Crowdsourcing Cybersecurity: Cyber Attack Detection using Social Media

Rupinder Paul Khandpur[1,2], Taoran Ji[1,2], Steve Jan[3],
Gang Wang[3], Chang-Tien Lu[1,2], Naren Ramakrishnan[1,2]
[1]Discovery Analytics Center, Virginia Tech, Arlington, VA 22203, USA
[2]Department of Computer Science, Virginia Tech, Arlington, VA 22203, USA
[3]Department of Computer Science, Virginia Tech, Blacksburg, VA 24060, USA

ABSTRACT

Social media is often viewed as a sensor into various societal events such as disease outbreaks, protests, and elections. We describe the use of social media as a crowdsourced sensor to gain insight into ongoing cyber-attacks. Our approach detects a broad range of cyber-attacks (e.g., distributed denial of service (DDoS) attacks, data breaches, and account hijacking) in a weakly supervised manner using just a small set of seed event triggers and requires no training or labeled samples. A new query expansion strategy based on convolution kernels and dependency parses helps model semantic structure and aids in identifying key event characteristics. Through a large-scale analysis over Twitter, we demonstrate that our approach consistently identifies and encodes events, outperforming existing methods.

CCS CONCEPTS

• **Information systems → Data mining**; **Information retrieval**; *Information retrieval query processing*; *Retrieval models and ranking*; *Similarity measures*; • **Computing methodologies → Information extraction**; *Natural language processing*; Artificial intelligence;

KEYWORDS

Social Media; Twitter; Dynamic Query Expansion; Event Detection; Cyber Security; Cyber Attacks

1 INTRODUCTION

Cyber-attacks are now widespread, e.g., most recently of the US Democratic National Committee and at companies such as Sony, Verizon, Yahoo, Target, JP Morgan, Ashley Madison as well as at government agencies such as the US Office of Personnel Management. Consequences and implications of cyber-attacks range from data leaks about sensitive personal information about users to potential to cause loss of life and disruptions in critical infrastructure. To develop adequate cyber-defenses it is imperative to develop good 'ground truth', i.e., an authoritative record of cyber incidents reported in the media cataloged alongside key dimensions. Availability of high quality ground truth events can support various analytics efforts, e.g., identifying precursors of attacks, developing predictive indicators using surrogate data sources, and tracking the progression of events over space and time.

It has been well argued that, because news about an organization's compromise sometimes originates *outside* the organization, one could use open source indicators (e.g., news and social media) as indicators of a cyber-attack. Social media, in particular, turns users into social sensors empowering them to participate in an online ecosystem of event detection for happenings such as disease outbreaks [31], civil unrest [20, 37], and earthquakes [29]. While the use of social media cannot fully supplant the need for internal telemetry for certain types of attacks (e.g., use of network flow data to detect malicious network behavior [4, 12, 21]), analysis of such online media can provide insight into a broader range of cyber-attacks such as data breaches, account hijacking and newer ones as they emerge.

At the same time it is non-trivial to harness social media to identify cyber-attacks. Our objective is to detect a range of different cyber-attacks as early as possible, determine their characteristics (e.g., the target, the type of attack), in an weakly supervised manner without any requirement for training phase or labeled samples. Prior work (e.g., [27]) relies on training with annotated samples with fixed feature sets which will be unable to capture the dynamically evolving nature of cyber-attacks over time and are also unable to encode characteristics of detected events, as we aim to do here.

Our main contributions are:

- **A framework for cybersecurity event detection from online social media.** We propose a dynamic typed query expansion approach that requires only a small set of general seed event triggers and learns to map them to specific event-related expansions and thus provide situational awareness into cyber-events in an unsupervised manner.
- **A novel query expansion strategy based on dependency tree patterns.** To model typical reporting structure in how cyber-attacks are described in social media, we propose a dynamic event trigger expansion method based on convolution kernels and dependency parses. The proposed approach also employs a word embedding strategy to capture similarities between event triggers and candidate event reports.
- **Extensive empirical evaluation for three kinds of cyber-attacks**. We manually catalog ground truth for three event classes—distributed denial of service (DDoS) attacks, data breaches, and account hijacking—and demonstrate that our

CIKM'17, , November 6–10, 2017, Singapore.
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3132866

approach consistently identifies and encodes events outperforming existing methods.

2 PROBLEM SETUP

The input to our methodology is a collection of time-ordered tweets $\mathbb{D} = \{\mathbb{D}_1, \mathbb{D}_2, \ldots, \mathbb{D}_p\}$ organized along p time slots. Let $\mathcal{D}$ denote the tweet space corresponding to a subcollection $\mathbb{D}_i$, let $\mathcal{D}_+$ denote the target tweet subspace (in our case, comprising cyber-attack events), and let $\mathcal{D}_- = \mathcal{D} - \mathcal{D}_+$ denote the rest of the tweets in the considered tweet space.

Definition 1. Typed Dependency Query: A ***typed dependency query*** is a linguistic structure that characterizes a semantically coherent event related topic. Different from n-grams, terms contained in a ***typed dependency query*** share both syntactic and semantic relationships. Mathematically, a ***typed dependency query*** is formulated as a tree structure $G = \{V, E\}$, where node $v \in V$ can be either a word, user mention, or a hashtag and $\varepsilon \in E$ represents a syntactic relation between two nodes.

Definition 2. Seed Query: A ***seed query*** is a manually selected typed dependency query targeted for a certain type of event. For instance, "hacked account" can be defined as a potential ***seed query*** for an account hijacking event.

Definition 3. Expanded Query: An ***expanded query*** is a typed dependency query which is automatically generated by the dynamic query expansion algorithm based on a set of seed queries and a given tweet collection $\mathcal{D}$. The ***expanded query*** and its seed query can be two different descriptions of the same subject. More commonly, an ***expanded query*** can be more specific than its seed query. For instance, "prime minister dmitry medvedev twitter account hack", an expanded query from "hacked account", denotes the message of an account hijacking event related with Russian Prime Minister Dmitry Medvedev.

Definition 4. Event Representation: An event e is defined as $(Q_e, date, type)$, where Q_e is the set of event-related expanded queries, $date$ denotes when the event happens, and $type$ refers to the category of the cyber-attack event (i.e., DDoS, account hijacking, or data breach).

Here Q_e is a defined as a set because, in general, a cyber-attack event can be presented and retrieved by multiple query templates. For instance, among online discussion and report about event "Fashola's account, website hacked", the query template most used are "fashola twitter account hack", "fashola n78m website twitter account hack" and "hack account".

Given the above definitions, the major tasks underlying the cyber-attack event detection problem are defined as follows:

Task 1: Target Domain Generation. Given a tweet subcollection $\mathcal{D}$, ***target domain generation*** is the task of identifying the set of target related tweets $\mathcal{D}_+$. $\mathcal{D}_+$ contains critical domain relevant information based on which the expanded queries can be mined.

Task 2: Expanded Query Extraction. Given target domain $\mathcal{D}_+$, the task of ***expanded query extraction*** is to generate a set of expanded queries $Q = \{q_1, \ldots, q_n\}$ which represents the relevant concepts delivered by $\mathcal{D}_+$. Thus set Q can be used to retrieve event related information from other collection sets.

Task 3: Dynamic Typed Query Expansion. Given a small set of seed queries Q^0 and a twitter collection $\mathcal{D}$, the task of ***dynamic typed query expansion*** is to iteratively expand $\mathcal{D}_+^k$ and Q^k until all the target related messages are included.

3 METHODOLOGY

In traditional information extraction (IE), a large corpus of text must first be annotated to train extractors for event triggers, defined as main keywords indicating an event occurrence [8]. However, in our scenario using online social media, a manually annotated label set is impractical due to the huge volume of online media and the generally noisy characteristics of social media text. In this section, we discuss in detail the key components of our system, illustrated in Fig. 1, to automatically mine cybersecurity related queries over which the event tracking is performed.

3.1 Target Domain Generation

In this subsection, we describe our target domain generation wherein crowdsourced social indicators (tweets) of cyber-attack events are retrieved. Given a query and a collection of tweets $\mathcal{D}$, the typical way to retrieve query-related documentation is based on a bag of words model [30] which comes with its attendant disadvantages. Consider the following two tweets: "have Riseup servers been compromised or **data leaked**?" and "@O2 You completely screwed me over! My phones back on, still **leaking data** and YOU are so UNHELPFUL #CancellingContract #Bye". Though the important indicator "data leak" for a data breach attack is mentioned in both tweets, the second tweet is complaining about a phone carrier and would be considered noise for the cybersecurity domain. To address this problem, syntactically bound information and semantic similarity constraints are jointly considered in our proposed method.

More specifically, each tweet in $\mathcal{D}$ is first converted into its dependency tree form. Thus for a given seed query q, the target domain $\mathcal{D}_+ \subseteq \mathcal{D}$ can be generated by collecting all tweets which are both syntactically and semantically similar to the seed query q. Mathematically, given the two dependency trees q and $d \in D$, a convolution tree kernel [9] is adopted to measure the similarity by computing all the common paths between two trees:

$$\mathrm{K}(q, d) = \sum_{\substack{u \in q \\ v \in d}} \Big(1 + \mathcal{H}(u, v)\Big)^{\mathbb{1}_{\mathbb{R}_{>0}}\big(\mathcal{H}(u,v)\big)} \tag{1}$$

where v and u are nodes from two trees q and d respectively, $\mathbb{R}_{>0}$ represents the set of positive real numbers, $\mathbb{1}(\cdot)$ is the indicator function, and $\mathcal{H}(v, u)$ counts the number of common paths between the two trees which peak at v and u, which can be calculated by an efficient algorithm proposed by Kate et al. [9], as described in Algorithm 1.

In Algorithm 1, $\lambda \in (0, 1]$ (set to 0.5) is a parameter used to down-weight the contribution of long paths, $\kappa(u, v)$ is the number of common paths between the two trees which originate from u and v, and can be recursively defined as:

$$\kappa(u, v) = \sum_{\substack{\mu \in C(u) \\ \eta \in C(v)}} (1 + \kappa(\mu, \eta))^{\mathbb{1}_{\mu \doteq \eta}(u, v)}, \tag{2}$$

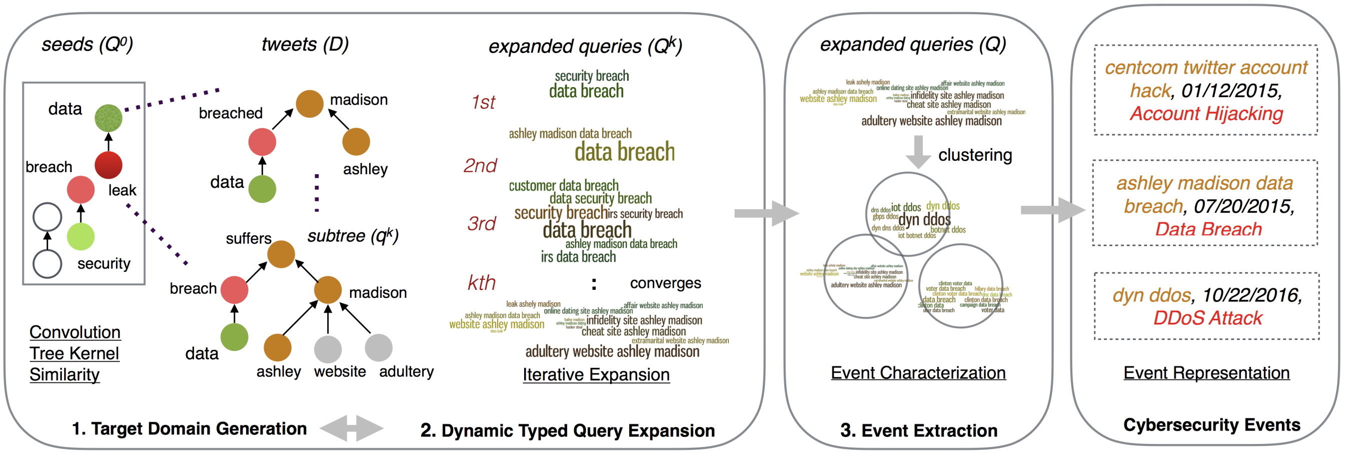

Figure 1: A schematic overview of the cybersecurity event detection system.

Algorithm 1: Compute all common paths.

```
   Input: $u \in q,\ v \in d$
   Output: $\mathcal{H}(u, v)$
1  Set $r = \kappa(u, v)$;
2  Set $C_u = children(u)$;
3  Set $C_v = children(v)$;
   // consider every pair of common child nodes
4  for $c_i, c_j \in C_u,\ i \neq j$ do
5      for $c_m, c_n \in C_v,\ m \neq n$ do
6          if $c_i \doteq c_m$ and $c_j \doteq c_n$ then
               // compute all common paths from children
7              $x = \kappa(c_i, c_m)$;
8              $y = \kappa(c_j, c_n)$;
9              $r = r + \sqrt{\lambda} + \lambda x + \lambda y + \lambda xy$ // increment
10 $\mathcal{H}(u, v) = r$;
```

where $C(\cdot)$ denotes the set of children node. This reinforces the common paths which are linguistically meaningful thereby reducing the noise introduced by coincidentally matched word chains. In addition the occurrence of long-range dependencies between words, for example "Ashley Madison" → "breached" as shown in Fig. 2, which may decrease the kernel performance when considering a fixed-window, are avoided because functionally related words are always directly linked in a dependency tree.

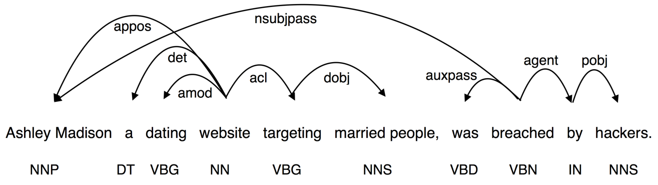

Figure 2: A dependency tree diagram of a cyber-attack related tweet illustrating short-range (local) and long-range word-word dependencies.

A key novelty of our convolution tree kernel is the use of a word embedding approach (instead of exact string matching) to compare the similarity between two words. In both (line 6) Algorithm 1 and Equation 2, we use the similarity operator ≐ to denote the comparison of word representations in vector space. This allows words that occur in similar contexts (that have similar embeddings) to be semantically compared (as measured by cosine similarity). This in turn helps improve recall & precision of target domain generation, by allowing the kernel to explore longer paths for example when comparing "data"←"leak" with "data"←"breach" but reject paths such as when comparing "website"←"hack" with "life"←"hack".

3.2 Dynamic Typed Query Expansion

In this subsection, we propose a way to dynamically mine an expanded query given a small collection of seed queries, as shown in Table 1. By providing a small set of seed queries (unigrams), Zhao et al. [37] proposed a dynamic query expansion (DQE) method which is able to iteratively expand the seed query set from a currently selected target tweet subspace until convergence. Looking beyond the simple unigram-based expansion, we propose a *dynamic typed query expansion* method that uses dependency-based tree structures for expansion. Our choice of cyber-attack categories is based on two assumptions: first, these capture over 80% of the yearly, cyber-attack occurrences [1] and second, that these categories have a direct impact on users, and thus crowdsourcing and social media have the potential to capture them. The detection of other attacks such as insider attacks or network intrusion still rely on traditional network-based approach. Nevertheless, even though the types of attacks focused by our paper can be viewed as "user-centric attacks" our approach (described below) can be extended to other categories.

Let us denote Q^k, $\mathcal{D}_+^k$ as the expanded query set and target domain at the kth iteration. At the start of the iteration, Q^0 is initialized with a small set of domain-relevant seed queries, as shown in Table 1. With $\mathcal{D}$ and Q^0, then $\mathcal{D}_+^0$, the target domain (at iteration 0) tweets are retrieved using the kernel similarity function described

[1] http://www.hackmageddon.com/category/security/cyber-attacks-statistics/

Table 1: Seed queries for cyber-attack events.

Category	Seed Query
Data breach	data leak, security breach, information stolen, password stolen, hacker stole
DDoS	DDoS attack, slow internet, network infiltrated, malicious activity, vulnerability exploit, phishing attack
Account Hijacking	unauthorized access, stolen identity, hacked account

in Equation 1. At the kth iteration, given the last expanded query set Q^{k-1} and last generated target domain $\mathcal{D}_+^{k-1}$, our approach first prepares candidate expanded queries for each matched $q_i \in Q^{k-1}$ and $d \in \mathcal{D}_+^{k-1}$:

$$\hat{q}_i^k = \text{subtree}\Big(\underset{v \in d}{\text{argmax}}\big(\sum_{u \in q_i} \mathcal{H}(v, u)\big)\Big), \tag{3}$$

where v and u are term nodes in tweet d and q_i respectively, and subgraph$(\cdot)$ is an operator to extract the subtree structure from entire tree with v as root. Thus the candidate query expansions are collected based on the relevant document and query space, that is $\mathcal{D}_+^{k-1}$ and Q^{k-1}. To identify the best (candidates) expanded queries, query terms are then ranked based on the Kullback-Leibler divergence [18] between the target domain $\mathcal{D}_+^{k-1}$ and the whole tweet collection $\mathcal{D}$:

$$KL(f, \mathcal{D}_+^{k-1} | \mathcal{D}) = \log \frac{\Pr(f | \mathcal{D}_+^{k-1})}{\Pr(f | \mathcal{D})} \Pr(f | \mathcal{D}_+^{k-1}), \tag{4}$$

where $KL(f, \mathcal{D}_+^{k-1} | \mathcal{D})$ denotes the Kullback-Leibler divergence, f is a term in $\hat{q}_i^k$, $\Pr(f | \mathcal{D}_+^{k-1})$ and $\Pr(f | \mathcal{D})$ is the probability of term f appearing in $\mathcal{D}_+^{k-1}$ and $\mathcal{D}$, respectively. Using the KL divergence to rank query terms we are able to assign scores to terms that best discriminate relevant and non-relevant expansions. For example query terms such "account" and "twitter" both appear frequently in the candidate expansions but they have little informative value as they will have a similar (random) distribution in any subset of the twitter collection, whereas terms such as "hacked" will have comparatively higher probability of occurrence in the relevant subspace. These high ranked candidates will then act as the expanded queries set to run the next iteration until the algorithm converges.detailed dynamic typed query expansion algorithm is shown in Algorithm 2.

3.3 Event Extraction

Given an expanded query set Q, we extract $Q_s \mid q_i \not\subseteq q_j \mid q_i, q_j \in Q_s$. For example, if the surface string representations of a set of expanded queries Q is ("data breach", "data leak", "ashley madison", "ashley madison data breach") then Q_s will be ("ashley madison data breach"). We then cluster the query expansions in Q_s using affinity propagation [7] and also extract *exemplars* q_e of each query set Q_e that are representative of clusters, where each member query is represented by a vector $\tilde{\boldsymbol{q}}$ calculated from the word embedding $\tilde{\boldsymbol{u}}$ of the lemma of each query term $u \in q$ as:

$$\tilde{\boldsymbol{q}} = \sum_{u \in q} \tilde{\boldsymbol{u}}. \tag{5}$$

Algorithm 2: Dynamic Typed Query Expansion Algorithm.

```
Input: Seed Query Set $Q^0$, Twitter sub-collection $\mathcal{D}$
Output: Expanded Query Set $Q$
Set $D_+^0 = match(Q^0, \mathcal{D}), k = 0$
repeat
    $k = k + 1$
    for $q_i \in Q^{k-1}, d \in \mathcal{D}_+^{k-1}$ do
        $\widehat{q}_i^k = subtree(\text{argmax}_{v \in d} \sum_{u \in q_i} CPP(v, u))$ // new candidate
        for $f \in \widehat{q}_i^k$ do
            $\Pr(f | \mathcal{D}_+^{k-1}) = \frac{tf(f)}{|\mathcal{D}_+^{k-1}|}$
            $\Pr(f | \mathcal{D}) = \frac{tf(f)}{|\mathcal{D}|}$
            $w(f) = KL(f, \mathcal{D}_+^{k-1} | \mathcal{D})$ // feature score
        $w(\widehat{q}_i^k) = \sum_{f \in \widehat{q}_i^k} w(f)$ // query score
    $Q^k = topN(w(\widehat{Q}^k)), \widehat{Q}_k = \{\widehat{q}_1^k, \ldots, \widehat{q}_{|\widehat{Q}^k|}^k\}$
    $\mathcal{D}_+^k = match(Q^k, \mathcal{D})$ // filter new target subspace
until $\bigcup_{i=0}^{k} Q^i - \bigcup_{i=0}^{k-1} Q^i \neq \emptyset$ // DQE iteration;
$Q = Q^k$
```

Each exemplar query q_e is then annotated to a cyber-attack type. For this purpose, we first compute the cosine similarity between an exemplar query expansion q_e and seed query $q_j \in Q^{(0)}$ as:

$$\text{sim}(q_e, q_j) = \frac{\tilde{\boldsymbol{q}}_e \cdot \tilde{\boldsymbol{q}}_j}{||\tilde{\boldsymbol{q}}_e|| \cdot ||\tilde{\boldsymbol{q}}_j||}. \tag{6}$$

The $q_j \in Q^{(0)}$ which has the highest similarity value with q_e determines the event type to which Q_e belongs to. For the complete event representation $(Q_e, date, type)$ date information is extracted based on the time interval chosen for DQE; for example in our experiments we run DQE on a daily aggregated collection of tweets. In this way we extract the final set of event tuples.

4 EVALUATION

4.1 Evaluation Setup

4.1.1 Dataset and Gold Standard Report. We evaluate the proposed method on a large stream of tweets from GNIP's decahose (10% sample) collected from August 2014 through October 2016. The total raw volume of our Twitter dataset across these 27 months is 5,146,666,178 (after removing all retweets and non-English tweets). Then, from this raw volume we create two subset collections:

- **Fixed Keyword Filtered Tweets:** We filtered 79,501,789 tweets that contain at least one matching term from a list of cyber-attack related keywords. These are top 1000 keywords (ranked by TFIDF) extracted from description texts of events in our gold standard report (see below).
- **Normalized Tweet Texts:** We extract and normalize tweet texts (after removing accents, user mentions and urls) to produce a collection of 3,267,567,087 unique texts to train a 200 dimensional word embedding via Gensim's word2vec software [25].

Note that the experimental results for the performance of our event detection approach are done using the entire raw volume of over

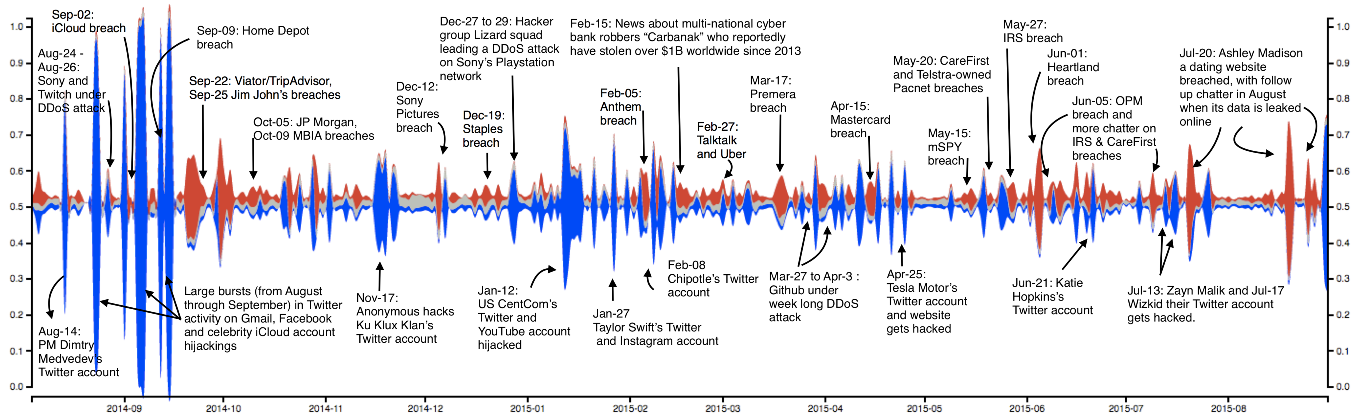

Figure 3: Streamgraph showing normalized volume of tweets (August 2014 through August 2015) tagged with data breach (red), DDoS activity (grey) and account hijacking (blue) types of cyber-security events.

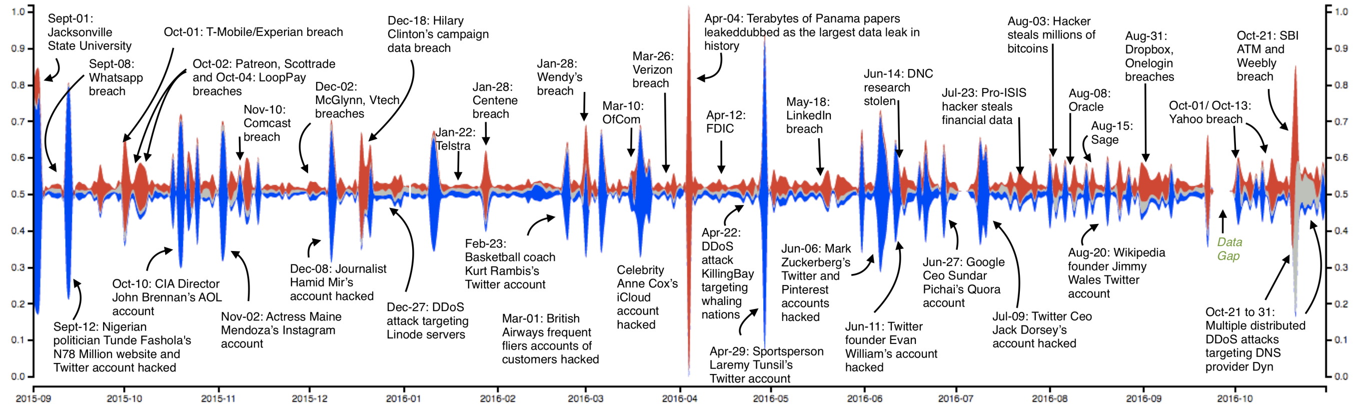

Figure 4: Streamgraph showing normalized volume of tweets (September 2015 through October 2016) tagged with data breach (red), DDoS activity (grey) and account hijacking (blue) types of cyber-security events.

5 billion tweets. The total volume of tweets filtered from query expansion algorithm is 1,093,716 over the entire time period.

To evaluate our methods, we organize a Gold Standard Report (GSR) on cyber security incidents to serve as a ground truth database. In particular, we focus on high impact events about data breach, DDoS and account hijacking incidents based on two different sources: Hackmageddon[2] and PrivacyRights[3]. In both sources, each event is characterized by an event type, date (when the event was publicly reported), victim organization(s), and a short description.

- **Hackmageddon** is an independently maintained website that collects public reports of cybersecurity incidents. Between January 2014 and December 2016, we extract 295 account hijacking events and 268 DDoS events. For account hijacking, since we are using social media data, we mainly focus on hijacking attacks on social media accounts (Twitter, Instagram, Facebook) by cyber crimes. After filtering US-based events and matching the time range of our Twitter data, we obtain 55 account hijacking events and 80 DDoS events to include in the GSR.
- **PrivacyRights** is a highly reputable repository for data breach incident reports. Between January 2014 and December 2016, we extract 1064 reported data breach events. To enhance the accuracy of GSR, we choose events reported by four large, well-known sources — "Media", "KrebsOnSecurity", "California Attorney General", and "Security Breach Letter". Then, we filter out data breaches involving non-cyber reasons (*e.g.*, physical theft) and focus on the HACK category. After removing events with an unknown size of data loss, and matching the time range with our data, we have 85 data breach events for inclusion in the GSR.

4.1.2 Baselines and Comparison Methods. We use two Twitter-based baselines to independently evaluate the performance of our target domain and cyber-attack detection methods:

(1) *Target Domain Generation using Expectation Regularization (baseline 1)* [27]: This baseline is trained based on a small

[2] http://hackmageddon.com
[3] https://www.privacyrights.org/data-breaches

Table 2: A sample of negative instances for cyber-attack events used in the evaluation of target domain generation methods.

Event Entity	Date	Sample Tweet
white house	2014-08-08	Toddler causes perimeter **breach** at White House
green zone	2016-04-30	Anti-Government Protesters **Breach** Baghdad's Green Zone
avijit roy	2015-02-27	American-Bangladeshi blogger Avijit Roy **hacked** to death by Islamist extremists
jessica jones	2016-01-13	NBC thinks it's **hacked** Netflix's ratings, says 'Jessica Jones' bests 'Master of None'
arbor networks	2015-03-25	Arbor Networks, Cisco partner on **DDoS** protection
zenedge	2016-07-30	ZENEDGE Debuts Always-On **DDoS** Protection #Bitcoin

Table 3: Contingency table used to assess cyber-attack related tweet detection results.

Method	Data Breach				DDoS				Account Hijacking			
	TP	FP	FN	TN	TP	FP	FN	TN	TP	FP	FN	TN
Typed DQE	1110	389	528	1085	516	129	93	30	2028	1	2976	200
Baseline 1 [27]	1526	1391	112	83	295	113	314	46	2182	29	2322	172

set of seed events for each type of attack. For training, we randomly selected 10 ground-truth events for each of the three attack types (from our GSR). Additionally, keeping the similar proportions of per attack-type events, in the test phase we also included several negative sample events (as shown in Table 2) from manual search.
Following the dataset preparation process described in [27], we retained only those event-related tweets that contained keywords - "hacked" (for account hijacking events), "breach" (data breach events), and "ddos" (DDoS events). Following this step, the feature set was generated by collecting a window of contextual words and POS tags surrounding the seed event keyword, where this window size was set to 4 in our evaluation, the target expectation was set to 0.55, l_2 regularization term to 100, and expectation regularization term λ^U set to 10 times the number of labeled samples. In total, we collected 8943 and 8585 tweet samples for training and testing, respectively. Further, we were able to extract 8969, 6178 and 10760 features from data breach, DDoS, and account hijacking event related tweets, respectively.

(2) *Cyber-attack Event Detection using Bursty Keywords (baseline 2)* [11]: This baseline method identifies time periods in which a target event is uncharacteristically frequent or bursty on a set of static keywords. An event is extracted if the size of this set of bursty keywords is larger than a threshold T_b. In this experiment, we use the 79.5 million Fixed Keyword Filtered Tweets and the 1000 static keywords to apply the baseline method. We set the threshold T_b based on small scale empirical tests on a few months of data, and manually examine the detected events. We set T_b =36 which returns a better event/noise ratio. We apply this threshold on all the data and detects 81 events from August 2014 through October 2016. Each detected event is characterized by a date and a set of bursty keywords.

4.1.3 Matching Detected Events with GSR. Given a detected event presented by $e = (Q_e, date, type)$, we developed a semi-automatic method to detect if e is matched with any event in the GSR:

(1) For named entity in e, we check if it matches any event description in GSR and obtain a matched collection from GSR, say *ME*;
(2) Further filter *ME* by matching the event date between *date* in e and *ME*, with a time window as 3 (one day before *date*, and one day after *date*), and obtain a new filtered event set, say *FME*;
(3) Compare the event type between e and event in *FME*; if the event type also matches, then event e is consider as a matched event.

However considering that the detected events are mined from the Twitter environment which may not use formal keywords to describe the event. We also manually double check the event e if it fails step 1. Detected events by the baseline method use the same approach to match against GSR. The only adjustment is to match the bursty keywords of the detected events instead of named entities.

4.2 Measuring Performance

Target Domain Generation. In terms of precision and recall (see Table 4), our approach achieves a 70% accuracy in extracting target domain tweets, outperforming the comparison to baseline 1 [27] in two categories, viz. data breach and DDoS. In case of account hijacking our accuracy is only slightly less due to our lower recall, because TypedDQE will reject tweets by way of down-ranking expansion candidates that are not specific enough (for example if they contain only one keyword such as "hacked") and are below a certain support threshold. The use of kernel similarity (as opposed to fixed context window) provides higher precision, seen clearly from Table 3 where our approach detected only 1 tweet incorrectly as account hijacking-related in comparison to 29 *false positives* by the baseline. Also worth noting is the high specificity (*true negative rate*) of 71% as compared to baseline's 16%.

Cyber-attack Event Detection. Precision and recall over different types of cyber-attack events are summarized in Table 4 using a second baseline [11]. These results show that with only a small set of seed queries (as shown in Table 1), our approach can obtain

Table 4: Overall evaluation of cyber-attack detection.

Method	Data Breach			DDoS			Account Hijacking		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Typed DQE	0.74	0.68	0.71	0.80	0.85	0.82	0.99	0.45	0.61
Baseline 1 [27]	0.52	0.93	0.67	0.72	0.48	0.58	0.99	0.48	0.65
Baseline 2 [11]	0.21	0.20	0.20	0.01	0.02	0.01	0.01	0.01	0.01

Table 5: Matching the detected events with the GSR.

Matched with GSR	Data Breach		DDoS		Account Hijacking	
	TP	FP	TP	FP	TP	FP
Yes	22	0	20	0	8	0
No	156	49	29	12	51	31

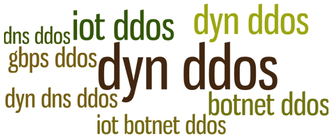

Figure 5: Query expansions (size is proportional to the query's feature score) produced on October 22 2016 for the DDoS attack on DNS provider Dyn.

around 80% of precision for data breach and DDoS events. This means our approach is able to handle the noisy Twitter environment and perform cyber-attack event detection accurately. The precision for account hijacking is not as high (66%). On manual analysis (using online search) we identified several events detected by our system even though they were not covered by the GSR. We show this disparity in Table 5 where we can notice that it is highest for account hijacking type of events (as social media tends to provide greater coverage to celebrity and other individual cases of hacked accounts). Data breach events have a higher recall (75%). The relatively low recall for account hijacking and DDoS is explainable. Both DDoS and account hijacking events have a rather short life cycle from occurrence to being addressed. Thus their signal in social media is relatively weaker. For instance, DDoS events often result in several minutes to a few hours of slow Internet access, and thus may end even before people realize it. This intuition is validated in the baseline performance. The extremely low precision and recall shows that relying on burstiness is difficult to capture such events, possibly due to their weak signals over noise.

4.3 Case Studies

We comprehensively show in Fig. 3 and Fig. 4, the wide range of events that our system is able to detect. Notice the clear bursts in Twitter activity that our query expansion algorithm is able to detect. Through the following case studies we highlight some of the interesting cases.

Targeted DDoS Attacks on Dyn. In late November 2014, a hacker group calling itself "The Guardians Of Peace" hacked their way into Sony Pictures, leaving the Sony network crippled for days. We capture 12 separate events of DDoS attacks including four in the last week of August 2014, beginning with the first on August 24th. Further in 2015, more ensuing attacks are captured, one highlighted by the data breach of their movie production house (on December 12th) and then a massively crippling targeted DDoS attack on their PlayStation network in late December 2015. Another noteworthy case of DDoS attacks in 2016 is the multiple distributed denial-of-service attack on DNS provider "Dyn" from October 21st through 31st in 2016, that almost caused an worldwide Internet outage. Our system generates several query expansions, shown in Fig. 5 which clearly characterizes the nature of these attacks where the hackers turned a large number of Internet-connected devices around the world into botnets executing a distributed attack.

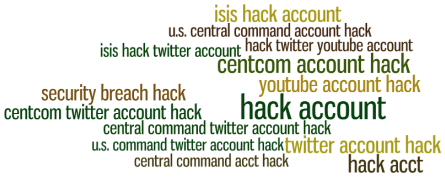

Figure 6: Query expansions (size is proportional to the query's feature score) produced from the U.S CentCom Twitter account hijacking event.

Ashley Madison Website Data Breach. In July 2015 a group calling itself "The Impact Team" stole the user data of Ashley Madison, an adult dating website. The hackers stole all the website's customer data and threatened to release personally identifying information if the site was not immediately shut down. Between 18th and 20th August, the group leaked more than 25 gigabytes of company data. The word clouds in Fig. 7 clearly show how our method iteratively expands from the seed queries to the expanded queries in the last iteration (iteration 3) capturing rich characteristics of the breach. After the initial burst as seen in the figure, we also see a second corresponding burst a month later (on August 19th) when the user data was released online.

Twitter Account Hijackings. We were also able to detect with very high date accuracy, several high profile cases of account hijackings of social media accounts of known personalities and government institutions including the Twitter account for U.S. Central Command which was hacked by ISIS sympathizers on January 12th, 2015. We show in Fig. 6 that our method not only identifies the victim ("central command twitter account hack") but also the actor who perpetrated the hacking ("isis hack twitter account").

5 RELATED WORK

Cyber-attack Detection and Characterization. Detecting and characterizing cyber-attacks is highly challenging due to the

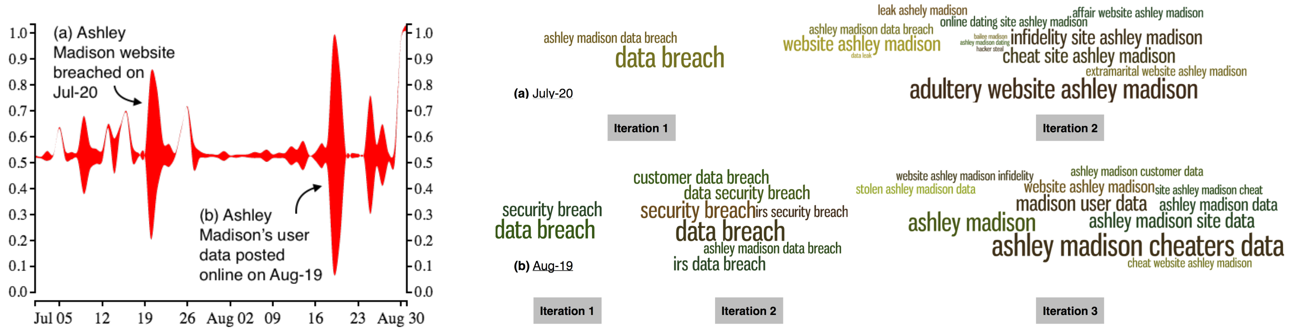

Figure 7: Ashley Madison website data breach. The streamgraph shows the bursty normalized volume of tweets related to the data breaches. Along with all the query expansions (size is proportional to the query's feature score) produced at each iteraton of TypedDQE.

constant-evolving nature of cyber criminals. Recent proposals cover a large range of different methods, and Table 6 lists representative works in this space. Earlier work primarily focuses on mining network traffic data for intrusion detection. Specific techniques range from classifying malicious network flows [13] to anomaly detection in graphs to detecting malicious servers and connections [4, 5, 12, 21]. More recently, researchers seek to move ahead to predict cyber-attacks before they happen, for early notifications [14]. For example, Liu et al. leverage various network data associated to an organization to look for indicators of attacks [16, 17]. By extracting signals from mis-configured DNS and BGP networks as well as spam and phishing activities, they build classifiers to predict if an organization is (or will be) under attack. Similarly, Soska et al. apply supervised classifiers to network traffic data to detect vulnerable websites, and predict their chances of turning malicious in the future [32].

In recent years, online media such as blogs and social networks have become another promising data source of security intelligence [19, 36]. Most existing work focuses on technology blogs and tweets from *security professionals* to extract useful information [34]. For example, Liao et al. builds text mining tools to extract key attack identifiers (IP, MD5 hashes) from security tech blogs [15]. Sabottke et al. leverage Twitter data to estimate the level of interest in existing CVE vulnerabilities, and predict their chance of being exploited in practice [28]. Our work differs from existing literature since we focus on crowdsourced data from the much broader user population who are likely the *victims* of security attacks. The most related work to ours is [27] which uses weakly supervised learning to detect security related tweets. However, this technique is unable to capture the dynamically evolving nature of attacks and is unable to encode characteristics of detected events.

Event Extraction and Forecasting on Twitter. Another body of related work focuses on Twitter to extract various events such as trending news [1, 26], natural disasters [29], criminal incidents [35] and population migrations [23]. Common event extraction methods include simple keyword matching and clustering, and topic modeling with temporal and geolocation constrains [3, 33, 38]. Event forecasting, on the other hand, aims to predict future evens based on early signals extracted from tweets. Example applications include detecting activity planning [2] and forecasting future events such as civil unrest [24] and upcoming threats to airports [10]. In our work, we follow a similar intuition to detect signals for major security attacks. The key novelty in our approach, different from these works, is the need for a typed query expansion strategy that provides both focused results and aids in extracting key indicators underlying the cyber-attack.

6 CONCLUSION

We have demonstrated a weakly supervised approach with no training phase requirement to dynamically extract and encode cyber-attacks reported and discussed in social media. We have motivated the need for a careful structured query expansion strategy, and how the use of dependency parse trees and word embeddings supports context-rich retrieval. Our retrieval algorithm can be easily extended to other languages (by training additional embedding models) and different data sources of real-time, text streams such as users' status updates and blog posts commonly found in several online social networks other than Twitter. We have performed a comprehensive evaluation of our approach achieving over 70% accuracy in retrieving cyber-attacks related content from social media streams, and 80% precision in successfully detecting cyber security related events, outperforming the two considered baselines. Given the widespread prevalence of cyber-attacks, tools such as presented here are crucial to providing situational awareness on an ongoing basis. Future work is aimed at broadening the class of attacks that the system is geared to as well as at modeling sequential dependencies (from occurrence to reporting) of cyber-attacks. This will aid in capturing characteristics such as the increased prevalence of attacks on specific institutions or countries during particular time periods.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the funding support for this research from National Science Foundation (under award numbers CNS-1717028, DGE-1545362, IIS-1633363) and Northrop Grumman (NGC 7500144847).

Table 6: Comparison of our work to past research.

	Method			Event			Goal	Data
	No Training	Keyword Expansion	Information Extraction	Characterize Event	Type	Detection		
[16]	✗	✗	✗	✗	✗	✓	Cyberattacks	Network Data
[27]	✗	✓	✓	✗	✓	✓	Cyberattacks	Twitter
[22]	✗	✗	✗	✗	✗	✗	Cyberattacks	WINE
[39]	✗	✗	✓	✗	✗	✗	Malware	Papers
[12]	✗	✗	✗	✗	✗	✗	Malware	WINE
[28]	✗	✗	✓	✗	✗	✓	Vulnerability	Twitter
[5]	✓	✗	✗	✗	✗	✗	Intrusion	Network Data
[21]	✓	✗	✗	✗	✗	✗	Intrusion	Network Data
[6]	✓	✗	✗	✗	✗	✗	Intrusion	Network Data
[4]	✓	✗	✗	✗	✗	✗	Insider	Access Log
[15]	✓	✗	✓	✗	✗	✗	IOC	Tech Blogs
Ours	✓	✓	✓	✓	✓	✓	Cyberattacks	Twitter

REFERENCES

[1] Farzindar Atefeh and Wael Khreich. 2015. A Survey of Techniques for Event Detection in Twitter. *Comput. Intell.* 31, 1 (2015), 132–164.

[2] Hila Becker, Dan Iter, Mor Naaman, and Luis Gravano. 2012. Identifying Content for Planned Events Across Social Media Sites. In *Proc. WSDM'12*.

[3] Hila Becker, Mor Naaman, and Luis Gravano. 2012. Beyond Trending Topics: Real-World Event Identification on Twitter. In *Proc. ICWSM'14*.

[4] Michael Davis, Weiru Liu, Paul Miller, and George Redpath. 2011. Detecting Anomalies in Graphs with Numeric Labels. In *Proc. CIKM'11*.

[5] Qi Ding, Natallia Katenka, Paul Barford, Eric Kolaczyk, and Mark Crovella. 2012. Intrusion As (Anti)Social Communication: Characterization and Detection. In *Proc. KDD'12*.

[6] W. Eberle and L. Holder. 2007. Discovering Structural Anomalies in Graph-Based Data. In *Proc. ICDMW'07*.

[7] Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science* 315, 5814 (2007), 972–976.

[8] Heng Ji, Ralph Grishman, et al. 2008. Refining Event Extraction through Cross-Document Inference. In *Proc. ACL'08*.

[9] Rohit J Kate. 2008. A dependency-based word subsequence kernel. In *Proc. EMNLP'08*.

[10] Rupinder P. Khandpur, Taoran Ji, Yue Ning, Liang Zhao, Chang-Tien Lu, Erik R. Smith, Christopher Adams, and Naren Ramakrishnan. 2016. Determining Relative Airport Threats from News and Social Media. In *Proc. AAAI'16*.

[11] Jon Kleinberg. 2002. Bursty and Hierarchical Structure in Streams. In *Proc. KDD'02*.

[12] Bum Jun Kwon, Jayanta Mondal, Jiyong Jang, Leyla Bilge, and Tudor Dumitras. 2015. The Dropper Effect: Insights into Malware Distribution with Downloader Graph Analytics. In *Proc. CCS'15*.

[13] Wenke Lee and Salvatore J. Stolfo. 1998. Data Mining Approaches for Intrusion Detection. In *Proc. USENIX Sec'98*.

[14] Frank Li, Zakir Durumeric, Jakub Czyz, Mohammad Karami, Michael Bailey, Damon McCoy, Stefan Savage, and Vern Paxson. 2016. You've Got Vulnerability: Exploring Effective Vulnerability Notifications. In *Proc. USENIX Sec'16*.

[15] Xiaojing Liao, Kan Yuan, XiaoFeng Wang, Zhou Li, Luyi Xing, and Raheem Beyah. 2016. Acing the IOC Game: Toward Automatic Discovery and Analysis of Open-Source Cyber Threat Intelligence. In *Proc. CCS'16*.

[16] Yang Liu, Armin Sarabi, Jing Zhang, Parinaz Naghizadeh, Manish Karir, Michael Bailey, and Mingyan Liu. 2015. Cloudy with a Chance of Breach: Forecasting Cyber Security Incidents. In *Proc. USENIX Sec'15*.

[17] Yang Liu, Jing Zhang, Armin Sarabi, Mingyan Liu, Manish Karir, and Michael Bailey. 2015. Predicting Cyber Security Incidents Using Feature-Based Characterization of Network-Level Malicious Activities. In *Proc. IWSPA'15*.

[18] Qiaozhu Mei and ChengXiang Zhai. 2005. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proc. KDD'05*.

[19] A. Modi, Z. Sun, A. Panwar, T. Khairnar, Z. Zhao, A. DoupÃĺ, G. J. Ahn, and P. Black. 2016. Towards Automated Threat Intelligence Fusion. In *Proc. IEEE CIC'16*.

[20] Sathappan Muthiah, Bert Huang, Jaime Arredondo, David Mares, Lise Getoor, Graham Katz, and Naren Ramakrishnan. 2015. Planned Protest Modeling in News and Social Media. In *Proc. AAAI'15*.

[21] Caleb C. Noble and Diane J. Cook. 2003. Graph-based Anomaly Detection. In *Proc. KDD'03*.

[22] Michael OvelgÃĺonne, Tudor Dumitras, B. Aditya Prakash, V. S. Subrahmanian, and Benjamin Wang. 2016. Understanding the Relationship between Human Behavior and Susceptibility to Cyber-Attacks: A Data-Driven Approach. In *Proc. TIST'16*.

[23] J. Piskorski, H. Tanev, and A. Balahur. 2013. Exploiting Twitter for Border Security-Related Intelligence Gathering. In *Proc. EISIC'13*.

[24] Naren Ramakrishnan et al. 2014. 'Beating the News' with EMBERS: Forecasting Civil Unrest Using Open Source Indicators. In *Proc. KDD'14*.

[25] Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proc. LREC Workshop of NLP Frameworks*.

[26] Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. 2012. Open Domain Event Extraction from Twitter. In *Proc. KDD'12*.

[27] Alan Ritter, Evan Wright, William Casey, and Tom Mitchell. 2015. Weakly Supervised Extraction of Computer Security Events from Twitter. In *Proc. WWW'15*.

[28] Carl Sabottke, Octavian Suciu, and Tudor Dumitras. 2015. Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting Real-World Exploits. In *Proc. USENIX Sec'15*.

[29] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proc. WWW'10*.

[30] Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval. (1986).

[31] 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.

[32] Kyle Soska and Nicolas Christin. 2014. Automatically Detecting Vulnerable Websites Before They Turn Malicious. In *Proc. USENIX Sec'14*.

[33] Hristo Tanev, Maud Ehrmann, Jakub Piskorski, and Vanni Zavarella. 2012. Enhancing Event Descriptions through Twitter Mining. In *Proc. ICWSM'14*.

[34] Flora S. Tsai and Kap Luk Chan. 2007. Detecting Cyber Security Threats in Weblogs Using Probabilistic Models. In *Proc. PAISI'07*.

[35] Xiaofeng Wang, Matthew S. Gerber, and Donald E. Brown. 2012. Automatic Crime Prediction Using Events Extracted from Twitter Posts. In *Proc. SBP'12*.

[36] David J. Weller-Fahy. 2017. Towards Finding Malicious Cyber Discussions in Social Media. In *Proc. AICS'17*.

[37] Liang Zhao, Feng Chen, Jing Dai, Ting Hua, Chang-Tien Lu, and Naren Ramakrishnan. 2014. Unsupervised spatial event detection in targeted domains with applications to civil unrest modeling. *PloS one* 9, 10 (2014), e110206.

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

[39] Ziyun Zhu and Tudor Dumitras. 2016. FeatureSmith: Automatically Engineering Features for Malware Detection by Mining the Security Literature. In *Proc. CCS'16*.