDoS	2.50	58.40	0.32	5.13	29.15
Malware	0.67	44.64	0.46	9.42	32.26
Phishing	0.02	2.81	0.10	1.96	6.21
Ransom	0.33	2.55	0.06	4.10	2.35

Impact severity index analysis results. TABLE 5 demonstrates the severity of the same type of attacks for different domains. The results indicate that the *ICS* industry and the government have experienced the highest threat impact severity among all five industries. In particular, *DDoS* and *malware* threats have incurred the most serious impacts on these two domains. Specifically, the impact severity indices of *DDoS* to *ICS* and government are 58.40 and 29.15 respectively. The *malware* impact severity indices to *ICS* and government are 44.64 and 32.26 respectively. Meanwhile, sophisticated *APT* attackers seem to be aiming at breaking into the government agencies, and they infiltrate the systems and hibernate for months or even years to find the right targets to breach sensitive political messages. In recent years, as more and more *ICS* are connected to the Internet, many high-value *ICS* devices and systems are exposed to the evildoers on the Internet. *ICS* has in fact become the preferred target of *DDoS* attacks. Moreover, hackers often launch catastrophic ransomware attacks towards the financial domain, which often take advantage of the virtual currency such as bitcoin.

Definition 4 *(Domain-normalized impact severity index).* Given a *Impact severity index* H_{a_j,f_k} and the average threat index of domain d_k being targeted by N types of attacks. *Domain-normalized impact severity index* assesses which attack type a_i induces the most severe impact towards domain d_k as follows:

$$V_{a_j,d_k} = \frac{H_{a_j,d_k}}{Average_{d_k}}, \tag{4}$$

$$Average_{d_k} = \frac{1}{N} \sum_{a_j \in A} t_{d_k}, \tag{5}$$

where H_{a_j,d_k} represents the impact severity index of the attack type of a_j towards domain d_k, $Average_{d_k}$ is the average threat index of domain d_k being targeted by N types of attacks, $\sum_{a_j \in A} t_{d_k}$ is a collection of threat description indicating that attack type a_j affects domain d_k, and N records the number of different attack types that threaten domain d_k.

TABLE 6: Domain-normalized impact severity index under different types of attacks

Domain / Type	IoT	ICS	education	finance	government
APT	0.70	0.33	0.29	0.61	2.02
DDoS	3.50	2.52	1.60	1.10	0.62
Malware	0.94	1.93	2.32	1.01	1.83
Phishing	0.03	0.12	0.49	1.46	0.39
Ransom	0.46	0.11	0.30	1.48	0.15

Domain-normalized impact severity index analysis results. Domain-normalized impact severity index concentrates on evaluating the normalized severity level of each attack type for a specific domain, which is able to reflect the threat proportion of different types of attacks in a particular domain. TABLE 6 illustrates the normalized severity of five attack types for each domain. *DDoS* attacks constitute the most prominent threat to the IoT domain as the largest Threat-index (**3.50**) in the column of “IoT” is associated with “DDoS” (see the first column in TABLE 6). In light of Mirai attack, a possible explanation for this trend is that IoT devices such as cameras and sensors have become increasingly popular, but most of which have low security standard. Meanwhile, for government, *APT* attack is the most popular attack type, which requires more specialized attack techniques compared to other types of attacks, and costs more energy and resources. Such attacks are often initiated by hackers with advanced intrusion techniques, whose purpose is not to damage the system but to steal important confidential files in the system. The high stake involved in government documents makes the government an obvious target for *APT* attackers.

It is worth-noting that the threat indices of phishing attack and ransomware attack are very close in the financial industry. It is doubtful whether attackers often integrate these two types of attack methods into one tool to invade financial devices and systems, and the suspicion is proved by checking a large quantity of threat descriptions for financial domain.

TABLE 7: Well-known security vendors and their security products. ("✓" indicates that the manufacturer provides this type of product, and "-" means that there is no corresponding type of product.)

Company	DDoS	APT	Phishing	Ransomware	Trojan
Cisco	✓	✓	✓	✓	✓
Microsoft	✓	-	✓	-	✓
Symantec	✓	✓	✓	✓	✓
Mcafee	✓	-	✓	✓	✓
Raytheon	✓	-	✓	✓	✓
IBM	✓	-	-	✓	✓
HPE	✓	-	✓	-	✓
Checkpoint	✓	-	-	-	✓
Palo Alto	✓	-	✓	✓	✓
Oracle	✓	-	-	✓	✓
Splunk	✓	-	✓	✓	✓
Kaspersky	✓	✓	✓	✓	✓
Palantir	-	-	-	✓	✓
Synopsys	-	-	-	✓	✓
Huawei	✓	-	✓	✓	✓
FireEye	✓	✓	✓	✓	✓
BAE	-	-	✓	✓	✓
BT	-	-	-	✓	✓
SonicWall	✓	-	✓	✓	✓
Cloudflare	✓	✓	✓	✓	✓

Protection guidelines. The *Threat-Index* not only quantitatively evaluates the severity of threats for each domain, but it also sheds light on security protections. Here, we further explore if the existing security products can offer sufficient protections to alleviate the threat impact severity for certain industries. In other words, it is desired to know whether current security products can meet the needs that protect cyber system in different domains from malicious intrusion. The understanding of existing security product landscape is crucial for designing the next-generation security protection products. The products from 20 well-know security vendors(e.g., *Cisco, Symantec, Kaspersky, etc.*) are studied, and their major protection products are listed in TABLE 7.

Based on data analysis, the frequency and intensity of attacks against ICS and government are the highest among all the domains. Therefore, security vendors should put more efforts in addressing the attacks towards these two domains and develop advanced protection products. For ransomware attack, the financial industry has been more severely targeted compared with other domains. However, as illustrated in TABLE 7, there are too few products to protect against ransomware attack to meet the current protection needs of the financial industry. In fact, intelligent ransomware defense tools are needed urgently. As can be seen from TABLE 5, government agencies should be most concerned with APT attacks. Nevertheless, only five security enterprises claim to provide security products against APT attacks. Moreover, although most vendors claim they have the ability to protect against phishing attacks, the descriptions of product designs and tools reveal that most of them are only capable of protecting against low-level phishing attack, while none is specialized in defending advanced spear-attacks and watering hole attacks.

Only a quarter of security companies protect against all types of attacks, while most vendors can only professionally defend against one or two types of attacks, indicating there is a huge gap between cyber attack and cyber defense. Although every domain has experienced a growing number of novel attacks, most security organizations are not well grounded to conquer these unknown cyber attacks. Meanwhile, most of the security products are generic ones, which are not designed for specific attack types for particular domains. However, even the same type of attacks often present different implementations and behaviors when targeting different domains. Actually, domain-specific security products developed for attacks targeting different domain are crucial to protect these diverse systems against infringement.

Recommendation. Security vendors should carefully study the details of each type of attack in different domains to design and develop more specialized and targeted protection strategies. One promising direction is to model the attack behavior characteristics of each attack type towards a specific domain, and use machine learning models to create more effective targeted protection mechanisms.

6 Discussion of Threat Trend

6.1 Evolution of Different Attack Types

One of the key contributions of this manuscript is to propose a novel method that can produce cyber threat intelligence (CTI) with domain tags. As a result, all CTIs can be grouped into corresponding CTI classes based on their domain tags, which can help effectively demystify the trend of threat evolution in a specific domain. Moreover, the categorized CTIs can be personalized to serve CTI subscribers, which allow them to focus on the useful CTIs in a specific domain where they are concerned. In fact, categorized CTIs make it possible to demystify the evolutionary trend of different threats in particular domains. In this section, three insights on three types of attacks for specific domains are identified by manually analyzing specific-domain CTIs, relevant threat descriptions, source codes, etc. The detailed discoveries are discussed as follows.

Discovery 1. The implementations of DDoS attacks vary significantly across multiple domains. When parsing the threat descriptions about the DDoS attack, it is found that attack details of DDoS vary across different domains. More specifically, (i) most of the educational DDoS attacks are TCP flood attacks, in which hackers send a large number of TCP connection requests to the target server, but purposely avoid sending an acknowledgement to the server, which results in a delay at the server. If the attackers send enough connection requests simultaneously, the server resources will be exhausted by such delays, preventing it from responding to requests of legitimate users. (ii) Most of government and ICS DDoS attacks are Domain Name System (DNS) reflector attacks, in which a large number of requests disguising attack target IP are continuously sent to the DNS server. The target service will receive a significant amount of reply packages from DNS, resulting in bandwidth exhaustion. (iii) For finance DDoS attacks, hackers often constantly submit query scripts to the target server for requesting resources. Target servers consume an enormous amount of resources to process these requests, leading to exhaustion of server resources and rejection of the legitimate requests [35]. (iv) In IoT DDoS attacks, however, the attackers invade the IoT devices (e.g., cameras, sensors) exposed on Internet to build botnets, and the compromised devices will be remotely controlled by a covert $C\&C$ server. All compromised devices unconsciously send requests to specific targets simultaneously upon the reception of commands from the $C\&C$ server. In fact, the compromised devices will operate normally except that it consumes more bandwidth, thus traditional safeguard tools are often incapacitated in identifying them.

Discovery 2. As soon as the ransomware appears, its variants will follow, and some will eventually evolve into mining viruses. By analyzing the financial CTIs and educational CTIs, it is found that the Ransomware attacks are increasing sharply in the domains. To explore the relationship between Ransomware attacks, relevant CTIs and source codes of multiple Ransomware are manually analyzed to demystify the origin and their evolution. Particularly, on May 12, 2017, the notorious WannaCry ransom first broke out, which caused unprecedented damage to many key information infrastructures. WannaCry targeted computers running the Microsoft Windows operating system by encrypting data and demanding ransom payments in the form of Bitcoin.

In June 2017, Petya, the variant of Wannacry, was used for a global cyber attack, primarily targeting Ukraine. It also propagates via the EternalBlue exploited by Wannacry. Actually, the impact of Petya on security communities is comparable to that of WannaCry. Scrutinizing the exploitation script of Petya, its attack process mainly consist of three steps: first, accessing disks to scan file system; second, overwriting the computer's master boot record (MBR) to prevent users from entering the system; third, setting restart menu to execute MBR that has been maliciously modified to encrypt the master file table of the NTFS file system. The key encryption function is shown in Listing 1, in which lines 10 to 12 implement file encryption and ransom notice.

```
1  v0=open(”\\c:”,0x4000000u,3u,0,3u,0,0);
2  if(v0)
3  {
4    if(deviceIocontrol(v0,0x70000u,0,0,&OutBuffer,0x18u,&BytesReturn,0))
5        {
6            v1=LocalAlloc(0,10*l_move);
7            if(GenAESkey(lpThreadParameter))
8             {
9                 {
10                    Crypt_file((&filename,a2-1,a3),15,lpThreadParameter);
11                    Write_ransome(1Mz7153HMUxXTur2R);
12                    CryptDestroyKey(*(_DWORD)*) lpThreadParameter+5));
13                }
14            }
15       }
16       CloseHandle(v0);
17 }
```

Listing 1: Petya encrypts files.

Both WannaCry and Petya belong to a family of ransomware based on the EternalBlue vulnerability. However, our analysis unveils notable differences between them. Petya exploits

CVE-2017-0199 vulnerability for phishing attacks, which is then propagated through EternalBlue and Eternal Ransom vulnerabilities. However, WannaCry automatically scans open 445 port of Windows or even electronic information screens, and drop illicit elements such as ransomware, remote control, Trojan horse, miner, and other malicious components in infected computers and servers.

```
 1 v6=openFile(FileName,0xc0000000,3u,0,3u,0,0);
 2 if(v6==(HANDLE)-1)
 3  {
 4     v5=CryptGenkey(*(_DWWORD)*)(a1+8), 0x660Eu,1u,(HCRYPTKEY*)(a1+20);
 5     if(v3)
 6      {
 7        hKey=*(_DWORD *)v1;
 8        CryptSetKeyParam(hkey,4u,pddata,0);
 9      }
10  }
11 if(FileSize.QuadPart <=0x10000)
12  {
13     fileoffsetlow=FileSize.Lowpart;
14     CryptEncrypt_AES(*(_DWORD*)(a2+20));
15  }
16 else
17  {
18     fileoffsetlow=FileSize.Lowpart;
19     CryptEncrypt_RSA(*(_lpBuffer*)(0x200));
20  }
```

Listing 2: NotPetya encrypts disk.

NotPetya, a variant of Petya, spread across the Internet on June 27, 2017. Compared with Petya, NotPetya is more destructive since it can encrypt and lock the whole hard disk and extract passwords from memory or local file system, whose key function is listed in Listing 2. NotPetya chooses to encrypt the file using either *AES* or *RSA* depending on the file size: lines 4 and 9 show the code for generating *AES* and *RSA* keys, line 14 demonstrates the file encryption using *AES* method, and line 19 implements the *RSA* encryption.

In recent years, with the growing popularity of virtual currency, hackers are frantically developing cryptocurrency mining malware. In August 2017, CoinMiner, an advanced ransomware variant using WMI (Windows Management Instrumentation), broke out in the world. It utilizes the WMI standard event script (*scrcons.exe*) to exploit EternalBlue Vulnerability (MS17-010) in order to invade the targeted system and embed the virus permanently in the system.

Discovery 3. The complexity of the phishing strategy is positively correlated with the value of the target. Phishing attack is a type of Internet fraud that seeks to acquire user's credentials through deception, which attempts to obtain sensitive information such as accounts, passwords, and other confidential information. Hackers typically deceive the victims to enter private information on fake websites that look and feel the same as legitimate ones via spamming emails. In this study, our domain-specific CTI analysis allows us to trace the evolution of phishing attacks. It is discovered that they are evolving from *Email phishing* to *Spear phishing*, and eventually to the most complicated *Watering hole phishing*. The phishing strategy adapts according to the value of the target under attack.

(i) Email phishing. *Email phishing* is the simplest and most basic phishing attack, which sends

elaborated emails that the victim trusts to deceive them to respond with the account number, password and other personal information. It often entices the victim to connect to a malicious website that is disguised as a legitimate site such as official online payment website, so that an inattentive victim offers sensitive information. *Email phishing* is frequently used to attack individual users with less values, e.g., to steal game accounts or social media passwords. However, with the popularization of anti-spam software and the improvement of security awareness, this crude phishing has become almost inoperative in recent years.

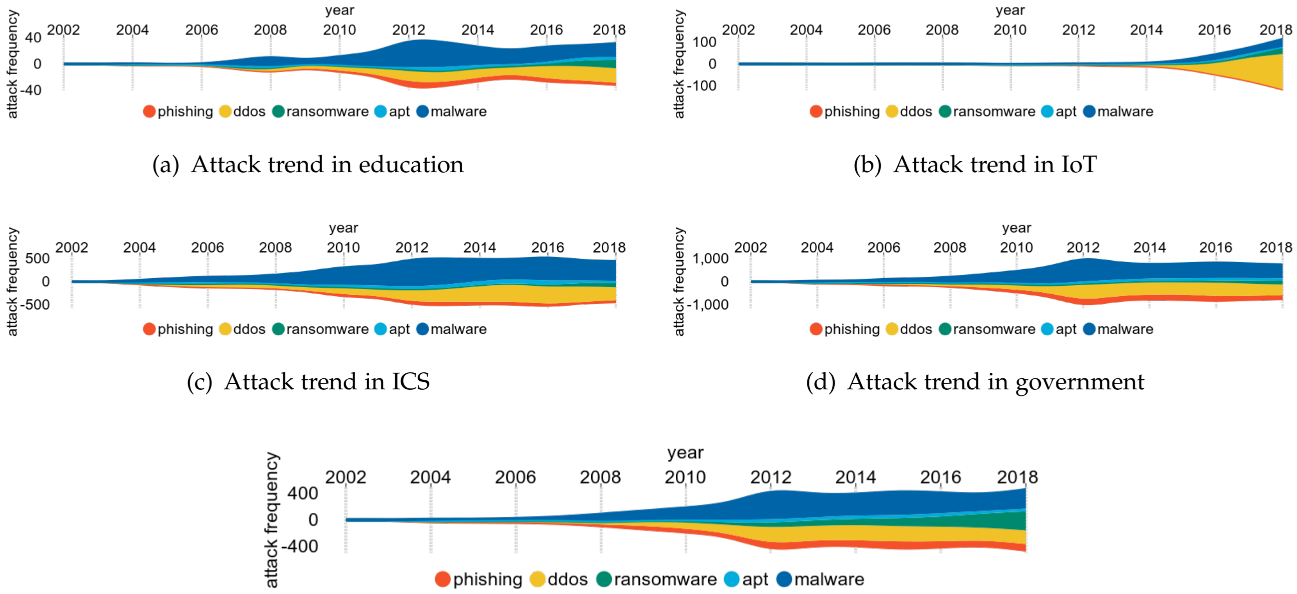
(ii) Spear phishing. Currently, hackers prefer to adopt *Spear phishing* to escape interception of traditional anti-phishing system. *Spear phishing* is a more advanced phishing attack, which sends the victim an email with an attractive headline to entice the victim to open the email carrying Trojan virus. There are two major differences between the *spear phishing* and the *email phishing*: first, *spear phishing* uses more extensive social engineering techniques to gather as much as information about the attack targets, such as the business, cooperation, and trade records of the organization; second, the attacker sends more personalized messages that seem to include the information that the victims are most concerned with. Therefore, the victims are more likely to fall into the trap.

(iii) Watering hole phishing. To escape the most advanced anti-phishing systems, attackers cunningly propose *watering hole phishing*, which is a more advanced form of phishing attack. With *watering hole phishing*, the attackers first identify a set of websites the target group frequently browses, and inject malicious scripts into these websites by exploiting website vulnerabilities. Once victims browse the infected website, malicious elements are automatically downloaded and executed to steal vital secrets or to destroy critical infrastructures. As *watering hole phishing* usually relies on the websites that the attack targets trust, this type of phishing is the most dangerous one compared with *email phishing* and *spear phishing*. Our analysis further exposes that this form of phishing attack is often used by politically connected hacking groups to break into government networks and the highly valuable ICS system.

6.2 Longitudinal Threat Analysis of Different Domains

Based on the domain-specific CTIs, the threat trends of different types of attacks on specific domains are analyzed, and the statistical results are shown in Fig. 8. Particularly, Fig. 8(a) shows that DDoS, phishing, and malware attacks have experienced a significant growth over the years in education domain. Specifically, the attack frequency of malware attack fluctuated over the years, and reached its zenith in 2012. Since then, this type of attack is gradually weakening. Overall, DDoS and malware threats display an upward trend. From 2015 to 2017, it is noticed that the rapid growth of ransomware. In particular, WannaCry broke out on May 12, 2017, and paralyzed the facilities of many educational institutions by encrypting 230,000 computers within a single day. During that time, the attack received widespread attention and was placed on numerous news headlines. As such, the threat description of ransomware attacks reached its peak around that time.

In recent years, the IoT-related threats have developed rapidly due to the growing number of IoT devices. Most IoT devices do not support automated firmware updates or software repairs, and users often do not pay close attention to the security issues including default account and password (e.g., root, administrator, admin, admin123, test), which makes them an enticing attack



(a) Attack trend in education

(b) Attack trend in IoT

(c) Attack trend in ICS

(d) Attack trend in government

(e) Attack trend in finance

Fig. 8: Streamgraph of attacks for different industries. y-axis represents the frequency of different kinds of attacks, where the signs of positive or negative have no real mathematical meaning, i.e., both "400" and "-400" represent 400 attacks that occur in finance as shown in subfigure (e).

target. Meanwhile, many users are indifferent on whether their devices have been maliciously exploited or not, which drive IoT devices to become the most attractive targets for building botnets. A botnet with many compromised devices can effectively evade anti-DDoS system that monitors the IP addresses of incoming requests. As the botnet's DDoS requests are very similar to those of legitimate access, it becomes difficult for traditional DDoS detection systems to recognize such attacks. As illustrated in Fig. 8(b), the DDoS attack has a substantial advantage over other types of attacks in the IoT domain. Since 2015, along with the rapid development of IoT, DDoS attacks related to the IoT devices have seen an explosive growth, while in other domains DDoS attacks are relatively stable over the years. In 2016, Mirai [36] broke out, which uplifted the DDoS attack in IoT to reach an unprecedented threat impact.

In finance industry, *domain-specific CTIs* analysis shows that phishing attacks are dominated, which avoid deliberately destroying files and programs, but stealthily hibernate in the system to collect sensitive information including accounts, passwords, and other personal information. As shown in Fig. 8(e), it is found that since 2007, the frequency of ransomware attacks for the financial industry have increased year by year. Especially since 2013, the frequency of ransomware attacks has shown a linear upward threat. The boom in virtual currencies during that time may have led to this growing trend.

7 Conclusion

Security companies increasingly rely on cyber threat intelligence to enhance resilience against cyber attacks. In this paper, TIMiner, a novel CTI extraction framework, is proposed to automatically extract IOCs and generate categorized CTIs with domain tags from social media. More specifically, first, a domain tagging method based on the variant of CNN is presented to label the domain tags for threat descriptions. Then, a hierarchical IOC extraction approach based on word embedding and syntactic dependency is presented, which is capable of identifying unknown IOCs effectively. Finally, IOCs are combined with their corresponding domain tags to generate the domain-specific CTIs. Domain-specific CTIs can be shared with relevant CTI subscribers, and allow them to quickly identify the security posture in their respective industries. Moreover, Threat-Index is proposed to quantitatively measure the threat severity caused by different types of attack in each domain. By analyzing the domain-specific CTIs generated by TIMiner, new insights about the threats are uncovered and threat trend analysis is performed to facilitate the design of better cyber defense mechanisms for multiple domains.

Acknowledgements

This work was supported by the China NSFC program (No. 61872022, 61421003), the National Key R&D Program China (2018YFB0803503), the 2018 joint Research Foundation of Ministry of Education, China Mobile (MCM20180507), the Opening Project of Shanghai Trusted Industrial Control Platform (TICPSH202003020-ZC), and the Beijing Advanced Innovation Center for Big Data and Brain Computing. Specifically, Qiben Yan is supported in part by the National Science Foundation grants CNS1950171, CNS-1949753.

References

[1] F. Skopik, G. Settanni, R. Fiedler, A problem shared is a problem halved: A survey on the dimensions of collective cyber defense through security information sharing, Computers & Security 60 (2016) 154–176.

[2] S. Singh, P. K. Sharma, S. Y. Moon, D. Moon, J. H. Park, A comprehensive study on apt attacks and countermeasures for future networks and communications: challenges and solutions, Journal of Supercomputing (2016) 1–32.

[3] X. Shu, F. Araujo, D. L. Schales, M. P. Stoecklin, J. Jang, H. Huang, J. R. Rao, Threat intelligence computing, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2018, pp. 1883–1898.

[4] W. Tounsi, H. Rais, A survey on technical threat intelligence in the age of sophisticated cyber attacks, Computers & Security 72 (2018) 212–233.

[5] Q. Chen, R. A. Bridges, Automated behavioral analysis of malware a case study of wannacry ransomware, in: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), 2017, pp. 454–460.

[6] O. Catakoglu, M. Balduzzi, D. Balzarotti, Automatic extraction of indicators of compromise for web applications, in: International Conference on World Wide Web, 2016.

[7] S. Qamar, Z. Anwar, M. A. Rahman, E. Al-Shaer, B.-T. Chu, Data-driven analytics for cyber-threat intelligence and information sharing, Computers & Security 67 (2017) 35–58.

[8] C. Sabottke, O. Suciu, T. Dumitra, Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits, in: Usenix Conference on Security Symposium, 2015.

[9] A. Sapienza, A. Bessi, S. Damodaran, P. Shakarian, K. Lerman, E. Ferrara, Early warnings of cyber threats in online discussions, international conference on data mining (2017) 667–674.

[10] R. Danyliw, J. Meijer, Y. Demchenko, The incident object description exchange format, International Journal of High Performance Computing Applications 5070 (5070) (2007) 1–92.

[11] S. Barnum, Standardizing cyber threat intelligence information with the structured threat information expression (stix), Mitre Corporation 11 (2012) 1–22.

[12] T. D. Wagner, E. Palomar, K. Mahbub, A. E. Abdallah, Towards an anonymity supported platform for shared cyber threat intelligence, conference on risks and security of internet and systems (2017) 175–183.

[13] Fireeye., Openioc, https://www.fireeye.com/blog/threat-research/2013/10/openioc-basics.html, accessed April 20, 2019.
[14] T. Kokkonen, Architecture for the cyber security situational awareness system, in: Internet of Things, Smart Spaces, and Next Generation Networks and Systems, Springer, 2016, pp. 294–302.
[15] J. Sexton, C. Storlie, J. Neil, Attack chain detection, Statistical Analysis Data Mining 8 (5-6) (2015) 353–363.
[16] V. G. Li, M. Dunn, P. Pearce, D. McCoy, G. M. Voelker, S. Savage, K. Levchenko, Reading the tea leaves: A comparative analysis of threat intelligence, in: 28th USENIX Security Symposium, 2019, pp. 851–867.
[17] V. Mavroeidis, S. Bromander, Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence, in: European Intelligence Security Informatics Conference, 2017, pp. 91–98.
[18] D. F. Vazquez, O. P. Acosta, C. Spirito, S. Brown, E. Reid, Conceptual framework for cyber defense information sharing within trust relationships (2012) 1–17.
[19] X. Liao, Y. Kan, X. F. Wang, L. Zhou, R. Beyah, Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence, in: Acm Sigsac Conference on Computer Communications Security, 2016.
[20] O. Catakoglu, M. Balduzzi, D. Balzarotti, Automatic extraction of indicators of compromise for web applications, The web conference (2016) 333–343.
[21] C. Sabottke, O. Suciu, T. Dumitras, Vulnerability disclosure in the age of social media: exploiting twitter for predicting real-world exploits, in: USENIX Security, 2015.
[22] S. Jamalpur, Y. S. Navya, P. Raja, G. Tagore, G. R. K. Rao, Dynamic malware analysis using cuckoo sandbox, in: 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), IEEE, 2018, pp. 1056–1060.
[23] M. Ebrahimi, C. Y. Suen, O. Ormandjieva, Detecting predatory conversations in social media by deep convolutional neural networks, Digital Investigation 18 (2016) 33–49.
[24] I. Deliu, C. Leichter, K. Franke, Extracting cyber threat intelligence from hacker forums: Support vector machines versus convolutional neural networks, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE, 2017, pp. 3648–3656.
[25] J. M. Ahrend, M. Jirotka, K. Jones, On the collaborative practices of cyber threat intelligence analysts to develop and utilize tacit threat and defence knowledge, in: 2016 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (CyberSA), IEEE, 2016, pp. 1–10.
[26] H. P., What is threat intelligence? definition and examples, https://www.recordedfuture.com/threatintelligence-definition, accessed October 15, 2019.
[27] J. Ray, Understanding the threat landscape: Indicators of compromise (iocs) (2015).
[28] J. C. Haass, G.-J. Ahn, F. Grimmelmann, Actra: A case study for threat information sharing, in: Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security, ACM, 2015, pp. 23–26.
[29] C. Z. Liu, H. Zafar, Y. A. Au, Rethinking fs-isac: An it security information sharing network model for the financial services sector, Communications of The Ais 34 (1) (2014) 2.
[30] T. Mikolov, K. Chen, G. S. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv: Computation and Language.
[31] Y. Kim, Convolutional neural networks for sentence classification, empirical methods in natural language processing (2014) 1746–1751.
[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.
[33] K. Yuan, H. Lu, X. Liao, X. Wang, Reading thieves' cant: automatically identifying and understanding dark jargons from cybercrime marketplaces, in: 27th {USENIX} Security Symposium ({USENIX} Security 18), 2018, pp. 1027–1041.
[34] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, arXiv preprint arXiv:1508.01991.
[35] H. Shan, Q. Wang, Q. Yan, Very short intermittent ddos attacks in an unsaturated system, in: International Conference on Security and Privacy in Communication Systems, Springer, 2017, pp. 45–66.
[36] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, Ddos in the iot: Mirai and other botnets, Computer 50 (7) (2017) 80–84.

APPENDIX

Our data collection system continuously collects threat-related text/posts from the social media data sources shown in TABLE 8.

TABLE 8: The list of threat-related social media data sources

Source	URL
AlienVault	www.alienvault.com/blogs/labs-research

BlueCoat	www.bluecoat.com/security/security-blog
Carnal0wnage	http://carnal0wnage.attackresearch.com/
Cert	http://www.cert.org/blogs/
Coresecurity	https://blog.coresecurity.com/
CounterMeasures	https://www.symantec.com/blogs/threat-intelligence
CloudFlare	https://blog.cloudflare.com/
Crowdstrike Blog	https://www.crowdstrike.com/blog/
Crowdstrike Threat	https://www.crowdstrike.com/blog/category/threat-intel-research/
Cryptome	http://cryptome.org/
Cytegic	https://www.cytegic.com/blog/
Darknet	https://www.darknet.org.uk/
Darknet Posts	https://www.darknet.org.uk/popular-posts/
DeepEnd Research	http://www.deependresearch.org/
Ddanchev Blog	http://ddanchev.blogspot.com/
Fireeye Blog	https://www.fireeye.com/blog.html
Fireeye Threat	https://www.fireeye.com/blog/threat-research.html
Forcepoint	https://www.forcepoint.com/blog/x-labs
Fox IT	http://blog.fox-it.com/
Garwarner Blog	http://garwarner.blogspot.com/
Hexacorn	http://www.hexacorn.com/blog
Hotforsecurity	https://https://hotforsecurity.bitdefender.com/
Hotforsecurity Threat	https://hotforsecurity.bitdefender.com/blog/category/e-threats/alerts
Hphosts	http://hphosts.blogspot.com/
Hacker News	https://thehackernews.com/
Hacker Attack	https://thehackernews.com/search/label/Cyber%20Attack
Hacker Malware	https://thehackernews.com/search/label/Malware
Hack Forums	https://hackforums.net
Hacker Vulnerability	https://thehackernews.com/search/label/Vulnerability
Hacker Breach	https://thehackernews.com/search/label/data%20breach
Honeynet	https://www.honeynet.org/blog
Infosecinstitute	https://resources.infosecinstitute.com/
Info Security	https://www.infosecurity-magazine.com/news/
IBM News	https://securityintelligence.com/news/
IBM Threat	https://securityintelligence.com/category/x-force/
Infoblox	http://internetidentity.com/blog/
Juniper	https://forums.juniper.net/t5/Blogs/ct-p/blogs
Kaspersky	https://securelist.com/
Kahusecurity	http://www.kahusecurity.com/2018.html
kahusecurity	http://www.kahusecurity.com/
Krebsonsecurity	http://https://krebsonsecurity.com/
Looking	https://www.lookingglasscyber.com/blog/

Mobile Security	https://blog.trendmicro.com/category/mobile-security/
Microsoft Blog	https://www.microsoft.com/security/blog/
Malwarebytes	https://www.malwarebytes.com/
Malwr	https://malwr.com/
Nakedsecurity	https://nakedsecurity.sophos.com/
Netscout	https://www.netscout.com/blog
Paloa	https://unit42.paloaltonetworks.com/
Paloaltonetworks	https://blog.paloaltonetworks.com/
Radware	https://blog.radware.com/
Radware Ddos	https://blog.radware.com/security/ddos/
Recordedfuture	https://www.recordedfuture.com/blog/
RSA Blog	http://blogs.rsa.com/
Schneier Blog	https://www.schneier.com/
Secniche	http://secniche.blogspot.com/
Schneier News	https://www.schneier.com/news/
Skullsecurity	blog.skullsecurity.org
Spider Labs	https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/
Sucuri Blog	https://blog.sucuri.net/
Sans	https://isc.sans.edu/
SecureAuth	https://www.secureauth.com/blog
Securosis	https://securosis.com/blog
Sight	http://www.isightpartners.com/blog/
Security Intelligence	https://securityintelligence.com/
Security News	https://securityintelligence.com/news/
Trend Micro	https://blog.trendmicro.com/trendlabs-security-intelligence/category/social-media/
Trend Micro Blog	https://blog.trendmicro.com/
Trustwave	https://www.trustwave.com/en-us/resources/
Trustwave Blog	https://www.trustwave.com/en-us/resources/blogs/trustwave-blog/
Taosecurity	http://taosecurity.blogspot.com/
Tripwire	https://www.tripwire.com/state-of-security/
Veracode	https://www.veracode.com/blog
Verisign Blog	https://blog.verisign.com/category/security/
Webroot	https://www.webroot.com/blog/
Welive Security	https://www.welivesecurity.com/
Webroot Intelligence	https://www.webroot.com/us/en/business/threat-intelligence
X-Force	https://securityintelligence.com/x-force/
Zscaler Blog	https://www.zscaler.com/blogs