2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

CyberTwitter: Using Twitter to generate alerts for Cybersecurity Threats and Vulnerabilities

Sudip Mittal*, Prajit Kumar Das*, Varish Mulwad†, Anupam Joshi*, Tim Finin*
University of Maryland, Baltimore County, Baltimore, MD, USA
{smittal1, prajit1, joshi, finin}@umbc.edu
†GE Global Research
varish.mulwad@ge.com

***Abstract*—In order to secure vital personal and organizational system we require timely intelligence on cybersecurity threats and vulnerabilities. Intelligence about these threats is generally available in both *overt* and *covert* sources like the National Vulnerability Database, CERT alerts, blog posts, social media, and dark web resources. Intelligence updates about cybersecurity can be viewed as temporal events that a security analyst must keep up with so as to secure a computer system. We describe *CyberTwitter*, a system to discover and analyze cybersecurity intelligence on Twitter and serve as a OSINT (Open–source intelligence) source. We analyze real time information updates, in form of tweets, to extract intelligence about various possible threats. We use the Semantic Web RDF to represent the intelligence gathered and SWRL rules to reason over extracted intelligence to issue alerts for security analysts.**

I. Introduction

In the broad domain of security, analysts and policy makers need knowledge about the state of the world to make timely critical decisions, operational/tactical as well as strategic. This knowledge has to be extracted from a variety of different sources, and then represented in a form that will enable further analysis and decision making. Some of the data underlying this knowledge is in textual sources traditionally associated with Open–source Intelligence (OSINT) [1].

OSINT is intelligence gathered from publicly-available *overt* sources such as newspapers, magazines, social-networking sites, video sharing sites, wikis, blogs, etc. In cybersecurity domain, information available through OSINT can compliment data obtained through traditional security systems and monitoring tools like Intrusion Detection and Prevention Systems (IDPS) [2], [3]. Cybersecurity information sources can be divided into two abstract groups, formal sources such as NIST's National Vulnerability Database (NVD), United States Computer Emergency Readiness Team (US-CERT), etc. and various informal sources such as blogs, developer forums, chat rooms and social media platforms like Twitter[1], Reddit[2] and Stackoverflow, these provide information related to security vulnerabilities, threats and attacks. A lot of information is published on these sources on a daily basis making it nearly impossible for a human analyst to manually comb through, extract relevant information, and then understand various contextual scenarios in which an attack might take place. A manual approach even with a large number of human analysts would neither be efficient nor scalable. Automatically extracting relevant information from OSINT sources thus has received attention from the research community [4]–[6].

The real time nature of information on Twitter has allowed researchers to provide significant insights during 'high impact events'. It has been used to analyze emergencies like earthquakes [7], forest fires [8], terrorist attacks [9], natural hazards [10] and so on. Such applications and analysis of Twitter data have solidified its reputation as an *important OSINT source*. On Twitter, several organizations such as Adobe(@AdobeSecurity), Github (@githubstatus), WhatsApp (@wa_status) report on security incidents related to their products. Individual users, often ethical hackers, also report about newly discovered vulnerabilities via Twitter (Figure 1). Such updates can form viable intelligence inputs for human analysts to protect their systems. Detection and updates to various threats and vulnerabilities can be considered as cybersecurity events that impact computer systems. Hence, we believe Twitter can be an effective source to gather information about cybersecurity threats.

In this paper, we present *CyberTwitter*, a framework to analyze tweets about cybersecurity and to issue timely threat alerts to security analysts based on an organization's 'system profile'. Alerts generated by CyberTwitter can then serve as an input to various other security systems who can use them depending upon local organizational security policies. One such system is [2].

In our system, we begin by collecting Twitter data. In the collected tweets we identify, tag and extract various real world conceptual entities related to cybersecurity vulnerabilities such as means of an attack, consequences of an attack, affected software, hardware, vendors, etc. using a Security Vulnerability Concept Extractor (SVCE) [11]. Concepts and entities extracted by SVCE are then linked to existing concepts and entities present in external, publicly available semantic knowledge bases, to further enrich our extracted data. In our system, this information is represented as a set of RDF triples in a semantic knowledge base. We allow analysts to describe a system profile which captures information about installed software and / or hardware. We develop an *intelligence* ontology and use it along with SWRL rules to address time sensitive nature of cybersecurity events. CyberTwitter performs

[1]https://twitter.com/hashtag/cybersecurity?lang=en
[2]https://www.reddit.com/r/cybersecurity/

IEEE/ACM ASONAM 2016, August 18-21, 2016, San Francisco, CA, USA
978-1-5090-2846-7/16/$31.00 ©2016 IEEE

reasoning using this system profile, data in the knowledge base and varying time slices to generate the most relevant and important alerts for human review. Given, the sometimes, unreliable nature of information on Twitter [12] along with the possibility of different local security and organizational policies, we believe that it's best for a human analyst to be *'in loop'* with the system.

Fig. 1: Both users and organizations use Twitter to report potential threats.

Fig. 2: Sample tweet from an individual user about a recent security vulnerability

The nature of intelligence in any security system is that it has a temporal dimension. A piece of information can be considered important at a given time and useless at some other. In our system we define "Intelligence" as an actionable information for a human analyst which makes them aware about a new threat or vulnerability in a software / hardware that they are interested in. In our system we analyze information about these temporal cybersecurity events as they appear on Twitter. For example, in 2015, a 72 hour long DDoS attack on Github was live tweeted by their status account and it served as the go to source of information for the general public [13]. A combination of real time tweets from affected users, ethical hackers, as well organizational accounts can allow both analysts as well as systems to infer a pattern or a larger ongoing attack and generate alerts to provide rapid response. Relevancy of these alerts will also depend on the timestamps of tweets given the highly temporal nature of cyberattacks. Information relevant in one time window is not necessarily relevant in another. Human analysts and policy makers will not only require relevant alerts from their system perspective, but also relevant to a particular time window. For example, any alerts related to the 2015 Github DDOS attack would have been valid in that particular time frame. Thus, any such system would have to make this temporal nature an important aspect of its design.

Rest of the paper is organized as follows– We discuss related work in Section II, our CyberTwitter framework in Section III. We execute and evaluate our system in Section IV & V respectively. We present our conclusions and future work in Section VI.

II. Related Work

A. Twitter as an OSINT source

Over the past decade, Twitter has become a vital source for open source intelligence. The social media site's data has been used by researchers to gather intelligence about the impact of natural disasters [7] [8], terrorists attacks [9], government elections [14], predicting stock markets [15], etc. In our work, we are interested in using Twitter as a source of information to study various cybersecurity events. Twitter users as in when new vulnerabilities are made public, tweet about these vulnerabilities (Figure 1 and 2) to spread information on the network so that others can use that particular information to secure their systems. Individuals or reputed security experts like Brian Krebs (an investigative journalist who writes about cybercrime) can be valuable resources for cybersecurity incidents. Established companies like @web_security or @intersecww or disseminate news, tips and latest information on web security, web application protection, hacker incidents, data breaches, penetration testing results, etc. Other organization specific accounts like @githubstatus, @FirebaseStatus, @herokustatus, @stripestatus, @DOStatus (DigitalOcean), @redditstatus, @twitchstatus, @AdobeSecurity, @JuniperSIRT etc. report on security incidents with respect to their platforms and products. For obvious reasons such organizational accounts mentioned above are valuable sources of information with respect to cybersecurity events. We wish to use these Twitter updates to mine intelligence about various cybersecurity threats and vulnerabilities, Section III gives details about our system.

B. Text Analysis for Cybersecurity

The use of semantic knowledge bases (KB) in cybersecurity has gained traction in the past few years. Considerable atten-

tion has been dedicated to develop techniques for extracting concepts related to security vulnerabilities, affected software, hardware, and organizations and generating its semantic representation [16] [17] [18] [19] [20] [21]. While previous research focused on sources such as NVD and security blogs, our work is applied to Twitter where the content is different from other sources.

C. Knowledge Base systems for Cybersecurity

Research has also focused on integrating data from traditional monitoring and security tools with such KBs and reasoning over it for early threat detection and prevention [3] [22]. However, previous work has relied on cybersecurity KBs generated from NVDs and blogs that are often updated post facto i.e. after the threat or the vulnerability has been known for some time and has been vetted by various security professionals and analysts, whereas *CyberTwitter* generates personalized alerts using a KB that is updated in real time based on a 'user's system profile'.

III. CyberTwitter Framework

We develop CyberTwitter, a framework to automatically issue cybersecurity vulnerability alerts to users (Figure 3). CyberTwitter begins by collecting relevant tweets by querying the Twitter API. The tweet Collection module collects, cleans and stores tweets returned by the API. Every tweet is further processed by the Security Vulnerability Concept Extractor (SVCE) [11] which extracts various terms and concepts related to security vulnerabilities. Intelligence from these terms and concepts is then converted to RDF statements using our *intelligence* ontology. We use UCO ontology (Unified Cybersecurity Ontology) [20] to provide our system with cybersecurity domain information. RDF Linked Data representation is stored in our "Cybersecurity Knowledge Base" allowing our alert system to reason over the data. Finally we issue alerts to the end user based on a "User System Profile". We will further explain various details and sub-modules present in our system in the next few subsections.

Our system can be divided into two major parts. The first is a dynamically populated "Cybersecurity Knowledge Base" that contains information about cybersecurity threats and vulnerabilities. The second is an alert system that issues alerts to the end user based on their "User System Profile" using the "Cybersecurity Knowledge Base".

A. User System Profile

We obtain information about the user's system and store it in the "User System Profile" file. The profile contains information about the operating system, various installed softwares and their version information. We use the profile information as part of our rules. The system information is converted into SWRL rules [23] (see Section III-F), that allows us to reason over them and generate cybersecurity alerts. A sample profile "User System Profile" is shown in Table I.

Software	Type	Version
Ubuntu	Operating System	14.04
Adobe Flash	Software	11.2.202.616
Java	Software	7.0
Chromium Browser / Google Chrome	Browser	49.0.2623.112
Firefox	Browser	45.0.2
Adobe Flash Player (Chromium)	Extention	21.0.0.216-r1

TABLE I: User System Profile.

B. Tweet Collection

CyberTwitter collects data through the Twitter Stream API[3] based on a set of keywords. These keywords are derived from the "User System Profile" and a list of cybersecurity terms (see Figure 4). For our system we limit ourselves to tweets in English language[4]. After collecting a good number of tweets we clean the data using WordNet, which is a large lexical database for English [24].

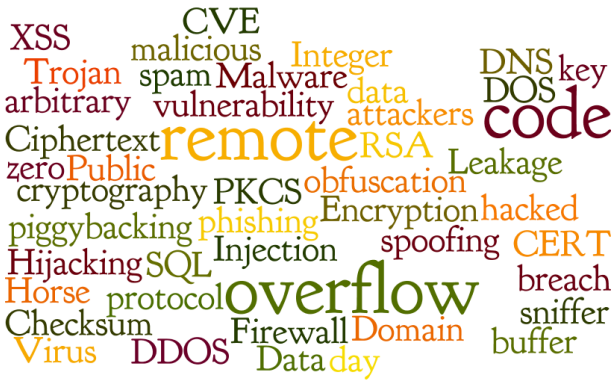

Fig. 4: Data collection keywords.

C. Security Vulnerability Concept Extractor

The Security Vulnerability Concept Extractor (SVCE) consists of a custom Named Entity Recognizer (NER) [11] which extracts terms related to security vulnerabilities. The NER was trained using text from security blogs, Common Vulnerabilities and Exposures (CVE) descriptions and official security bulletins from Microsoft and Adobe. It tags every sentence with the following concepts: Means of an attack, Consequence of an attack, affected software, hardware and operating system, version numbers, network related terms, file names and other technical terms.

The use of the custom NER provides us multiple advantages. SVCE discards all tweets for which the NER fails to identify even a single concept, thus further cleaning up the data. The extracted concepts are also used to generate an RDF Linked Data representation for every tweet that maybe queried by security systems to protect against potential attacks.

D. Filtering and Cleaning Data

In our "Cybersecurity Knowledge Base" we wanted to store highly relevant tweets only. We filter tweets out based on

[3]https://dev.twitter.com/docs/streaming-api
[4]https://dev.twitter.com/streaming/overview/request-parameters#language

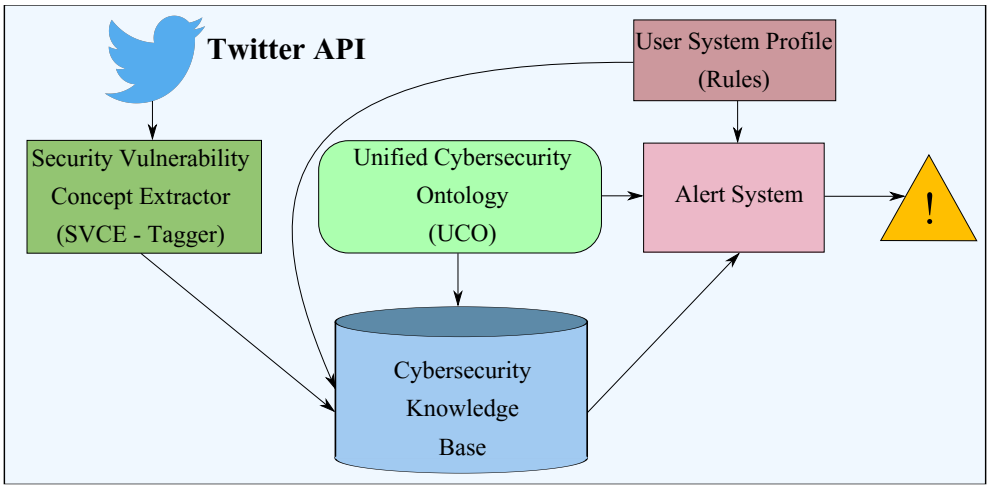

Fig. 3: CyberTwitter: A framework for monitoring and analyzing tweets related to cyber attacks.

```
Example tweet:

ASUS wireless router updates are vulnerable
to a MITM attack http://www.intelligent
exploit.com/view-details.html?id=20071

SVCE Output:

[[  (u'ASUS', u'PRODUCT,'),
      (u'wireless', u'OTHER,'),
      (u'router', u'OTHER,'),
      (u'updates', u'O'),
      (u'are', u'O'),
      (u'vulnerable', u'O'),
      (u'to', u'O'),
      (u'a', u'O'),
      (u'MITM', u'ATTACK,'),
      (u'attack', u'ATTACK,'),
      (u'http:www.intelligent
     exploit.comview-details.html?id=20071',
      u'O')]]
```

Fig. 5: Labelled output generated by the Security Vulnerability Concept Extractor (SVCE).

the output of our Security Vulnerability Concept Extractor (SVCE). In our system we only keep those tweets which contain two or more tags as generated by our SVCE. Such a threshold helps us realize the goal of including only highly relevant tweets in our knowledge base. We discuss the impact of this threshold in Section V.

E. Cybersecurity Ontologies and Knowledge Bases

A data feed sent through the Twitter Stream API essentially consists of a stream of strings that computers can process. However, in the real world, strings represent terms and concepts that may sometimes be ambiguous and computers are not programmed to handle ambiguity. Computer systems can be aided in this task by various Semantic Web technologies that represent real world as concepts. These concepts are then associated with Uniform Resource Identifiers (URIs) [25]. For example, the string "Apple" can be associated with the company Apple Inc. or the fruit apple. Also, these concepts can have various attributes and relations to other concepts. An entity 'Apple' can have an attribute 'type' with a value 'organization' or 'plant'. These attributes are vital so as to differentiate between two completely different concepts having same spellings.

For an intelligent system like CyberTwitter, it is vital to understand the difference between various real world concepts and also to posses a comprehensive knowledge about the cybersecurity domain. In this paper we use various publicly available cybersecurity ontologies and knowledge bases to support information integration and cyber-situational awareness:

1) UCO: Unified Cybersecurity Ontology [20]: The ontology integrates heterogeneous data and knowledge schemas from different cybersecurity systems and standards.
2) DBpedia [26]: DBpedia is a project to extract structured content from the information created as part of the Wikipedia project[5].
3) YAGO (Yet Another Great Ontology) [27]: It is a knowledge base automatically extracted from Wikipedia and other sources.

We have used UCO to provide our system with knowledge about the cybersecurity domain. We use DBpedia and YAGO to link the output generated by our Security Vulnerability Concept Extractor (SVCE) to real world concepts. Entity matching process is performed by using DBpedia[6] [26] and YAGO[7] APIs with the MaxHits parameter set as 1. For example we can use DBpedia to map the string "Adobe Flash" to *dbr:Adobe_Flash* [8]. Both these external knowledge bases help us map string entities to real world conceptual instances. The output from

[5]https://wikimediafoundation.org/wiki/Our_projects
[6]https://github.com/dbpedia-spotlight/dbpedia-spotlight
[7]https://github.com/yago-naga/aida
[8]http://dbpedia.org/page/Adobe_Flash

the SVCE module enlists various cybersecurity related entities in textual tweets like, Means of an attack, Consequence of an attack, affected software, etc. We use UCO, DBpedia and YAGO to link these entities to real world concepts. After entity linking we store the linked data as RDF triples [28] in our "Cybersecurity Knowledge Base".

In our CyberTwitter system we need information of cybersecurity events. Events are temporal in nature. UCO though gives us a domain overview of cybersecurity it cannot handle temporal nature of events. So as to handle *time* in events we create an *Intelligence* ontology.

In our system we define 'Intelligence' as an actionable information for the human analyst which makes them aware about a new threat or vulnerability in a software / hardware that they list in their user system profile. The nature of intelligence in any security system is that it has a temporal dimension. A piece of information can be considered as vital information at a given time and useless at some other instance of time. So to incorporate time we included the following properties in the ontology:

1) *hasCounter(int:Intelligence, X)*: The number of tweets collected (X) with the given intelligence. This data property helps us attach a counter to the intelligence so as to map and group tweets with the intelligence they provide.
2) *hasBeginTime(int:Intelligence,, Y)*: This data property helps us mark the time when we got the first tweet (Y) that gives the system various details about a new vulnerability intelligence.
3) *hasLastIntelTime(int:Intelligence, Z)*: This data property helps us include the time stamp of the last tweet received (Z) with a particular intelligence.
4) *hasVulnerability(int:Intelligence, uco:Vulnerability)*: This object property holds an instance of the extracted vulnerability.
5) *productInUSP(int:Intelligence, L)*: This data property holds a boolean variable L which is set to 'True' if the vulnerability exists in one of the products listed in the 'user system profile'.
6) *isCurrentlyValid(int:Intelligence, M)*: This runtime inferred data property holds a boolean value M which is set to 'True' if the intelligence entity is 'valid and current'. A valid and current intelligence is a one that gives details about an open, temporally significant vulnerability or threat in an affected software / hardware. This property is updated by various SWRL rules listed in Section III-F.

To give an example Figure 6 shows a graphical representation of an intelligence, 'Int1242611341'. The particular intelligence instance is about a vulnerability 'Vul1426796181' that has a consequence of a 'man in the middle attack' that affects 'Asus_wireless_router'. The intelligence is supported by 251 number of tweets and the first tweet with this intelligence was received by the system at time 1457685000 and the latest tweet was received at time 1457669700. If the product is listed in the user system profile the boolean productInUSP data property is set to True.

Creating a comprehensive 'Cybersecurity Knowledge Base' is vital for our system as it provides us with a set of rules and information in form of triples on which we can reason so as to issue vulnerability and threat alerts to the user. The end user can also be given access to the Knowledge Base which they can query using a SPARQL interface [29] which is quite similar to SQL.

F. Cybersecurity Alerts

In the final module of CyberTwitter we generate and issue alerts using the cybersecurity knowledge base and the user system profile. After creating the knowledge base we need an intelligent system to reason over various RDF statements and evaluate if the system should raise an alert to inform the user about a potential threat or vulnerability that may exist.

After creating the cybersecurity knowledge base we include various SWRL rules [23] to our system. SWRL rules contain two parts, *antecedent* part (body), and a *consequent* (head). The body and head consist of conjunctions of a set of '*atoms*' [23]. Informally, a rule may be read as meaning that if the antecedent holds (is "true"), then the consequent must also hold.

We have logically divided this module in 2 different parts. In the first part we compute if an intelligence is 'valid and current' and in the second part we use a valid intelligence to raise an alert. When a tweet with actionable intelligence that already exists in the knowledge base arrives in the system the intelligence entity corresponding to that vulnerability gets updated (For a tweet with new intelligence a new entity is created). When the alert system is triggered value of an 'inferred property' *isCurrentlyValid(int:Intelligence, M)* is computed through SWRL rules.

The first rule is used to compute the inferred property *isCurrentlyValid(int:Intelligence, M)* which depend on the value of last tweet time, if the product is in the user system profile and how 'old' is the intelligence. The variable T is a system parameter provided by the user so as to specify a time window. This time window determines if an intelligence entity is 'new' enough to issue an alert. For example, if the user sets T as 24 hours, then an intelligence entity which was last updated in the last 24 hour time period will be considered valid and current by the system.

```
hasLastIntelTime() ^
productInUSP() ^
withinRange( , CurrentSysTime - T
    ,CurrentSysTime)
=>
isCurrentlyValid()
```

The SWRL rules used to raise alerts use the inferred property *isCurrentlyValid(int:Intelligence, M)*, number of tweets associated with that intelligence entity. N is a system parameter specified by the user. This parameter can be used by the user to tweak the system so as to give alerts only if the number of tweets associated with an intelligence is substantial or if the system must inform the user about intelligence which are supported by a few tweets. For example, if the value of N set by the user is 10, then all intelligence entities with at-least 10 tweets supporting it are used to generate an alert.

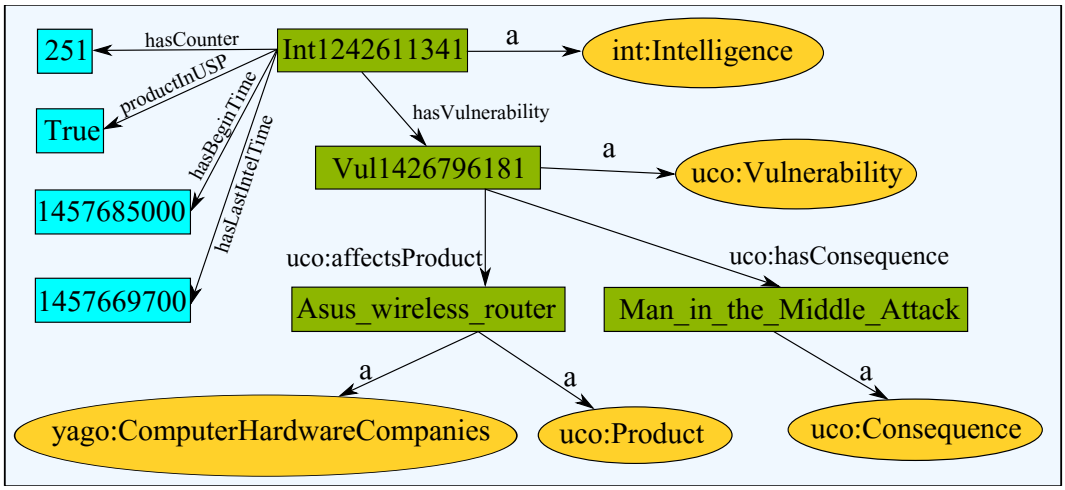

Fig. 6: Graphical representation of RDF for example tweet shown in Figure 5.

```
Rule using consequences:
  isCurrentlyValid() ^
  hasConsequence() ^
  hasCounter() ^
  swrlb:greaterThanOrEqual(N, )
  =>
  RaiseAlert()

Rule using means:
  isCurrentlyValid() ^
  hasMeans() ^
  hasCounter() ^
  swrlb:greaterThanOrEqual(N, )
  =>
  RaiseAlert()
```

Using the above two rules we determine if various RDF statements have actionable intelligence that may be of interest to the user. and we issue alerts. In our system we have purposefully separated the two rules to generate the value for hasIntelligence. One for hasConsequence and another one for hasMeans. This can create multiple repeated alerts in our system. We can combine the two rules to produce a more concrete rule where both consequences and means are present in the RDF statement. However to ensure a better throughput and performance we use two different rules in our system.

After generating the alerts we display them to the human analyst along with a link to the list tweets and SVCE tags through which the particular alert was generated. These alerts can then be used by human analysts and policy makers to make vital decisions to secure their organizational / personal systems.

IV. Experimental Setup

CyberTwitter can help security analysts make important decisions by keeping them updated about various vulnerabilities and threats. After creating the system we executed it under experimental conditions for a period of ten days (March 10th, 2016 to March 20th, 2016) using the 'user system profile' listed in Table I. We set the 2 system parameters T and N as 24 hours and 50 respectively for this execution.

In this ten-day period the system collected 143,701 tweets. This dataset consists of only those tweets which are in English as determined by the Twitter API. These tweets were then passed though the Security Vulnerability Concept Extractor (SVCE) so as to tags every sentence with the following concepts – means of attack, consequence of attack, affected software, hardware and operating system, version numbers, network related terms, file names and other technical terms.

After tagging all tweets in our dataset we filter out tweets that have fewer than two tags. After filtering out various tweets we are left with a dataset of 10,004 relevant tweets. These tweets were then used to create various intelligence entities which had information about threats and vulnerabilities relevant to various software. We got 158 such intelligence entities. These 158 entities also had intelligence about other products which are not part of the 'user system profile' listed in Table I. Using these 158 intelligence entities our alert system issued 15 alerts for software listed on the 'user system profile' in Table I.

V. Evaluation

We performed an initial evaluation of our prototype system using the tweets collected over the ten-day time frame described above. We evaluate three aspects: the quality of the tags generated by the SVCE module, the quality of the alerts generated, and how often our system missed intelligence because it discarded relevant tweets. We did not evaluate our entity matching process as it was done through DBpedia and YAGO APIs with the MaxHits parameter set to 1. Human assessments and annotation was done by doctoral students familiar with the domain of cybersecurity.

For our first evaluation measure we check the quality of tags generated by our SVCE module. We tagged 250 randomly selected tweets and then manually checked the tags. The annotation task involved various annotators manually checking the SVCE output and selecting one of the three options. The

options were if the SVCE output was correct, partially correct or wrong. Our annotators agreed on the fact that 143 tweets were marked correctly by the SVCE module and out of the remaining 107 tweets, 83 were tagged completely wrong and the remaining were tagged partially correct.

To evaluate the quality of the alerts generated by the system we did a two-part evaluation in which we first study the 158 intelligence entities generated in the above mentioned execution and then if we are overlooking vital intelligence in the form of discarded tweets. Each part is described below.

In evaluating the quality of generated alerts, we found that out of the 158 entities, 121 contained intelligence on threats and vulnerability found in software/hardware that was not part of the 'user system profile' listed in Table I. Out of the 37 related to the provided 'user system profile', 15 alerts were issued. To evaluate the quality of these alerts we conducted a small user study where we asked five assessors to judge the usefulness of alerts (options: *useful*, *maybe*, *useless*) given the set of tweets responsible for the alert. Out of 15 alerts generated 13 were marked as useful and the remaining two were marked as *maybe*.

We evaluated the loss of intelligence because of discarded tweets, i.e., those not included in the dataset of 10,004 tweets. A random sample of 300 tweets was generated from the discarded tweets. In these, our annotators found 44 tweets with actionable intelligence out of which 16 were related to the provided 'user system profile'. We believe that these tweets were wrongfully tagged by our SVCE module because of spelling mistakes, unidentifiable characters, informal slang expressions, non-English words, etc. However the intelligence provided by these discarded tweets was already extracted from other tweets in this execution period.

VI. Discussion and Future Work

In this paper we describe our CyberTwitter framework which gives the end user cybersecurity intelligence alerts using publicly available data from Twitter. We employ a Security Vulnerability Concept Extractor (SVCE) to extract terms related to security vulnerabilities. We store the intelligence we extract as RDF [28] triples in a cybersecurity knowledge base and use SWRL rules to create alerts for the security analysts based on a 'user system profile' which enlists various system details like operating system, software installed, version numbers, etc. We create an 'intelligence' ontology to analyze temporal cybersecurity events and also to ensure that the generated alerts are current and relevant. These alerts can then be used by the user to keep the organization's system updated and secure.

In the future we will like to incorporate user feedback on various alerts issued. We can also add a module which will incorporate user feedback and improve the quality of cybersecurity alerts that are generated by the system. We would also like to incorporate other blogs, news websites and social networks like Reddit, Hacker News etc. as they are vital platforms for different users to discuss and debate cybersecurity vulnerabilities and threats. Adding a diverse set of information sources can further improve our system. We also foresee a challenge to our system when we have missing or partial information. Specific research is required so as to handle missing information which can further improve our system. Our system can also be extended to include updates about various 'patch updates' to remove vulnerabilities issued by a product provider.

We expect that we can increase the recall of tweet selection by using a semantic textual similarity system (STS) developed in our lab [30] which includes a module that is optimized to work on tweets [31]. This will allow us to recognize relevant tweets based on how similar in meaning their content is to a seed set of concepts, words and phrases. We will extend STS's current word embedding model, which was trained on general text, by augmenting it with a model trained on cybersecurity text following the technique described in [32].

Acknowledgment

Support for this work was provided by NSF grants 0910838 and 1228198, a Supplement from United States Department of Defense to NSF award 1439663, and funds from the Oros Family Professorship.

References

[1] R. D. Steele, "Open source intelligence: What is it? why is it important to the military," *American Intelligence Journal*, vol. 17, no. 1, pp. 35–41, 1996.

[2] M. L. Mathews, P. Halvorsen, A. Joshi, and T. Finin, "A collaborative approach to situational awareness for cybersecurity," in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*, Oct 2012, pp. 216–222.

[3] S. More, M. Matthews, A. Joshi, and T. Finin, "A knowledge-based approach to intrusion detection modeling," in *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*. IEEE, 2012, pp. 75–81.

[4] F. Neri and P. Geraci, "Mining textual data to boost information access in osint," in *Information Visualisation, 2009 13th International Conference*. IEEE, 2009, pp. 427–432.

[5] L. C. Pouchard, J. D. Dobson, and J. P. Trien, "A framework for the systematic collection of open source intelligence," in *AAAISS*, 2009.

[6] P. Maciolek and G. Dobrowolski, "Cluo: Web-scale text mining system for open source intelligence purposes," *Computer Science (AGH)*, vol. 14, pp. 45–62, 2013.

[7] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 851–860.

[8] B. De Longueville, R. S. Smith, and G. Luraschi, "Omg, from here, i can see the flames!: a use case of mining location based social networks to acquire spatio-temporal data on forest fires," in *Proceedings of the 2009 international workshop on location based social networks*. ACM, 2009, pp. 73–80.

[9] O. Oh, M. Agrawal, and H. R. Rao, "Information control and terrorism: Tracking the mumbai terrorist attack through twitter," *Information Systems Frontiers*, vol. 13, no. 1, pp. 33–43, 2011.

[10] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen, "Microblogging during two natural hazards events: what twitter may contribute to situational awareness," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2010, pp. 1079–1088.

[11] R. Lal, "Information extraction of security related entities and concepts from unstructured text." *Master's Thesis, University of Maryland Baltimore County*, 2013.

[12] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi, "Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy," in *Proceedings of the 22nd international conference on World Wide Web companion*. International World Wide Web Conferences Steering Committee, 2013, pp. 729–736.

[13] M. Nestor, "Github has been under a continuous ddos attack in the last 72 hours," May 2015. [Online]. Available: http://goo.gl/KBucR0

[14] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting elections with twitter: What 140 characters reveal about political sentiment," in *ICWSM*, 2010.

[15] J. Bollen, H. Mao, and X.-J. Zeng, "Twitter mood predicts the stock market," *CoRR*, vol. abs/1010.3003, 2011.

[16] V. Mulwad, W. Li, A. Joshi, T. Finin, and K. Viswanathan, "Extracting information about security vulnerabilities from web text," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, vol. 3. IEEE, 2011, pp. 257–260.

[17] A. Joshi, R. Lal, T. Finin, and A. Joshi, "Extracting cybersecurity related linked data from text," in *Semantic Computing (ICSC), 2013 IEEE Seventh International Conference on*, Sept 2013, pp. 252–259.

[18] N. McNeil, R. A. Bridges, M. D. Iannacone, B. Czejdo, N. Perez, and J. R. Goodall, "Pace: Pattern accurate computationally efficient bootstrapping for timely discovery of cyber-security concepts," in *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, vol. 2. IEEE, 2013, pp. 60–65.

[19] C. L. Jones, R. A. Bridges, K. M. Huffer, and J. R. Goodall, "Towards a relation extraction framework for cyber-security concepts," in *Proceedings of the 10th Annual Cyber and Information Security Research Conference*. ACM, 2015, p. 11.

[20] Z. Syed, A. Padia, M. L. Mathews, T. Finin, and A. Joshi, "UCO: A unified cybersecurity ontology," in *Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security*. AAAI Press, 2015, pp. 14–21.

[21] M. Iannacone, S. Bohn, G. Nakamura, J. Gerth, K. Huffer, R. Bridges, E. Ferragut, and J. Goodall, "Developing an ontology for cyber security knowledge graphs," in *Proceedings of the 10th Annual Cyber and Information Security Research Conference*. ACM, 2015, p. 12.

[22] M. L. Mathews, A. Joshi, and T. Finin, "Detecting botnets using a collaborative situational-aware idps," in *Second International Conference on Information Systems Security and Privacy*, February 2016.

[23] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "Swrl: A semantic web rule language combining owl and ruleml," May 2004. [Online]. Available: https://www.w3.org/Submission/SWRL/

[24] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: An on-line lexical database*," *International journal of lexicography*, vol. 3, no. 4, pp. 235–244, 1990.

[25] T. Berners-Lee, T. Bray, D. Connolly, P. Cotton, R. Fielding, M. Jeckle, C. Lilley, N. Mendelsohn, D. Orchard, N. Walsh, and S. Williams, "Uniform resource identifier," December 2004. [Online]. Available: https://www.w3.org/TR/webarch/#identification

[26] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, *DBpedia: A Nucleus for a Web of Open Data*. Springer, 2007.

[27] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 697–706.

[28] "Resource description framework (rdf)." [Online]. Available: http://www.w3.org/RDF/

[29] "Sparql protocol and rdf query language 1.1 overview." [Online]. Available: http://www.w3.org/TR/sparql11-overview/

[30] A. Kashyap, L. Han, R. Yus, J. Sleeman, T. Satyapanich, S. Gandhi, and T. Finin, "Robust semantic text similarity using lsa, machine learning, and linguistic resources," *Language Resources and Evaluation*, vol. 50, pp. 126–161, 2016.

[31] T. W. Satyapanich, H. W. Gao, and T. Finin, "Ebiquity: Paraphrase and Semantic Similarity in Twitter using Skipgram," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, June 2015, pp. 51–55.

[32] M. P. Chavan, "Cybersecurity text corpus and its application for augmenting semantic text similarity," Master's thesis, University of Maryland, Baltimore County, May 2014.