Cybersecurity Event Detection with New and Re-emerging Words

Hyejin Shin
hyejin1.shin@samsung.com
Samsung Research

WooChul Shim*
woochul.shim@samsung.com
Samsung Research

Jiin Moon
jiin.moon@samsung.com
Samsung Research

Jae Woo Seo
jaewoo13.seo@samsung.com
Samsung Research

Sol Lee
soll.lee@samsung.com
Samsung Research

Yong Ho Hwang
yongh.hwang@samsung.com
Samsung Research

ABSTRACT

There is plenty of threat-related information in open data sources. Early identification of emerging security threats from such information is an important part of security for deployed software and systems. While several cybersecurity event detection methods have been proposed to extract security events from unstructured text in open data sources, most of the existing methods focus on detecting events that have a large volume of mentions. On the contrary, to respond faster than attackers, security analysts and IT operators need to be aware of critical security events as early as possible, no matter how many mentions about an event are made. In this paper, we propose a novel event detection system that can quickly identify critical security events, such as new threats and resurgence of an attack or related event, from Twitter regardless of their volume of mentions. Unlike the existing methods, the proposed method triggers events by monitoring new words and re-emerging words, making it possible to narrow down candidate events among several hundreds of events. It then forms events by clustering tweets linked with the trigger words. This approach enables us to detect new and resurgent threats as early as possible. We empirically demonstrate that our system works promisingly over a wide range of threat types.

CCS CONCEPTS

• **Information systems** → **Information extraction**; • **Human-centered computing** → **Social networking sites**.

KEYWORDS

Event detection; Twitter; EWMA; statistical significance

ACM Reference Format:
Hyejin Shin, WooChul Shim, Jiin Moon, Jae Woo Seo, Sol Lee, and Yong Ho Hwang. 2020. Cybersecurity Event Detection with New and Re-emerging Words. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security (ASIA CCS '20), October 5–9, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3320269.3384721

*Corresponding author

ASIA CCS '20, October 5–9, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-6750-9/20/10…$15.00
https://doi.org/10.1145/3320269.3384721

1 INTRODUCTION

As new technologies like cloud computing, Internet of Things (IoT), artificial intelligence (AI), and 5G are competitively adopted in the digital market, attack surfaces have been continuously widened. This leads to a significant growth of cyber threats and results in exposing individuals as well as organizations to higher security risk. According to Symantec's Internet Security Threat Report [5], not only has the sheer volume of cyber threats increased every year, but the threat landscape has become more diverse, so more and more threats come from new and unexpected sources.

To mitigate the risk from increasing cyber threats, it is important for organizations to sense the ongoing cybersecurity events as early as possible and analyze a potential impact of the detected events on their products, services, and infrastructures. A lot of information related to security threats, vulnerabilities, and attacks is published on various informal sources like social media platforms, blogs, and developer forums on a daily basis, making it almost impossible for a human analyst to manually review and evaluate its relevance to his/her organization. For this reason, technologies for automatic event detection and alert generation from open data sources have attracted great attention in research communities and industries.

Social media is an efficient way to be continuously informed on emerging cybersecurity threats. Among others, Twitter is the most fruitful source to gather threat-related information in terms of volume and diversity. Diverse groups of security stakeholders, ranging from individual security experts, mainstream news to security firms, run Twitter accounts and use Twitter as the source of information diffusion. Tweets posted by these accounts vary from security contest announcement, product promotion, new vulnerability findings to sharing the latest cybersecurity events like ransomware attacks, DDoS attacks, and data breaches. Many recent studies have also demonstrated the possibilities of Twitter as an early sensor for cybersecurity events [41, 49].

In many application areas, the events mentioned on Twitter by several users are important, for example, natural disasters detection or breaking news detection. So, many event detection algorithms have been designed to detect events that are mentioned by a lot of people. However, in security domain, the number of mentions may not be proportional to the importance of an event at the beginning of the event, as we show later in Section 3. A lot of cybersecurity events start to be discussed by only few users in their early stages

and remain unknown for a while until their impacts are analyzed. Such events take a few days to make a number of mentions. For example, the *Magellan* vulnerability, a SQLite remote code execution vulnerability, was mentioned only twice by *@tencent_blade* and *@NickyWu_* on its first day (December 11th, 2018) and less than 5 times by December 14th 2018. On December 15th 2018, the number of mentions exceeded 70. Also, new Android malware *Mysterybot* was first tweeted on June 7th 2018 by *@ThreatFabric*, but did not gain attention for 6 days until its analysis report was published. One day after the analysis report, more than 100 tweets were posted. These kinds of events cannot be detected in their early stages by event detection algorithms requiring a large volume of mentions. Thus, we aim to detect such security events regardless of their volume of mentions. Nonetheless, considering the number of daily security events and a large number of false positives, it is not desirable to detect all the security events from small to large. It is very challenging to decide which events should be perceived. From security perspectives, early identification of new cyber threats or events whose mentions are suddenly increased plays an important role in preventing cyber attacks. Therefore, in this paper, we focus on detection of new and resurgent threats among a large number of daily security events.

Several cybersecurity event detection methods have been proposed to extract events from Twitter [27, 29, 34, 40, 44]. However, as we discuss later in Section 2, they cannot be applicable to achieve the goal above due to at least one of the following limitations: (i) it focuses on detecting events of large volume, which means that there is a high likelihood of failing to detect an event in its early stage [27, 29], (ii) it incurs many false positives to capture events of small volume [27, 29], and (iii) it suffers from many false negatives although it has a capability of early event detection [34, 40, 44].

Motivated by the limitations of the existing methods, we propose a novel event detection system (W2E: Words to Events) that perceives new and resurgent cyber threats in their early stages with low false positive rate and high event detection coverage. W2E achieves this goal by adopting a word-level event monitoring rather than semantic clustering approaches. Among various types of words, W2E identifies new words and re-emerging words to discover new and resurgent cyber threats. We define new words as the words that have not been seen before the time of event detection. They are likely to be linked with new security events like new malware and new vulnerabilities. Re-emerging words are defined as the words that have appeared at least once before the time of event detection, but show a significant rise in their frequencies at the time of detection. They are likely to represent security events associated with the former victims or the former threats like the well-known malware and vulnerabilities. Re-emerging words include company names like "google", product names like "android" or "iphone", malware names like "mirai", vulnerability names like "heartbleed", and technologies like "wpa2". With re-emerging words, we can detect security events that have occurred earlier, but suddenly become an issue again. We can also detect new security events that have no new terms. After identifying new and re-emerging words, W2E merges or splits tweets grouped by the detected words with a clustering algorithm to form events. W2E is a word-level event monitoring, so it may have a performance issue without proper text handling. For example, monitoring words regardless of their parts of speech (noun, verb, adjective, etc.) or inflections results in too many false positives. W2E adopts many natural language processing (NLP) techniques such as Part-of-Speech (POS) tagging, lemmatization, and named entity recognition (NER) to reduce false positives as much as possible. In addition, W2E greatly reduces false positives by restricting data collection with the selected Twitter users.

Our contributions are summarized as follows:

- Despite popularity of Twitter as a data source for event detection, there is no evaluation on whether Twitter is really the right source for early cyber threat detection. We examine coverage and latency of 105 cybersecurity events in 2018 across several open data sources and demonstrate that Twitter is largely the first source, and sometimes the only source, to discuss security events. (Section 3)
- We develop a simple yet effective algorithm for a word-based event detection. The proposed approach enables to quickly identify critical security events, such as new cyber threats and resurgence of an attack or related event, regardless of their volume of mentions. (Section 4)
- We develop a cybersecurity event detection system that achieves high performance. Our evaluation results with real Twitter data show that W2E can achieve high recall (89%), high precision (80%), and low detection latency (45 out of 82 security events were detected with zero latency) over malware, exploit, vulnerability, DDoS attack, and data breach events. (Section 5)

2 RELATED WORK

Event detection has long been a research topic in various application areas. Several works have been proposed to detect special events from Twitter, such as earthquakes, infectious disease outbreak, and terrorist attacks [22, 42, 43, 45],. Many of the existing event detection methods detect events by clustering tweets based on their semantic distances. Other than semantic clustering approaches, there are word-based event detection methods. Unlike many event detection algorithms that are interested in events of large scale [38, 42, 43, 45, 48], Kleinberg [28] assumes that the appearance of an event in data stream is signaled by a burst of activity with certain features rising sharply in frequency as the event emerges. This opens door to the possibility of word monitoring for event detection. There are several methods that analyze word-specific signals in time or frequency domain to detect bursty events [24–26, 32, 47]. Fung et al. [25] proposed a probabilistic approach to detect a set of bursty keywords for a bursty event. Mathioudakis and Koudas [32] introduced an event detection system by identifying bursty keywords and then grouping them with their co-occurrences. Weng and Lee [47] proposed a method by applying wavelet transformation on word-specific signals, such as df-idf scores of words in time domain, to filter out trivial words and then clustering the remaining words to discover events. In a similar manner to Fung et al. [25] and Mathioudakis and Koudas [32], Fedoryszak et al. [24] proposed a real-time event detection system that discovers events whose occurrences are different from normal levels of conversations.

It is natural to consider applying general-purpose event detection methods in the literature to cybersecurity domain. In particular, one

can apply the word-based event detection methods [24–26, 32, 47] to cybersecurity event detection since they can detect events earlier than semantic clustering approaches. However, they are not directly applicable to cybersecurity event detection. In order to avoid a high number of false positives, the way of extracting the monitoring words needs to be redesigned to reflect attributes of cybersecurity events, which is quite challenging. Also, they have their own limitations in detection of critical security events like new and resurgent threats. In cybersecurity domain, new malware appears everyday and is often named with new words. At the beginning of new malware appearance, there are usually few mentions about it. However, they neither provide how to handle new words [25] nor detect cyber threats whose volume is small [24, 32, 47]. In addition, the existing methods [24, 25, 32] sometimes fail to detect resurgent threats like malware variants in their early stages. Since they identified bursty words for event detection by measuring the deviance of the observed occurrences from the expected occurrences computed based on their historical appearance proportions in Twitter stream, they fail to detect events that had been mentioned in large scale earlier, but emerge again with small volume. For example, when a new *Mirai* variant, called *Wicked*, appeared on May 17th 2018, the bursty event detection methods [24, 25, 32] fail to detect it with the word "mirai". This is because "mirai" had appeared over months with moderate to large number of mentions due to its several variants, so its expected occurrence was higher than its observed occurrence (8 tweets) at the time of emergence of the *Wicked* botnet.

Several methods for detecting cybersecurity events from Twitter have been proposed so far. Le Sceller et al. [29] proposed a cybersecurity event detector that identifies events by clustering tweets based on local sensitive hashing (LSH). They clustered tweets within a certain time window to build events and then detected the first stories of streaming tweets by checking whether new tweets form new clusters, as in Petrović et al. [38]. Khadpur et al. [27] proposed an event detection algorithm from Twitter through dynamic query expansion. They constructed events by collecting tweets based on the similarity of dependency graphs between a tweet and a query. They expanded an initial query set iteratively as collecting tweets with high similarity and then specified events from the expanded queries. Although both approaches are interesting, they have some limitations to quickly identify an event. They both require a large volume of tweets for an event to be detected. It means that both methods are likely to fail to detect events early in cases when it takes time for events to be mentioned enough. It also means that if mentions about some critical security events never reach a big number over several days, both methods may not detect such events. Of course, one can apply both methods to capture events with small mentions, but there would be a lot of false positives. According to Petrović et al. [38], a first story detection yields an incredible amount of new stories each day, most of which would be of no interest to anyone. Thus, they focus on detection of significant events that are mentioned by a lot of people. Khadpur et al. [19] reduce the false positive rate than Le Sceller et al. [29] by iteratively specifying events through query expansion. However, queries for events of a few mentions are likely to be expanded less than 2 iterations, which leads a lot of events with a high false positive rate.

There are other event detection methods that do not require a large volume of tweets for an event to be detected. Ritter et al. [40] suggested a seed-based weakly supervised method for extracting cybersecurity events from Twitter. However, their method suffers from many false negatives due to its strong dependency on named entity recognition (NER). The accuracy of NER [39] for entities of interest (e.g., product and company) is not high yet, as its recent application [49] showed that only 377,468 tweets remained among 976,180 tweets containing the keyword "vulnerability" after removing tweets without named entities. This leads that many tweets are likely to be dropped from event candidates. Mittal et al. [34] proposed an ontology-based alert generation system for cybersecurity threats dispersed on Twitter. They constructed a knowledge base system from the existing cybersecurity ontologies like unified cybersecurity ontology (UCO) [46] and DBpedia [23]. Although this approach enables to collect the extra information about a triggered event as well as make a predictive inference about the event by discovering hidden links, its performance highly depends on the information coverage of a knowledge base system. This limits its use in practice. Recently, Sapienza et al. [44] proposed a simple event detection method based on new words for early event detection. However, it can only detect events containing new terms. This results in a failure of sensing important events since not all the emerging events are described by new terms. For example, there may be no new terms to express *Spectre*, *Dirty cow*, and *Heartbleed* vulnerabilities.

3 TWITTER AS A DATA SOURCE

There are a lot of threat-related information feeds. Ideally, monitoring all the data feeds is the best for early event detection. However, there is no generic event detection algorithm that works for any data feeds. One may wonder which data source is a good start. Thus, we evaluate which data source is good to monitor for early security event detection. We have explored the timelines of mentions about a sample of cybersecurity events which occurred in 2018 over various data sources from mainstream news to developer forum.

Setup. We selected 105 security events[1] which consisted of 12 ransomware attacks, 13 botnet attacks, 13 other malware attacks, 12 DDoS attacks, 11 phishing attacks, 6 (vulnerability targeting) exploits, 13 data breach incidents, 12 account hijacking incidents, and 13 disclosed vulnerabilities. We chose the events related to malware, phishing, account hijacking, and exploits from Hackmageddon [14] that met the following conditions: (i) the number of Google search results for an event in the period of past one month from and a week after the event date provided by Hackmageddon is top ranked in their corresponding threat type, and (ii) the description of an event is so detailed as to return less ambiguous search results. Since Hackmageddon does not cover many of data breaches, DDoS attacks, and named vulnerabilities like *Spectre* and *Drupalgeddon2*, we referred to annual security reports [6, 15, 21] to choose popular security events in those categories. We tried to keep the number of events similar for 9 threat types. Nonetheless, exploits were relatively small compared to other threat types due to their nature. We restricted our search domain to Twitter, Facebook, news media, blogs, forums, and security vendor reports. We extracted the keywords for each event from the description given in security news or reports and searched the mentions with the keywords over 6

[1] https://github.com/Samsung/W2E

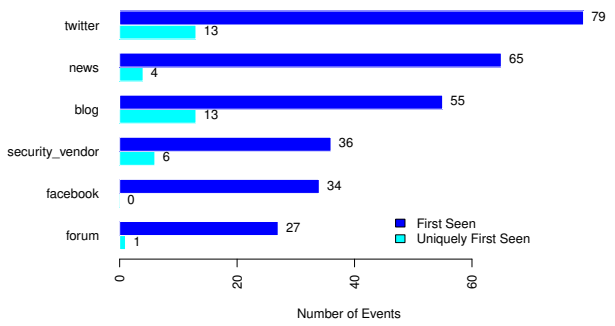

Figure 1: The distribution of data source types on the first day of 105 events. The value on bar is the number of events.

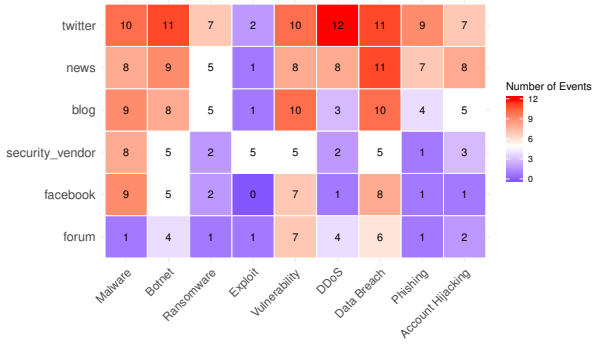

Figure 2: The distribution of data source types for the first mentions about 105 events by threat type.

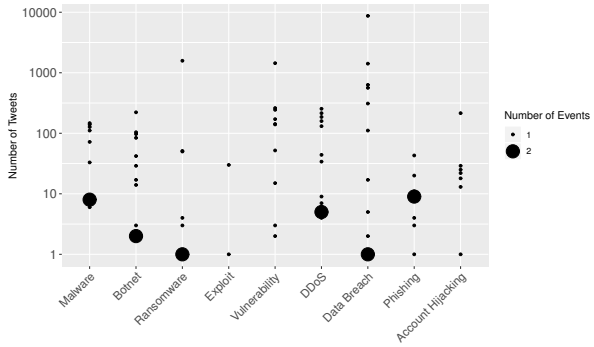

Figure 3: The number of tweets made on the first day of events per threat type.

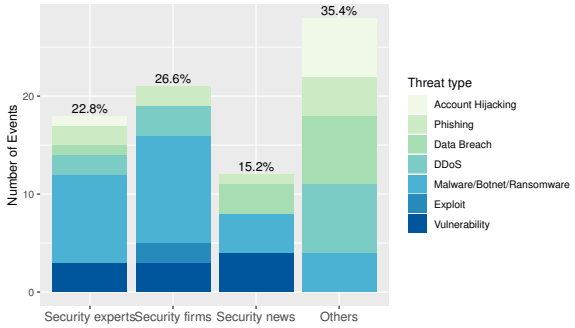

Figure 4: The distribution of Twitter users who first mention about events. The value on bar is the percentage of each user group.

data source types using Google and Recorded Future [18]. For each event, we manually reviewed whether the search results actually corresponded to the event and then got the timeline of the mentions. Note that Recorded Future is a commercialized threat intelligence company which provides the largest data platform in cybersecurity domain.

Findings. The followings are what we have observed from our source investigation.

1. **Twitter is largely the first source, and sometimes the only source, to discuss cybersecurity events.** Figure 1 displays the distribution of data source types on the first day of events. It shows that 75% of the events were discussed in Twitter on the same day as other source types or earlier. We believe that this is because people use Twitter as an information propagation platform. News media, security firms, and individual security researchers often use Twitter to quickly spread out their findings about malware and vulnerabilities after publishing their original articles or reports on their sites. Among the events mentioned on Twitter on their first day, 16% were uniquely seen first in Twitter. Figure 2 displays the distribution of data source types by threat type on the first day of events. It shows that events in most threat types are mentioned on Twitter once they appear.
2. **Twitter is informative enough to collect further information about cybersecurity events.** We found that 82% of the first tweets mentioned events on their first day had links for extra information. Among those external links, 56% were news articles, 34% were blog posts, and 7% were analysis reports like VirusTotal.
3. **Blogs and security vendor reports are also important for early detection of cybersecurity events.** Figure 1 shows that blogs and security vendors together exclusively discussed events more than Twitter – 18% of 105 events are first mentioned only in blogs and security vendor reports. Also, they discussed 68% of the events on the same day as other source types or earlier. Figure 2 shows that blogs and security vendor reports are good sources for sensing events related to malware, vulnerability, and data breach. This tells that one can expect higher coverage and lower latency in cybersecurity event detection by leveraging information from Twitter, blogs, and security vendor reports.
4. **The number of mentions is not proportional to the importance of a cybersecurity event in its early stage.** Figure 3 shows the number of tweets made on the first day of events per threat type. Although we select popular events, many events are mentioned little on their first day. In fact, 23% of the events mentioned on Twitter on their first day were mentioned less than 5 times and 37% of the events were mentioned less than 10 times. This indicates that several security events take time to get some attention from many people regardless of their importance.
5. **Security events related to malware, exploit, and vulnerability are mostly started by security experts, security firms,**

and news media, whereas those related to account hijacking, phishing, and DDoS are introduced more by other users than security-minded users. We have examined the authors of the tweets mentioning events for the first time. We have categorized them into 5 groups – individual security experts (security researchers and ethical hackers), security firms (including their research groups), news media, vulnerability feeds, and others. Figure 4 displays the distribution of the author groups. We observe that 65% of the authors who made the earliest mentions about the events were security experts, security firms, and news media. We also observe that 87% of events related to malware, exploit, and vulnerability were first mentioned by security experts, security firms, and security news while 61% of events related to account hijacking, phishing, and DDoS attack were first mentioned by other users.

6. **There is much less data available in Facebook than Twitter for cybersecurity event detection.** We observe from Figure 1 that 32% of the events were mentioned in Facebook on the first day of events, but they all were covered in Twitter on the same day. Facebook is dedicated to private communication between users, so much less data is available than Twitter although it provides public API for data collection. This makes Facebook less preferred than Twitter for cybersecurity event detection.

4 THE PROPOSED SYSTEM

W2E consists of four steps. The first step is the data collection from Twitter. The second step is the data preprocessing step for extracting the words to be monitored. In the third step, we find the words that trigger events by detecting both new words and re-emerging words. In the last step, we apply similarity-based clustering methods to the tweets associated with the triggered words in order to form events. Figure 5 illustrates the workflow of W2E.

4.1 Data Collection

For event detection, we have been collecting tweets of the selected Twitter users using Twitter streaming API[2] since 2018. While many of the earlier works restrict data collection with keywords [29, 40, 41, 49], we restrict users. We collected Twitter accounts of famous security experts [3, 4], well-known security news media, security firms and their research groups, and vulnerability feeds, together with Twitter accounts following by security experts in our organization. We then chose the users who had posted a certain number of tweets containing a given set of threat-related keywords during consecutive three months. After this process, 560 Twitter users were selected – 50% security experts, 5% news media, 9% security firms, 2% vulnerability feeds, and 34% others (e.g., IT professionals who post security related information).

Note that there are two reasons why we restrict users in our data collection for event detection. First, noise in event detection can be reduced. When security-minded users mention threat-related words like "vulnerability", "breach", and "hack", there are much less false positives. Second, it helps to make an event detection algorithm robust to adversarial attacks. There are many fake news created in social media like Twitter and Facebook. By restricting the data collection with the curated users, we can avoid a detection of events triggered by fake news unless the users we monitor turn to be an adversary.

[2]https://developer.twitter.com/en/docs/tweets/filter-realtime/overview

4.2 Data Preprocessing

Once tweets are collected in between the previous time $t-1$ and the time t of event detection, they pass through the filter with a given set of keywords. The set of keywords is a superset that includes the keyword sets for 5 event types of our interest – malware, exploit, vulnerability, DDoS attack, and data breach – as subsets. Examples of keywords are "malware", "ransomware", "botnet", "trojan", "vulnerability", "vuln", "bug", "exploit", "ddos", and "data breach". Note that the larger set of keywords, the less false negatives, but the more false positives. We then group the tweets into 5 event types according to their corresponding keyword sets. When there are tweets containing the keywords for more than 2 event types, they are grouped in the order of the event types of interest. If one is interested in endpoint device security, he/she may group the tweets in the order of malware, exploit, vulnerability, DDoS attack, and data breach. The remaining tweets containing keywords not for 5 event types are collected into "others" category. We remark that in W2E, any events related to ransomware, spyware, trojans, botnet, rootkits, adware, keyloggers, and any other malicious files are categorized into the malware event type. Note that, as shown later in Section 4.3, the order of categorization of tweets does not affect the performance of our event detection algorithm. It only affects where the detected event is categorized. We also note that we monitor the CVE-related events separately by collecting tweets containing CVE IDs.

The categorization of tweets helps to split multiple events associated with a word by event type. For example, when the word "linux" is detected as a re-emerging word, multiple events like new Linux vulnerability findings and new Linux malware appearance may occur on the same day. In our implementation, we collect very general keywords like "attack", "hack", and "leak" into the keyword set for "others" category so that we do not miss any important security events.

After filtering and categorization, tweets are preprocessed to construct a set of words to be monitored in the following step:

(1) Named entity recognition (NER) is applied to each tweet. Then, we construct the list of person names to remove them later.
(2) A part of speech (POS) tagging is applied to each tweet. Threat words of our interest are malware names, vulnerabilities, companies, and products. These words are mostly nouns, so we tag words in each tweet with parts of speech to extract nouns later.
(3) In each tweet, symbols, emails, URLs, and Twitter handles are removed. The tweets are lowercased. Stopwords are removed. Note that stopwords (e.g., the, a, of, or, to, etc.) are the most common words that appear in the most of the texts. Since many Twitter users overuse their Twitter handles for self-advertisement, Twitter handles make a lot of noises in the word monitoring.
(4) Technology/product terms and their alias are replaced by single representative terms in the form of a single token. For example, "wi-fi" is replaced by "wifi", and "smart tv" and

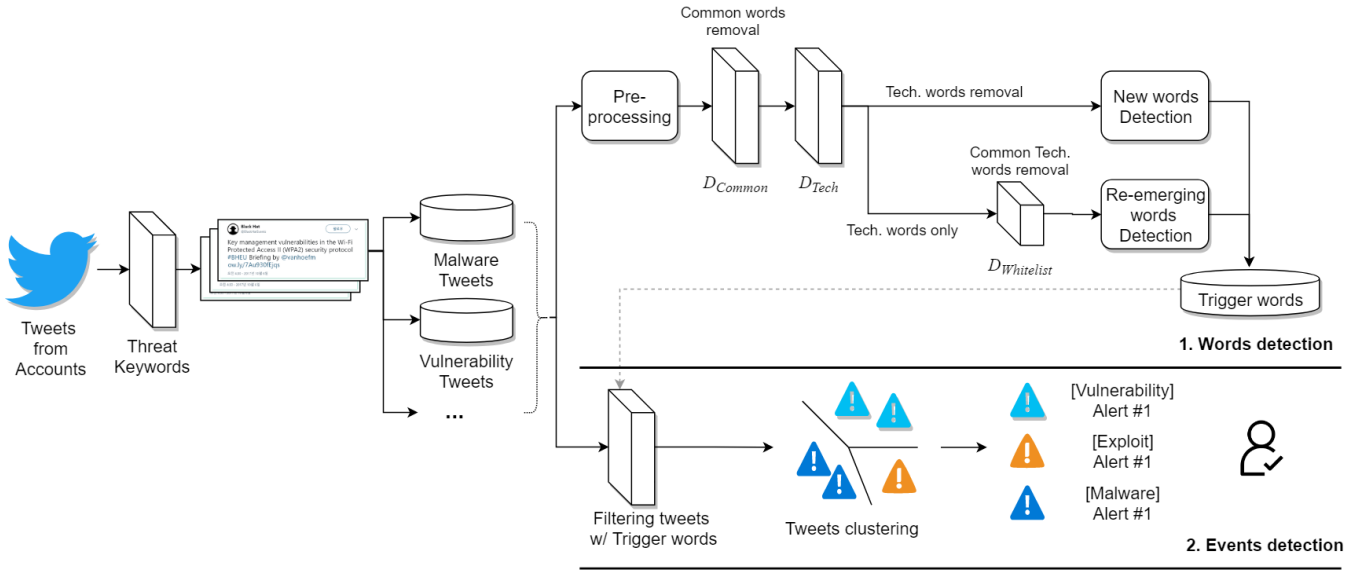

Figure 5: The workflow of our cybersecurity event detection system.

"smart-tv" are replaced by "smarttv". We refer to DBpedia for synonyms.

(5) The tweets are tokenized and only noun words are extracted to form candidate words to be monitored. We keep CVE IDs [9] as well. Then, the words in the list of person names are removed from the candidate words. We also remove the words of single characters. Note that we empirically observe that person names in most tweets are not of our interest although some person names are threat actors.

(6) Lemmatization is applied to each word in order to represent the inflected forms of a word as a single word.

Note that there are many opensource NLP tools for POS tagging and NER such as NLTK [30], CoreNLP [31], twitter_nlp [39], and Twokenizer [35].

4.3 New and Re-emerging Words Detection

Not only is monitoring all the candidate words time-consuming, but also it generates a lot of noise in event detection. Thus, it is important to configure the words that need to be filtered away and the words that can compose security events. For this purpose, we have built the following dictionaries:

- $\mathcal{D}_{\text{Tech}}$: This dictionary is for monitoring re-emerging words. It includes technical words as well as security-specific words like malware names, vulnerability names, company names, and software/hardware names. We built an initial dictionary by performing a statistical significance test for comparing two proportions of a word between security and non-security documents. We used 9,934 security news articles and 8,597 non-security news articles that had been collected from 9 mainstream news sites in 2017. We extracted the words in security news whose occurrences were significantly larger than those in non-security news at 5% significance level. There were 14,592 words in our initial dictionary. We had run our new word detection algorithm daily to 2.82 million tweets collected from famous security [3, 4] for 4 years from 2014 to 2017 and had updated the detected new words to this dictionary as described in below. By the end of 2017, there were 16,014 words in this dictionary.
- $\mathcal{D}_{\text{Common}}$: This dictionary is for deleting common English words. It includes common English words as well as common Twitter terms. To build this dictionary, we extracted the words that appeared significantly more often in non-security news than security news with a statistical significance test. Since the words in Twitter are quite different from those in English dictionary, we also included Twitter words by extracting top-100,000 words from 863 million tweets that had been collected from Twitter using public API in early 2015 without any restrictions on keywords or users. However, some important words in security events like "apple" and "google" were top words in both English dictionary and Twitter. In order to exclude such words from $\mathcal{D}_{\text{Common}}$, we manually reviewed the words that were intersected with $\mathcal{D}_{\text{Tech}}$, Fortune 500 Companies [13], Best Buy [7], Consumer Reports [11], and NVD CPE dictionary [16]. There were 72,623 words in this dictionary by the end of 2017.
- $\mathcal{D}_{\text{Whitelist}}$: This dictionary is for eliminating common technical words that are meaningless to monitor. Examples of such words are "cyber", "cybersecurity", "infosec", and "cyberattack". We extracted common technical words using IDF (inverse document frequency) for the words in $\mathcal{D}_{\text{Tech}}$ over 9,934 security news articles. To extract common technical words from Twitter, we also computed IDF values of the words in $\mathcal{D}_{\text{Tech}}$ over 101,604 tweets containing threat-related keywords that had been collected from January to December 2017. In addition, we included the conference names like "defcon", "bhusa", and "rsac" to this dictionary. There were 2,339 words in this dictionary by the end of 2017.

Note that news articles were processed in the same manner as in Section 4.2 except Twitter-specific processing. When constructing $\mathcal{D}_{\text{Common}}$, we skipped step (2) in Section 4.2 because POS tagging did not work perfectly, so words other than nouns were included in the set of words to be monitored. Also, note that we considered top-100,000 words from Twitter dataset because they covered roughly 98% of the word distribution of 863 million tweets.

We now explain how to detect new and re-emerging words. Let n be the total number of tweets containing a given set of keywords in between time $t-1$ and t. Also, let C be a set of words returned from data preprocessing at time t. Denote by $\mathcal{K}$ a given set of keywords.

New Words Detection. We detect new words by removing the words in $\mathcal{D}_{\text{Tech}} \cup \mathcal{D}_{\text{Common}}$ from the set C. Since the words in $\mathcal{K}$ are not one to monitor, we weed out those words from C as well. After getting candidate new words, we filter out the words whose occurrences in n tweets are not statistically significant [1]. In other words, we retain a word w that satisfies

$$p_t(w) \geq z_\alpha \sqrt{\frac{p_t(w)(1-p_t(w))}{n}},$$

where $p_t(w) = f_t(w)/n$ with the number $f_t(w)$ of tweets containing a word w at time t and z_α is the $(1-\alpha)$-percentile of the standard Normal distribution. Note that $z_{0.05} = 1.645$ for 95% confidence ($\alpha = 0.05$). If one wants to drop more words from candidate new words, he/she can increase the confidence level.

Re-emerging Words Detection. Since the event detection based on new words only works for the events involving new words, its coverage for event detection is very limited. First, it cannot cover threats formerly emerged or their variants. From our experiments, we observe that a new-words-based event detection approach cannot detect the variants of *Spectre* although they have been reported repeatedly since its first seen on January 3rd 2018. Next, it cannot discover the events earlier before new threats are named. We found that there were several tweets about *Key Reinstallation Attack (KRACK)* vulnerability a day before it was publicly announced on October 16th 2017. However, any of those tweets never mentioned new words like "KRACK". For example, the first tweet on October 15th 2017 was *"This is a core protocol-level flaw in WPA2 wi-fi and it looks bad. Possible impact: wi-fi decrypt, connection hijacking, content injection"*. Finally, it does not work for any types of events. Many data breach events do not involve new words. Tweets for data breaches usually mention the victim companies, the size of data breach, and what kinds of user data were exposed, so there are not many new words explaining this event type. To expand the coverage of new-words-based event detection approach, we additionally monitor re-emerging words.

Our algorithm for re-emerging words detection basically monitors the words in $\mathcal{D}_{\text{Tech}}$, but not in $\mathcal{D}_{\text{Whitelist}}$ (i.e., $\mathcal{D}_{\text{Tech}} \setminus \mathcal{D}_{\text{Whitelist}}$). For re-emerging words detection, let C_{R} be the list of the words in $C \cap (\mathcal{D}_{\text{Tech}} \setminus \mathcal{D}_{\text{Whitelist}})$. We first filter out the words in C_{R} whose occurrences are not statistically significant as we do in new words detection. Recall that the re-emerging words are defined as the words that have been seen earlier, but show a sudden increase in their frequencies at time t. Thus, we check if each word in C_{R} makes a statistically significant rise in its occurrence at time t compared to before. There are many ways to measure a change in occurrence of a word. For example, one may compute the difference in the number of tweets including a word at time t and $t-1$. In our algorithm, we measure the difference between the number of tweets containing a word at time t and its expected value based on the past occurrences. To define this mathematically, let $f_t(w)$ be the number of tweets containing a word w at time t. For each word w in C_{R}, we compute the expected number $\hat{f}_t(w)$ of mentions about w by the exponentially weighted moving average (EWMA) over the past k occurrences with the smoothing factor λ ($0 < \lambda < 1$), which is calculated by $\hat{f}_t(w) = \sum_{i=0}^{k-1} \lambda(1-\lambda)^i f_{t-i}(w)$. To determine whether there is a rapid increase in the number of mentions about w, we derive the range of values that $f_t(w)$ can take with high confidence. For this, we compute $\hat{\sigma}(\hat{f}_t(w)) = \hat{\sigma}\sqrt{\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2k})}$ with $\hat{\sigma} = \sqrt{\frac{1}{k}\sum_{i=0}^{k-1}(f_{t-i}(w) - \hat{f}_{t-i}(w))^2}$. If there is no critical issue related on the word w at time t, we expect that $f_t(w)$ takes the values in between $\hat{f}_t(w) - z_{\alpha/2} \cdot \hat{\sigma}(\hat{f}_t(w))$ and $\hat{f}_t(w) + z_{\alpha/2} \cdot \hat{\sigma}(\hat{f}_t(w))$ with 100(1-α)% confidence since $\frac{f_t(w) - \hat{f}_t(w)}{\hat{\sigma}(\hat{f}_t(w))}$ is approximately Gaussian distributed with its mean 0 and variance 1. Otherwise, $f_t(w)$ is more likely to take the value larger than the above upper bound. Therefore, we detect a word w as a re-emerging word if $f_t(w)$ satisfies

$$f_t(w) \geq \hat{f}_t(w) + z_{\alpha/2} \cdot \hat{\sigma}\sqrt{\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2k})}.$$

Note that a higher λ decreases the effects of older observations faster. Also, note that $z_{0.025} = 1.96$ for 95% confidence.

Figure 6 and 7 show examples of re-emerging words. The words "spectre" and "intel" are both detected on January 3rd 2018 when Intel CPU vulnerabilities were publicly disclosed. Both words showed a rapid rise in the number of mentions. In particular, the word "spectre" had never appeared in a month before event. In the case of the word "wifi", it had been constantly mentioned with various events from "wifi password hack", "wifi cracker" to "wifi firmware bug". Unlike "spectre" and "intel", the number of mentions was mostly below 10 times even when they are triggered as events.

Note that, although new and re-emerging words are extracted in each event type, the filtering rules of words are applied to the number of tweets mentioning each word across all the event types. Therefore, the order of categorization of tweets into event types does not affect the words detected. It only affects where the event is categorized.

We also note that, although we focus on event retrieval from Twitter, the proposed algorithm is applicable to security news monitoring as well as forum monitoring. In fact, we have successfully applied the proposed algorithm to event detection from security news although we do not report here.

Dictionary Update. Since new events keep appearing every day, we need to update $\mathcal{D}_{\text{Tech}}$ for re-emerging words detection as we detect new words. One may update $\mathcal{D}_{\text{Tech}}$ on a daily basis as new words are detected. However, automatic update of the detected new words accumulates noise in the dictionary $\mathcal{D}_{\text{Tech}}$. This leads to increase false positives in re-emerging words detection. Thus, to reduce cumulative noise caused by automatic dictionary update, either a daily human review or a conservative dictionary update policy is required. To make our system fully automatic, we choose the latter option. We decide to update each of the detected new words on $\mathcal{D}_{\text{Tech}}$ when it keeps being detected as new words at least

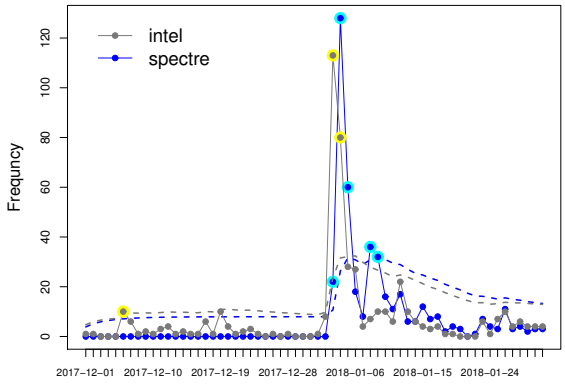

Figure 6: The number of mentions about "intel" and "spectre" from Dec 2017 to Jan 2018. The grey and blue dotted lines represent the upper bounds for frequencies of "intel" and "spectre", respectively. The yellow and cyan spotted circles represent the days that the words are flagged as re-emerging words. Both "intel" and "spectre" identify the "spectre" vulnerability on its first day (Jan 3rd 2018).

twice in a week. Until the detected new words are updated into $\mathcal{D}_{\text{Tech}}$, those words show up in the list of new words.

Monitoring CVE IDs. When new vulnerabilities are found or the known vulnerabilities are mentioned again, sensing those vulnerabilities is important for organizations to mitigate a potential risk. W2E monitors some vulnerabilities from the tweets including a given set of vulnerability-specific keywords. One may monitor CVE IDs in the vulnerability event category by including "cve" into the set of keywords. However, in that case, CVE IDs generate too many events as well as dominate top events in the vulnerability event category, so a human analyst may ignore the vulnerability events without CVE IDs unless they are sufficiently mentioned. Thus, W2E monitors CVE IDs separately. In CVE monitoring, we are interested in pre-NVD CVEs, which are defined as CVEs whose IDs are assigned, but are not published in NVD (National Vulnerability Database) yet. After running our CVE monitor on Twitter, we have found 345 pre-NVD CVEs during the periods from January to December 2018. Among them, 309 CVEs are published and 36 CVEs are not yet published on NVD, as of April 30th 2019. Those CVEs have been mentioned at least 1 day earlier than NVD and at most 450 days earlier than NVD (mentioned 46 days earlier than NVD on average). We are also interested in detecting the CVEs that already have published in NVD, but have mentioned again on Twitter in some reasons. If discourses on a CVE in NVD are rapidly increased, then organizations need to evaluate its risk on their products/services/infrastructures and check whether the CVE is patched. Unlike threat words detection algorithms introduced above, we use CVE lists fed from NVD right before the time t of event detection as the dictionary $\mathcal{D}_{\text{CVE}}$, instead of $\mathcal{D}_{\text{Tech}}$, $\mathcal{D}_{\text{Whitelist}}$, and $\mathcal{D}_{\text{Common}}$. Let $\mathcal{C}_{\text{CVE}}$ be a set of CVE IDs obtained from tweets collected in between time $t-1$ and t. In CVE monitoring, if CVE IDs in $\mathcal{C}_{\text{CVE}}$ are not in $\mathcal{D}_{\text{CVE}}$, then those CVEs are identified as new words. To avoid typos, we eliminate the CVE IDs not found in MITRE. For CVE IDs in $\mathcal{C}_{\text{CVE}} \cap \mathcal{D}_{\text{CVE}}$, we check if each CVE ID is mentioned enough and it shows a rapid rise in its occurrence. For re-emerging CVE detection, we exclude tweets from vulnerability feeds and we use the same filtering rules above, where n is the number of tweets containing CVE IDs.

4.4 Event Generation

Our system detects events by identifying new words and re-emerging words. However, this approach has a limitation that a word does not have one-to-one correspondence with an event. That is, (i) two or more detected words may represent one event – a new word and a re-emerging word or two new words can come from one tweet, and (ii) the detected word may not correspond to one event. The latter case happens more often in event detection through re-emerging words. For example, when the word "wifi" is detected as a re-emerging word, it may be buzzed with wifi firmware bugs and wifi inspector vulnerabilities on the same day.

To overcome the problem above, we develop an event generator that merges or splits candidate events triggered by the detected words as the final step of W2E. In each event type, our event generator performs clustering analysis on the tweets containing new words and re-emerging words. Many security events are described by context-specific words like the malware names, vulnerabilities, victims, and attack targets. Thus, mentions about the same event are likely to contain the same event-specific words. For this reason, we extract a set of such words from each tweet and measure the similarity between two tweets by computing the Jaccard similarity. We extract event-specific words from each tweet in the following steps: (1) We apply steps (3) and (4) in Section 4.2, (2) Security terms and their alias are replaced by single representative terms in the form of a single token. For example, "buffer overflow", "buffer-overflow", "buffer_overflow", and "buffer overrun" are replaced by "buffer-overflow", (3) After tokenization and lemmatization, we prune the words in $\mathcal{D}_{\text{Common}} \cup \mathcal{D}_{\text{Whitelist}} \cup \mathcal{K}$. We then group tweets by applying a hierarchical clustering method to the Jaccard distance matrix. After clustering tweets within each event type, we finally form events by grouping clusters of tweets across all the event types in a similar manner. Note that two tweets having the same external link form the same event.

Note that there are several clustering methods to group tweets for event detection purposes [22]. One can adopt word embeddings such as word2vec [33], GloVe [36], and ELMo [37] to represent tweets into a vector space so that semantic distance between two tweets is measured. However, we observe that, for tweets about security events, clustering with context-specific words performs much better than semantic clustering.

Since W2E runs in day to day, the same event can come up again and again while it is being discussed on Twitter. Generating the same alert repeatedly whenever an event is detected is inefficient and annoying. Thus, we develop an event manager that merges the events detected at time t into the events detected up to time $t-1$. Our event manager first takes over the events detected within the past 7 days. It then retains event-specific words that appear at least 50% of tweets in each event in order to extract context-specific words for an event. It finally merges the detected events at time t into the past events if the sets of context-specific words for two events have the Jaccard similarity greater than 0.7.

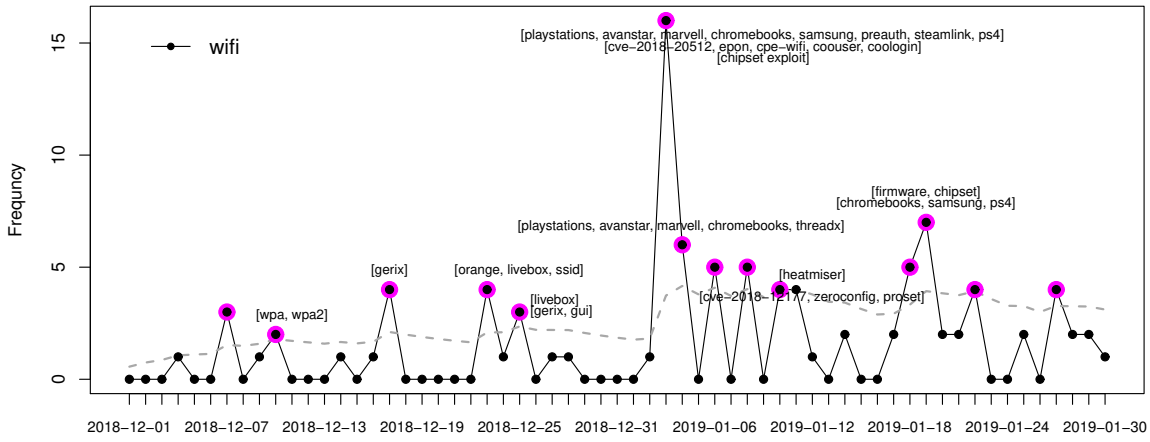

Figure 7: The number of mentions about "wifi" from Dec 2018 to Jan 2019. The grey dotted line represents the upper bound for frequency of "wifi". The magenta spotted circles represent the days that "wifi" is flagged as re-emerging word. On Jan 3rd 2019, our algorithm detects 3 events about wifi where the biggest event was "Marvell Avanstar Wifi SoC bug" and mentioned 9 times. The set of words in brackets is the words detected by our algorithm and indicates one event.

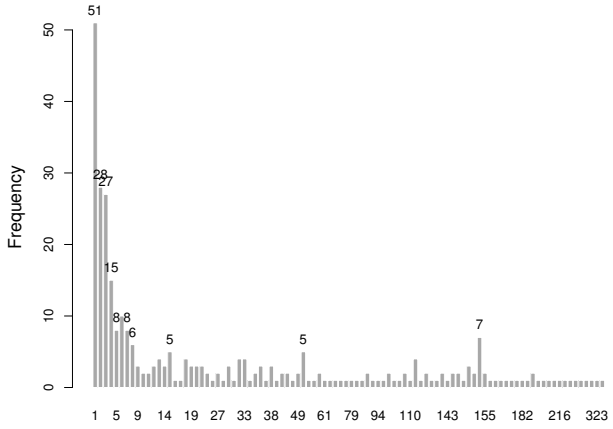

Figure 8: The number of days for pre-NVD CVEs found in Twitter to be published on NVD. The value on bar is the number of pre-NVD CVEs.

5 EVALUATION

5.1 Evaluation Setup

We have run W2E on a daily basis over tweets collected from 560 Twitter users during the period from January 2018 to April 2019. The total volume of our Twitter dataset is 1,647,629 including retweets.

We evaluate the performance of W2E in 3 aspects - (i) Clustering accuracy for daily event generation, (ii) Daily event detection accuracy, and (iii) Coverage and detection latency. For (i) and (ii), we selected the results of W2E in April 2019. Note that we observed similar results in another selected month (June 2018) to April 2019 although we did not report it here. There were roughly 5,900 unique tweets triggered by new/re-emerging words. Five security experts manually annotated the cluster label of each tweet and decided whether a detected event is a genuine security event or not. The annotator made the judgement by referring to either the external links in a tweet or Google search. For (iii), we took 82 events that fell in the malware, vulnerability, exploit, DDoS attack, and data breach event types from 105 security events in Section 3. We analyzed whether W2E detected them and, if so, when they were detected. For latency computation, we referred to the date of the first tweet about an event over the whole Twitter that we investigated in Section 3. We remind that W2E categorizes any events related to ransomware, spyware, trojans, botnet, rootkits, adware, keyloggers, and any other malware into the malware event type.

In our implementation, we use 200 keywords[1] from single words to terms – 28 keywords are for malware-related events, 11 for exploit-related events, 20 for vulnerability-related events, 6 for DDoS attacks, and 17 for data breaches. The initial keywords were chosen by reviewing the terms in CWE [10], CAPEC [8], STIX [19], and ENISA Threat Taxonomy [12]. We then included the plural form, inflections, and alias of each keyword into our keyword set. We set $\alpha = 0.05$ for both new words and re-emerging words detection. We use the dictionaries[1] that were constructed as explained in Section 4. Note that there were 72,623 words in $\mathcal{D}_{\text{Common}}$, 16,014 words in $\mathcal{D}_{\text{Tech}}$ and 3,078 words in $\mathcal{D}_{\text{Whitelist}}$ by the end of 2017. We use Stanford CoreNLP for POS tagging and NER.

5.2 Evaluation Results

Clustering accuracy. To measure the clustering accuracy of our event generator, we compared the estimated cluster by our event generator to the human-labeled cluster and then computed normalized mutual information (NMI) [2]. Note that NMI is one of the popular metrics to evaluate clustering quality. It is always a number between 0 and 1, and 1 means the perfect clustering. Figure 9 presents daily NMI of our event generator in April 2019. NMI was larger than 0.9 in most of days in the selected month. The average NMI over the month is 0.96 with standard deviation (SD) 0.06, which indicates that our event generator performs well enough

to split different tweets sharing the same event-specific word or merge similar tweets into one cluster with small errors.

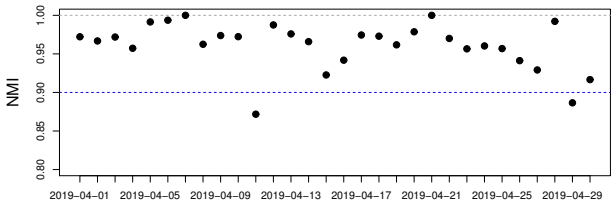

Figure 9: Daily NMI of our event generator in April 2019.

Daily event detection accuracy. We measure how many false positives are generated daily. Table 1 shows the precision of W2E over all the daily events in the selected month. The overall precision of W2E is 80% and the precision for each threat type is nearly or larger than 70%. W2E detected 2,359 daily events in total (79 daily on average), which form 930 unique events. Among 930 unique events, 763 events were genuine security events, so the precision of W2E for the unique events in the month is 82%. Table 4 shows examples of some important events in April 2019 detected by W2E.

Coverage and detection latency. We analyze the coverage of W2E for the events related to malware, vulnerability, exploit, DDoS attack, and data breach in Section 3. Table 2 shows the recall of W2E over 82 security events. The overall recall of W2E is 89% – 73 out of 82 events were detected. Among 73 events, 26 events were detected only by re-emerging words because they do not include any new terms in their tweets. This verifies the importance of re-emerging words monitoring. Although W2E shows high recall for malware attacks, exploit incidents, vulnerabilities, and data breach incidents, the recall for DDoS attacks is relatively low. We think that a lower coverage of DDoS attacks is because discussions about such events are likely to be started by any users who suffer from those attacks rather than security-minded users, as we observed in Section 3. We also observe that W2E leads to 0.67 days delay in detection on average after the first seen in Twitter. Nonetheless, 45 out of 82 events were detected on the day that the event first appears on Twitter. Also, 17 events were detected the next day. Note that, although 29 out of 82 events were mentioned less than 10 times on their first day in the whole Twitter, W2E could detect 12 events among them. In particular, W2E can detect botnet attacks, exploit incidents, and vulnerabilities without almost no latency, which is quite meaningful from early event detection perspectives. In addition, we observe that 31 out of 82 events were detected by W2E on the first day of events while there were 2.68 days delay in detection on average since the first day of the event. Note that there were 1.89 days delay between the first tweet and the first day of events. Table 3 lists some exemplary events that were detected by W2E on the day that the event first appeared.

5.3 Comparison with the existing methods

As discussed in Section 2, there are several cybersecurity event detection methods. Among them, we chose Ritter et al. [40] for performance comparison. Since Khadpur et al. [27] and Le Sceller et al. [29] require large volume of mentions about an event to be detected, they are not the competitors to W2E. On the other hand, Ritter et al. [40] and Sapienza et al. [44] are designed to detect events regardless of the number of mentions. Since W2E monitors both new words and re-emerging words, it widens the event detection coverage of Sapienza et al. [44]. Further, it greatly reduces false positives compared to Sapienza et al .[44] by applying many advanced NLP techniques to extract the monitoring words. Thus, we did not compare W2E to Sapienza et al. [44].

For comparison with Ritter et al. [40], we collected their event detection results in April 2019 from the web [20]. They detect events and categorize into 4 event types: data breach, DDoS, exploit, and vulnerability. There were 451 tweets in total. The same annotators judged whether each event is a genuine security event or not.

While W2E monitors the selected Twitter users for event detection, Ritter et al. [40] monitor the whole Twitter with specific keywords. Thus, their method is supposed to have lower false negative rate and earlier event detection than W2E. However, the comparison results show that W2E outperforms it – higher event detection coverage, lower detection latency, and lower false positive rate. Specifically, the precision of Ritter et al.'s method over all the daily events in data breach, DDoS, exploit, and vulnerability categories is 62% while that of W2E is 82%. For data breach, DDoS, exploit, and vulnerability categories, the number of unique security events detected by Ritter et al.'s method was 129 while that for W2E was 537, where 87 events were detected by both methods. W2E can cover 67% of security events detected by Ritter et al.'s method while Ritter et al.'s method covers only 16% of security events detected by W2E. In Table 4, we present some important events in April 2019 that were detected only by W2E, but not by Ritter et al.'s method. For example, W2E detected an event about prototype pollution flaw in jQuery JavaScript library, which is used on 74 percent of all internet sites [17], from April 16th 2019 and came up for few consecutive days. On the other hand, Ritter et al.'s method failed to detect it. Considering the popularity of jQuery, missing such vulnerability event may lead an undesirable situation.

For 87 events that both methods detected, we analyzed their first detection date. Table 5 shows the comparison results in detection time. We found that 29 events were detected earlier by W2E (2.7 days earlier on average) while 8 events were detected earlier by Ritter et al.'s method (2 days earlier on average). The remaining 50 events were detected on the same day by both methods. Although W2E collects data from a limited Twitter users, it can detect important security events much more and earlier than Ritter et al.'s method. We believe that higher false negative rate of Ritter et al.'s method is due to a poor performance of NER.

5.4 Case Studies

Among several events detected by W2E from January 2018 to April 2019, we have chosen 4 events – Lokibot malware, Drupal vulnerability, Firebase data breach, and WiFi firmware bug.

Lokibot (Malware) – This malware is a Trojan horse that steals information from the compromised computer. Trustwave researchers found new spam campaign pushing Lokibot and it was broadcasted by news media like Threatpost with its analysis and mitigation guidance on April 5th 2019. On the same day, W2E also detected the event with the words "zipx", "png", and "lokibot". However, the

Table 1: Precision of W2E over all the detected daily events.

Threat Type	Precision	# Total Daily Events	Daily Precision		# Daily Events	
			Mean	SD	Mean	SD
Malware	75.1%	819	72.8%	13.7%	27.3	13.2
Exploit	89.9%	519	89.5%	10.5%	17.3	9.8
Vulnerability	77.2%	697	77.2%	15.2%	23.2	13.5
DDoS	91.2%	193	88.8%	22.2%	6.4	4.3
Data breach	69.5%	131	69.5%	25.3%	4.4	3.1
Total	80.0%	2359				

Table 2: Recall of W2E over 82 security events in 2018.

Threat Type	Recall	# Events	# Detected Events[a]	Latency (days)	# Events of Zero Latency[b]
Malware	89%	38	34 (8)	0.76	20 (5)
Exploit	100%	6	6 (5)	0.17	5 (3)
Vulnerability	100%	13	13 (6)	0.46	10 (3)
DDoS	67%	12	8 (3)	1.12	2 (0)
Data breach	92%	13	12 (4)	0.58	8 (1)
Total	89%	82	73 (26)	0.67	45 (12)

[a] Values in the parenthesis are the number of events detected only by re-emerging words,
[b] Values in the parenthesis are the number of events mentioned less than 10 times on their first day, but detected by W2E.

Table 3: Examples of events detected by W2E on the first day of events.

Detected Word	Detected Date	Threat Type	Event Description
moneropay	2018-01-13	Malware	MoneroPay ransomware
iot	2018-01-24	Malware	Hide 'N Seek botnet
ioncube	2018-02-27	Malware	IonCube malware
prowli	2018-06-06	Malware	Prowli botnet
virobot	2018-09-21	Malware	Virobot botnet
doubledoor	2018-02-14	Exploit	DoubleDoor botnet
apache	2018-08-22	Exploit	Struts Vuln (CVE-2018-11776) attack
masterky	2018-03-13	Vulnerability	AMD masterkey
intel	2018-05-03	Vulnerability	Specre-NG
foreshadow	2018-08-14	Vulnerability	Window10 foreshadow
bleedingbit	2018-11-01	Vulnerability	Bleedingbit
protonmail	2018-06-27	DDoS	Protonmail DDoS attack
uaa	2018-03-29	Data breach	MyFitnessPal data breach
marriott	2018-11-30	Data breach	Marriott data breach
quora	2018-12-03	Data breach	Quora data breach

interesting point is that W2E detected an event related to Lokibot on April 2nd 2019, where some of tweets contain the download URLs of Lokibot. One of the URLs on the tweet, *bluewales.ml/wp/wp-content/uploads/2019/04/Panel/five/fre.php*, was detected as malicious by some engines in VirusTotal on April 4th 2019, which is 2 days later than W2E. This case shows the capability of W2E for collecting the recent indicators of compromise (IOCs).

Drupal (CVE-2018-7602) – This is a RCE vulnerability in Drupal, an open source contents management framework. W2E detected this critical vulnerability on April 25th 2018 as a pre-NVD CVE with the words "drupal" and "cve-2018-7602". We could get the description of this vulnerability and its mitigation from the external link in the tweets. The Drupal Security Team strongly recommended immediate update because the vulnerability is highly critical (its CVSS score is 9.8). Later, they confirmed that the vulnerability is being exploited in the wild. This case shows that early sensing of the vulnerability and immediate update are extremely important.

Firebase (Data Breach) – Firebase is a Backend-as-a-Service from Google that contains a vast collection of services. Mobile developers use it for making mobile and web-based apps. According to the report of Appthority researchers, thousands of iOS and Android apps leaked sensitive data of users via misconfigured firebase backend. Before publishing the report, the Appthority researchers notified Google about the issue and provided a list of affected apps and Firebase database servers. However, all the developers might not recognize the risk of Firebase and it took a while for Google to fix the problem. For mitigating the risk, the developers need to recognize the issue as soon as possible. W2E detected this issue earlier than news media and delivered the information about this incident as the alert. W2E first detected the word "firebase" on June 20th 2018, together with the informative words "android" and "database". Although only 2 tweets mentioned "firebase" on the day of detection, W2E could detect it by the proposed re-emerging words detection algorithm.

Marvell Avanstar (WiFi Firmware Bug) – The Marvell Avanstar Wifi chip SoC bug was publicly disclosed on January 18th 2019

Table 4: Examples of events in April 2019. The events detected by W2E only are marked with ○. Likewise, ● for Ritter et al. [40], and ◐ for both. "New" in word type represents "new word" and "Re" represents "re-emerging word".

Threat Type	Word Type	Detected Word (Description)	Date (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30)
Malware	Re	Exodus	○ ○ ○ ○ ○ ○ ○ ○ ○ ○
Malware	Re	Emotet	○ ○
Malware	New	DDoSMan	○ ○ ○ ○ ○ ○ ○ ○ ○ ○
Exploit	Re	WordPress (Yuzo Plugin)	○ ○ ○ ○ ○ ○ ○ ◐ ◐ ◐ ◐ ◐ ◐ ○ ○
Malware	New	Xloader	○ ○ ○ ○
Malware	Re	Bashlite	○ ○ ○ ○ ○ ○
Breach	Re	Facebook (App)	○ ○ ◐ ○ ○ ○
Malware	New	XWO (MongoLock)	○ ○ ○ ○
Malware	Re	WINNTI	○ ○ ○
Malware	New	BasBanke	○ ○
Vulnerability	Re	Xiaomi (GuardProvider)	○ ◐ ● ● ●
Malware	Re	Necurs (US Malware Host)	○ ○
Malware	Re	Lokibot	○ ○
Breach	New	AeroGrow	○ ○ ○
Malware	Re	Anubis	○ ○
DDoS	Re	Electrum	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ◐ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
Malware	New	Baldr	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
Malware	Re	Mirai (Variant)	○ ○ ○
Malware	New	Hoplight	○ ○ ○
Exploit	Re	WinRaR (CVE-2018-20250)	○ ◐ ○ ○ ○ ○
Vulnerability	New	Dragonblood	○ ○ ○ ○ ○ ○ ○
DDoS	Re	HTML5 (Hyperlink)	○ ○ ○ ○ ○
Breach	Re	Outlook	○ ◐ ◐ ◐ ◐ ◐ ◐ ◐ ◐ ◐ ◐ ◐ ◐ ◐ ◐ ◐ ◐ ◐ ◐ ◐
Malware	Re	Bayrob	○ ○ ○ ○
Vulnerability	Re	jQuery (Prototype Pollution)	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
Vulnerability	Re	Broadcom (WiFi)	○ ○ ○
Breach	Re	LinkedIn	○ ○ ○ ○ ○ ○
Malware	New	Karkoff	○ ○ ○
Malware	New	Videogames (Snuck)	○ ○ ○
Vulnerability	Re	Sony (Smart TV)	○ ○ ○ ○
Vulnerability	Re	Qualcomm (Chip)	◐ ○ ○ ○ ○ ○
Exploit	New	AESDDoS	○ ○ ○ ○
Vulnerability	New	iLinkP2P	○ ○ ○ ○ ○
Breach	Re	DockerHub	◐ ◐ ◐ ◐
Breach	Re	Database (80M US Households)	○

Table 5: Comparison of W2E with Ritter et al. [40].

	W2E only	[40] only	Common				
			W2E = [40]	W2E < [40][a]		[40] < W2E	
Threat Type	# Events	# Events	# Events	# Events	Mean Latency (days)	# Events	Mean Latency (days)
Data breach	25	17	9	6	-1	6	-2.3
DDoS	127	2	1	1	-9	0	0
Vulnerability	203	23	40	22	-2.9	2	-1
Exploit[b]	269	0	0	-	-	-	-
Total	450	42	50	29	-2.7	8	-2

[a] "A<B" means that A detects earlier than B and a negative number means how many days one method detects an event earlier than the other,
[b] It includes not only exploits in the wild but also PoC and exploit kits.

through Embedi blog and ZDNet. On January 3rd 2019, W2E detected the event with the words "wifi", "chromebooks", "marvell", "avanstar", "playstations", and "samsung" from the tweet *"unauth, unassoc remote code exec on the Marvell Avanstar Wifi chip SoC used in Playstations, Xbox, Surfaces, Chromebooks, Samsung phones and more in under five minutes attack time. Bonus second stage escalation in the linux drivers, PoC on steamlink. https://t.co/s54QBc5mDK"*. This case clearly shows the early detection capability of W2E as well as the benefit of monitoring open data sources.

6 LIMITATION

Twitter User Restriction. In W2E, there are various design choices that can affect its performance. We design W2E to achieve high performance on detection of security events that threaten end-user devices like smartphones, smart appliances, and IoT devices. For

this, we construct the list of users who mention security events related to end-user devices in accordance with our observation in Section 3 (Security events related to malware, exploit, and vulnerability are mostly started by security experts, security firms,and news media). This leads that our data collection is more from security-minded users than other users. Although user restriction is effective to reduce false positives, it can decrease event detection coverage and increase detection latency. As seen from the evaluation results in Section 5.2, W2E leads to 0.67 days delay in detection on average, although 45 out of 82 events were detected without delay. This time delay is likely caused by user restriction. In addition, we analyzed 9 events that could not be detected by W2E. We found that they were never mentioned or mentioned only once by the users in our Twitter user list. Also, as seen in Table 2, W2E showed a relatively lower recall for DDoS incidents than malware attacks, exploit incidents, vulnerabilities, and data breach incidents. We believe this is caused by our user construction strategy. As we observed in Section 3, discussions about events like DDoS, phishing, and account hijacking events are likely to be started by any users who suffer from those attacks rather than security-minded users.

Keyword-based Filter. W2E extracts tweets for candidate events through the filter with a set of threat-related keywords. Majority of the existing event detection methods in the cybersecurity domain take keyword matching as the first step to retrieve tweets related to security events [27, 29, 40, 49]. However, it is difficult to configure how keywords should be selected to keep the balance between false positive rate and false negative rate. Even with the security-specific keywords like "malware", we observed several tweets irrelevant to security events, which causes false positives. For example, there are several tweets hiring malware analysts in malware category. Due to this problem, Ritter et al. [40] and Zong et al. [49] built a classifier to determine whether each tweet containing keywords is a real security event or not. Since W2E monitors the words in tweets containing keywords, it is sensitive to noisy tweets. Although we restrict Twitter accounts, most of false positives in W2E come from noisy tweets that pass through the keyword-based filter. For this reason, replacing the keywords-based filter with a classification-based filter can help to reduce false positives.

7 CONCLUSION

We propose a novel word-based cybersecurity event detection system. The proposed system monitors new words and re-emerging words by analyzing the occurrences of words over time. Our new/re-emerging words detection algorithms are motivated by anomaly detection in the distribution of words or the occurrences of words in time domain. After identifying the words related to security events, our event detection algorithm clusters the triggered tweets for event construction. This approach enables to detect new and resurgent threats, regardless of their volume of mentions. We empirically demonstrate that the proposed event detection system performs promisingly over a wide range of cyber threat types.

REFERENCES

[1] [n.d.]. Statistical hypothesis testing. https://en.wikipedia.org/wiki/Statistical_hypothesis_testing.
[2] 2009. Evaluation of clustering. https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html.
[3] 2016. RSA Conference: Top 25 infosec leaders to follow on Twitter. https://techbeacon.com/rsa-conference-top-25-infosec-leaders-follow-twitter.
[4] 2016. Security experts you need to follow. https://www.securityinnovationeurope.com/blog/page/security-experts-you-need-follow.
[5] 2018. 2018 Internet Security Threat Report. https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-executive-summary-en.pdf.
[6] 2018. Revealed: The 21 biggest data breaches of 2018. https://www.digitalinformationworld.com/2018/12/biggest-data-breaches-of-2018.html.
[7] 2019. Best Buy: Shop all brands. https://www.bestbuy.com/.
[8] 2019. Common Attack Pattern Enumeration and Classification. https://capec.mitre.org/data/definitions/1000.html.
[9] 2019. Common Vulnerabilities and Exposures. https://cve.mitre.org/.
[10] 2019. Common Weakness Enumeration. https://cwe.mitre.org/data/definitions/1000.html.
[11] 2019. Consumer Reports Top Products. https://www.consumerreports.org/appliances/.
[12] 2019. ENISA Threat Taxonomy. https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/enisa-threat-landscape/threat-taxonomy/view.
[13] 2019. Fortune 500 Companies 2018. http://fortune.com/global500/.
[14] 2019. Hackmageddon. https://www.hackmageddon.com/.
[15] 2019. Kaspersky Lab DDoS Reports: DDoS Attacks in 2018. https://securelist.com/.
[16] 2019. NVD CPE dictionary. https://nvd.nist.gov/products/cpe.
[17] 2019. Popular jQuery JavaScript library impacted by prototype pollution flaw. https://www.zdnet.com/article/popular-jquery-javascript-library-impacted-by-prototype-pollution-flaw/.
[18] 2019. Recorded Future. https://www.recordedfuture.com/.
[19] 2019. Structured Threat Information eXpression (STIX) 1.x Archive Website. https://stixproject.github.io.
[20] 2019. Twitter Security Events. http://kb1.cse.ohio-state.edu:8123/events/hacked.
[21] 2019. Vulnerabilities Articles from Snyk. https://snyk.io/blog/category/vulnerabilities/.
[22] Farzindar Atefeh and Wael Khreich. 2015. A Survey of Techniques for Event Detection in Twitter. *Comput. Intell.* 31, 1 (Feb. 2015), 132–164. https://doi.org/10.1111/coin.12017
[23] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *The semantic web*. Springer, 722–735.
[24] Mateusz Fedoryszak, Brent Frederick, Vijay Rajaram, and Changtao Zhong. 2019. Real-time Event Detection on Social Data Streams. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. ACM, 2774–2782.
[25] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Philip S Yu, and Hongjun Lu. 2005. Parameter free bursty events detection in text streams. In *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment, 181–192.
[26] Qi He, Kuiyu Chang, and Ee-Peng Lim. 2007. Analyzing feature trajectories for event detection. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 207–214.
[27] Rupinder Paul Khandpur, Taoran Ji, Steve Jan, Gang Wang, Chang-Tien Lu, and Naren Ramakrishnan. 2017. Crowdsourcing cybersecurity: Cyber attack detection using social media. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 1049–1057.
[28] Jon Kleinberg. 2002. Bursty and Hierarchical Structure in Streams. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Edmonton, Alberta, Canada) *(KDD '02)*. ACM, New York, NY, USA, 91–101. https://doi.org/10.1145/775047.775061
[29] Quentin Le Sceller, ElMouatez Billah Karbab, Mourad Debbabi, and Farkhund Iqbal. 2017. SONAR: Automatic Detection of Cyber Security Events over the Twitter Stream. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*. ACM, 23.
[30] Edward Loper and Steven Bird. 2002. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1* (Philadelphia, Pennsylvania) *(ETMTNLP '02)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 63–70. https://doi.org/10.3115/1118108.1118117
[31] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (Baltimore, Maryland). Association for Computational Linguistics, 55–60. https://doi.org/10.3115/v1/P14-5010
[32] Michael Mathioudakis and Nick Koudas. 2010. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 1155–1158.
[33] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[34] Sudip Mittal, Prajit Kumar Das, Varish Mulwad, Anupam Joshi, and Tim Finin. 2016. Cybertwitter: Using twitter to generate alerts for cybersecurity threats and vulnerabilities. In *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE Press, 860–867.

[35] Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*. 380–390.

[36] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[37] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (New Orleans, Louisiana). Association for Computational Linguistics, 2227–2237. https://doi.org/10.18653/v1/N18-1202

[38] Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. Streaming first story detection with application to twitter. In *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics*. Association for Computational Linguistics, 181–189.

[39] Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 1524–1534.

[40] Alan Ritter, Evan Wright, William Casey, and Tom Mitchell. 2015. Weakly supervised extraction of computer security events from twitter. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 896–905.

[41] Carl Sabottke, Octavian Suciu, and Tudor Dumitras. 2015. Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting Real-World Exploits.. In *USENIX Security Symposium*. 1041–1056.

[42] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*. ACM, 851–860.

[43] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2013. Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Transactions on Knowledge and Data Engineering* 25, 4 (2013), 919–931.

[44] Anna Sapienza, Sindhu Kiranmai Ernala, Alessandro Bessi, Kristina Lerman, and Emilio Ferrara. 2018. Discover: Mining online chatter for emerging cyber threats. In *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 983–990.

[45] Alessio Signorini, Alberto Maria Segre, and Philip M Polgreen. 2011. The use of Twitter to track levels of disease activity and public concern in the US during the influenza A H1N1 pandemic. *PloS one* 6, 5 (2011), e19467.

[46] Zareen Syed, Ankur Padia, Tim Finin, M Lisa Mathews, and Anupam Joshi. 2016. UCO: A Unified Cybersecurity Ontology. In *AAAI Workshop: Artificial Intelligence for Cyber Security*.

[47] Jianshu Weng and Bu-Sung Lee. 2011. Event detection in twitter. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*. 401–408.

[48] Xiangmin Zhou and Lei Chen. 2014. Event detection over twitter social media streams. *The VLDB Journal* 23, 3 (2014), 381–400.

[49] Shi Zong, Alan Ritter, Graham Mueller, and Evan Wright. 2019. Analyzing the Perceived Severity of Cybersecurity Threats Reported on Social Media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 1380–1390.