Proceedings of the 8th International Conference on Communication and Electronics Systems (ICCES 2023)
IEEE Xplore Part Number: CFP23AWO-ART; ISBN: 979-8-3503-9663-8

2023 8th International Conference on Communication and Electronics Systems (ICCES) | 979-8-3503-9663-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCES57224.2023.10192794

A Twitter-based Software Vulnerability Alert Framework using Natural Language Processing

Yadhu Krishna M, Sneha C K, Thejaswi A R, Sayooj B Kumar, Greeshma Sarath
Department of Computer Science and Engineering,
Amrita School of Computing,
Amrita Vishwa Vidyapeetham, Amritapuri, India
{yadhukrishna, snehack, thejaswiar, sayoojbkumar}@am.students.amrita.edu
, greeshmasarath@am.amrita.edu

***Abstract*—With the increasing usage of networks, the frequency of cyber attacks are expected to increase. To prepare against these attacks, modern organizations require effective ways to ensure cyber security. Twitter is one of the most well-known social media platforms in the world. Tweets by users on Twitter cover a wide range of topics, from daily life to politics and breaking news. This social media platform has a sizable and diverse user base, provides great accessibility, and timeliness, resulting in the production of a huge amount of publicly available data, which in turn can be used effectively for a variety of purposes. For detecting security vulnerability events via Twitter, we introduce a Natural Language Processing-based approach. The tool can be used to provide alerts and notifications to users when a security vulnerability is detected.**

***Index Terms*—Natural Language Processing, Cyber Security**

I. Introduction

Twitter is one of the most common platforms for the information security community to share and discuss their findings. Organizations run software that is built using a variety of frameworks and libraries. It has security vulnerabilities or bugs that are not known to the official authors, and could be unpatched in official releases. However, these vulnerabilities might be known to the security community through unofficial sources like Twitter. Since this information is public, it will be available to attackers, who can misuse this to attack an organization. Fig. 1 and Fig. 2 shows two different tweets, which were posted publicly, indicating vulnerabilities that were identified in different software applications.

The remainder of the paper is structured as follows. Section II discusses similar work, Section III explores the tools and methods utilized in this study, Section IV provides key definitions referenced throughout the paper, Part V provides further detail about the suggested methods, and finally, in Part VI, the results are discussed along with future work.

II. Related Works

Over the recent years, Twitter has evolved into a key resource for open-source intelligence. With the advancement in web technologies and the rise of software vulnerabilities that are discovered, numerous studies have taken place in this area. In this work, we aim to identify software vulnerabilities from

Fig. 1. A tweet indicating a security vulnerability identified in Microsoft's Support Diagnostic Tool (MSDT).

Fig. 2. A tweet indicating a security vulnerability identified in one of the Grafana plugins.

tweets that are shared by the security community on Twitter. The following section presents a summary of past research in this field.

Mittal, Sudip, et al. proposed CyberTwitter [1], a framework that uses semantic web technologies and SVCE Tagger(NER) to collect and scans public tweets and issues threat alerts to cyber security analysts based on the system profile configured by the organization. In CySecAlert [2], Riebe, Thea, et al. proposed a system which generates real-time alerts for cyber

security events, using an active learning approach. Relevant tweets were identified and clustering was done to form events. An alert was generated in a cluster when the number of tweets in a cluster was greater than a predefined value.

Mulwad, Varish et al. employed an machine learning-based classifier [3] that can map text content to relevant concepts, entities, relations and events. It is based on Wikitology, and a computer security exploits ontology.

Dionísio, Nuno, et al. implemented a tool [4] to detect cyber threats from twitter using deep neural networks. This tool only collects tweets from curated list of Twitter accounts. The approach is based on high-level three-stage pipeline architecture, and a binary classifier which is based on CNN. Alperin, Kenneth, et al. proposed a framework [5] for Unsupervised classification and Data Mining of Tweets about Cyber Vulnerabilities, which is based on Zero-shot classification using a BART model. The BART model predicted relevance of text to security content without the use of supervised training. Liu, Xiaohua, et al. proposed a Named Entity Recognition (NER) system for tweets [6]. It uses KNN classifier and linear CRF model. It uses a semi-supervised learning and recognizes named entities in tweets

The following are some of the limitations that were identified with the existing works:

- Since most of the existing approaches work on the basis of information extracted from a predefined set of accounts, hacked or compromised accounts can lead to poisoning notification.
- Existing approaches do not consider polarity checking i.e. patch and security updates do not trigger alerts.
- They do not have user feedback and quality improvement techniques.
- Very less research has taken place in the application of Keyword extraction techniques to this field.

III. Proposed Solution

To solve this problem, we develop an application that can generate an alert to organization when a security vulnerability or bug is discovered in a software that the organization uses. The application uses Twitter as a data source, and collects relevant tweets with the help of Twitter Developer API. As one of the biggest social media platforms with vast amounts of real time information and security related data in the form of tweets,twitter is an apt and powerful tool for gaining insights into security vulnerabilities or bugs.The organization can set up our application, and configure it to receive alerts when a security vulnerability is detected in an application which is in the interest of the organization. The application also allows the administrators to receive information about updates and patches. The administrator can use this information to validate and fix the vulnerability to protect the organization.

The major contributions made in this paper are

1) The paper presents a novel approach to identify tweets which are related to security vulnerabilities in software applications.
2) The proposed approach can be used to generate notification alerts when a vulnerability is detected in an application that is in the interest of the organization.
3) The proposed approach can be easily extended to receive updates about patches and fixes to the applications of interest.
4) The approach also implements mechanisms to eliminate the number of wrong alerts issued.
5) The study provides a comprehensive literature review and a comparison of existing solutions in the relevant field
6) The proposed approach also eliminates the limitations identified to also provides a ground for further research.

IV. Tools & Techniques

The following section enlists some of the important libraries and softwares that were used in this research work.

1) **Twitter API** - enables advanced programmatic access to Twitter Platform. Twitter API was used to collect tweets from Twitter.
2) **Python Requests** - is an HTTP client for Python that provides functionalities to interact with HTTP Servers. This library is used to implement Twitter API client for collecting tweets via Twitter API.
3) **Yet Another Keyword Extractor!** (YAKE!) [7] - is a unsupervised keyword extraction tool. YAKE can automatically extract important keywords from text. The library was employed to extract important keywords from tweets.
4) **BeautifulSoup4** [8] - is an easy-to-use Python module that can be used to work with markup languages such as XML. The library was used for extracting CVE details from XML files, during identification of vulnerability keywords.
5) **Pandas** [9] - is a famous Python library used for data analysis and statistics.
6) **Atlassian Jira** - is a widely-used software project management tool that provides software development teams with a variety of features to streamline their workflows. In this project, we have utilized Jira to display alert messages for vulnerabilities or bugs found in tweets that are relevant to the organization.

V. Definitions

This section offers precise definitions of essential terms that will be frequently employed in the subsequent sections.

A. System Specification

vulnerability alert framework is interested in only alerting the user on detecting vulnerabilities in applications of the user's interest. Thus, the user or the organization inputs a list of applications in which they are interested in receiving an alert when a vulnerability is detected. Table I indicates a sample system specification input. The application generates alerts only for these applications.

TABLE I
EXAMPLE SYSTEM SPECIFICATION

Sl. No	Product Name
1	Google Chrome
2	Microsoft Windows NFS
3	Zyxel network-attached storage
4	Acrobat Reader
5	SourceCodester Online Employee Leave Management System
6	PAN-OS 10.0
7	Cisco CVR100W Wireless-N Wireless router

TABLE II
TOP 20 VULNERABILITY KEYWORDS AND THEIR OCCURRENCE

Keyword	Occurrence
cross-site scripting	307
remote code execution	222
sql injection vulnerability	214
stored cross-site scripting	179
crafted html page	173
reflected cross-site scripting	164
denial of service	163
potentially exploit heap	158
execute arbitrary code	158
exploit heap corruption	151
code execution vulnerability	151
xss	146
privilege vulnerability	135
allowed a remote	131
addressed with improved	130
arbitrary code execution	126
high privilege users	124
buffer overflow	117
command injection vulnerability	113

B. Vulnerability Keywords

Vulnerability Keywords can be defined as a common list of keywords that appear in security-related text. A list of keywords were collected by combining keywords obtained from:

1) **OWASP Top 10** [10] - is a common awareness document that enlists top security vulnerabilities.
2) **Dataset from MITRE CVE Database** - The MITRE CVE [11] database contain vulnerability information published by various organizations from around the world.

A dataset containing of 7914 entries was taken from cve.mitre.org. YAKE keyword extraction was performed on this dataset. This involves preprocessing the input text using NLP techniqes such as stemming and stopword removal.Futhermore, regular expressions were applied through NLP to replace specific pattern in the string with designated replacement strings.our approch also incorporated YAKE keyword extraction, which involved leveraging NLP techniqes to identify the most frequently occuring keywords. The keywords that occur most frequently are termed as Vulnerability Keywords. Fig. **??** displays the occurrence of vulnerability keywords identified. Table II indicates top 20 vulnerability keywords and their occurrence.

C. Vulnerability Score (V)

Vulnerability score (denoted by V) of a tweet is termed as the number of occurrence of vulnerability keywords in a tweet.

D. Polarity Value (P)

Polarity value (denoted by P) of a tweet is said to be negative if the given tweet is a tweet that contains information of a fix that has been made. It is positive if the tweet is about a vulnerability that has been identified in a software. Table IV shows sample polarity keywords. These are used for calculation of polarity value.

E. System Specification Score (S)

System Specification score (denoted by S) is a value that is assigned on the basis of how much a tweet matches the system specification input by the user. System specification has been detailed in section V-A.

F. Relevancy Score (R)

The relevancy score (denoted by R) of a tweet indicates how relevant each individual tweet is. The relevancy score is calculated as the product of polarity value, vulnerability score and system specification score. This can be mathematically represented as:

$$R = P * V * S \tag{1}$$

G. Final Score

The Final score (denoted by F) of a tweet indicates how relevant a tweet is. Each individual tweet is Grouped on the basis of its product ID and tags. Final score is calculated by adding together the individual system specification, vulnerability, and polarity scores for each product, then multiplying those results. This can be mathematically represented as:

$$F = \Sigma V * \Sigma S * \Sigma P \tag{2}$$

A value of 0 to Final score indicates that the tweet is completely irrelevant, and can be safely discarded. A negative Final score indicates that the tweet is about a patch or fix update. If there is only one tweet pertaining to specified system specification, the relevancy score of that tweet can be regarded as the final score.

Table III shows sample tweets and their calculated scores according to vulnerability keywords, system specification and polarity keywords as detailed in IV.

VI. METHODOLOGY

The methodology proposed by this paper has been pictorially represented in Fig. 4. The working of each internal stage is as stated below.

ProductId	Tags	VulnScore	ProductScore	PolarityScore	FinalScore
102	60 256 257	36.0	12.0	1.0	432.0
103	243 320	2.0	1.0	1.0	2.0
	60 256 257	33.0	11.0	1.0	363.0
105	243 264	2.0	1.0	1.0	2.0
	243 320	2.0	1.0	1.0	2.0
106	60 243 256 257	4.0	1.0	1.0	4.0

Fig. 3. The figure shows the distribution of Sample product IDs and their scores obtained from the dataset. Final scores above predefined relevancy threshold would be only considered for generating alerts.

TABLE III
SAMPLE TWEETS AND SCORE CALCULATION

Tweet	Vulnerability Score	Polarity Value	System Specification Score	Relevancy Score
Atlassian JIRA <8.2.4 pre-authentication RCE can be exploited after finding a vulnerable version with Purplemet.	3	1	2	4
ASUS wireless router updates are vulnerable to a MITM attack.	2	1	0	0
WhatsApp updates patch two critical and high-severity RCE vulnerability related to video calls and video files.	2	-1	1	-2

TABLE IV
SAMPLE KEYWORD LIST FOR SCORE CALCULATION

System Specification	Vulnerability Keywords	Polarity Keywords
Atlassian JIRA	RCE	Update
JIRA	Exploit	Patch
Windows Server	MITM	Fix
WhatsApp	Version	Mitigate
PostgreSQL	Vulnerable	Release

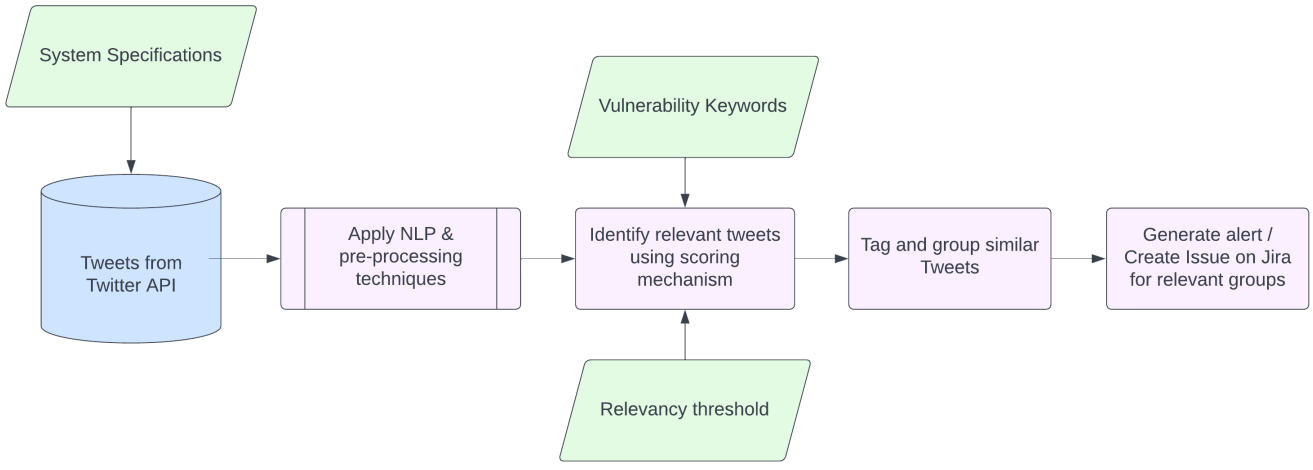

Fig. 4. The figure indicates the proposed architecture of the vulnerability detection framework.

A. Configuration

In the configuration stage, the user inputs system specification, and configures alert frequency and relevancy threshold. Alerts are generated only on the basis of the input system specification.

1) **Alert frequency** [10] - Alert frequency is a customizable threshold set by the organisation to regulate the number of alert they receive.
2) **Relevancy threshold** - Relevancy threshold is the minimum score predetermined by the organisation to determine the relevancy of the tweet

In the project, the Relevancy threshold was set to 5. Furthermore, the Alert frequency value of 15 was utilized to regulate the number of alerts received by the organization.

B. Data Collection

In this stage, the application begins to collect live tweets from Twitter using Twitter API. The module makes request to twitter in a specified interval of time, which is configured by the user in the previous stage. The collected tweets are sent for preprocessing.

1) **Twitter API Configuration** [10] - In this stage the application begins to collect live tweet from Twitter using Twitter API. In order to interact with Twitter REST API, A twitter developer account was established and acquired necessary developer API credentials. These credentials consist of consumer key, consumer secret, access token and access token secret, which authenticate the application when connected with the twitter API.
2) **System configuration** - During this stage, We specify the system specifications and parameters that govern the accuracy of the alerts generated. These system specifications aid in identifying the product that are relevant to the user's interests.We also define values for parameters that control the number of tweets that is collected in an iteration and alert generation sensitivity.

The collected tweets are sent for preprocessing.

C. Preprocessing

In the preprocessing stage, the collected tweets are cleaned up. Special characters, emojis, links, tags and mentions are removed from tweets. Basic Natural Language Processing (NLP) techniques such as stop words removal and stemming are performed on the collected data. YAKE! keyword extraction is applied to identify relevant keywords from the tweet.

D. Identification of Relevant Tweets

Identification of relevant tweets are based on relevancy score. The calculation of relevancy score is as detailed in Section V-F.

E. Tagging

In this stage, the collected tweets are tagged using a vulnerability keyword dataset to group them effectively for further analysis.

F. Grouping

Grouping is a technique that organizes tweets based on similarity. Tweets having same system specification and same vulnerability keywords are grouped together. A final score is also calculated based on this.

G. Alert Generation

For each relevant groupings, an issue would be raised on Atlassian Jira board. This serves as an alert to the organization. The issue generated would contain the product name, vulnerability keyword identified and the relevancy score associated with the alert. The administrator can then look into this issue, and can resolve them.

VII. Evaluation Criteria

The test data for the study was collected by utilizing the Twitter Developer API. We randomly selected 38 software products and used the Twitter data collection module to gather relevant tweets for each product. However, we found that 12 of the products had no relevant tweets available. For the remaining 26 products, we collected a total of 373 tweets, which were carefully labeled by human annotators. After the tweets were collected, we applied a cleaning process to remove any extraneous information and prepare the data for classification. The cleaned tweets were passed through the classifier. The result of this classification was compared with the annotations by human annotators to calculate the accuracy, precision and recall.

VIII. Results

After running our classifier on the collected data, an accuracy score of 0.87 was obtained, with a precision score of 0.60 and a recall score of 0.75. Our approach for achieving high accuracy in our vulnerability Keyword dataset involved a two-fold process. Firstly, we collected a promising dataset from the CVE dataset, which served as the foundation for our keyword identification process. Secondly, we manually inspected each keyword and removed any irrelevant entries, thereby retaining only those keywords relevant to security. This rigorous curation process ensures that our Vulnerability keyword dataset is both accurate and comprehensive. Although these results are promising, further improvements can be made to enhance the quality of alerts by improving the keywords. The results indicate that the system is effective in identifying potential vulnerabilities and providing important information for addressing such threats.

IX. Conclusion & Future Work

In this paper, we have presented a software vulnerability alert framework that utilizes Natural Language Processing to analyze publicly available data from Twitter. Our framework provides a novel approach to proactively secure software and applications by generating alerts based on the analysis of tweets related to vulnerabilities.

The results of our evaluation demonstrate that our framework can effectively identify tweets related to software vulnerabilities and generate alerts with high accuracy, precision, and

recall. Furthermore, our framework can be easily extended to include other sources such as security blogs and other social network platforms. In future, the developed application can be also used to provide alerts for fixes or updates that are released.

REFERENCES

[1] S. Mittal, P. K. Das, V. Mulwad, A. Joshi, and T. Finin, "Cybertwitter: Using twitter to generate alerts for cybersecurity threats and vulnerabilities," in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2016, pp. 860–867.

[2] T. Riebe, T. Wirth, M. Bayer, P. Kühn, M.-A. Kaufhold, V. Knauthe, S. Guthe, and C. Reuter, "Cysecalert: An alert generation system for cyber security events using open source intelligence data," in *International Conference on Information and Communications Security*. Springer, 2021, pp. 429–446.

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

[4] N. Dionísio, F. Alves, P. M. Ferreira, and A. Bessani, "Cyberthreat detection from twitter using deep neural networks," in *2019 international joint conference on neural networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[5] K. Alperin, E. Joback, L. Shing, and G. Elkin, "A framework for unsupervised classificiation and data mining of tweets about cyber vulnerabilities," *arXiv preprint arXiv:2104.11695*, 2021.

[6] X. Liu, S. Zhang, F. Wei, and M. Zhou, "Recognizing named entities in tweets," in *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, 2011, pp. 359–367.

[7] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt, "Yake! keyword extraction from single documents using multiple local features," *Information Sciences*, vol. 509, pp. 257–289, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025519308588

[8] L. Richardson, "Beautiful soup documentation," *Dosegljivo: https://www. crummy. com/software/BeautifulSoup/bs4/doc/.[Dostopano: 7. 7. 2018]*, 2007.

[9] W. McKinney *et al.*, "pandas: a foundational python library for data analysis and statistics," *Python for high performance and scientific computing*, vol. 14, no. 9, pp. 1–9, 2011.

[10] [Online]. Available: https://owasp.org/www-project-top-ten/

[11] [Online]. Available: https://cve.mitre.org/

[12] K. Makice, *Twitter API: Up and running: Learn how to build applications with the Twitter API*. " O'Reilly Media, Inc.", 2009.

[13] R. V. Chandra and B. S. Varanasi, *Python requests essentials*. Packt Publishing Ltd, 2015.

[14] M. Yadhu Krishna, S. Sanjana, and M. Thushara, "Ad service detection-a comparative study using machine learning techniques."

[15] S. B. Kumar, K. Rajeev, S. Dileep, A. M. Ashraf, and T. Anjali, "Organization security framework—a defensive mechanism," in *Proceedings of Third International Conference on Sustainable Expert Systems: ICSES 2022*. Springer, 2023, pp. 105–113.

[16] B. Venugopal and G. Sarath, "A novel approach for preserving numerical ordering in encrypted data," in *2016 International Conference on Information Technology (ICIT)*. IEEE, 2016, pp. 61–68.

[17] S. Sanjay, M. A. Kumar, and K. Soman, "Amrita_cen-nlp@ fire 2015: Crf based named entity extractor for twitter microposts." in *FIRE Workshops*, 2015, pp. 96–99.

[18] A. Aswathy, R. Prabha, L. S. Gopal, D. Pullarkatt, and M. V. Ramesh, "An efficient twitter data collection and analytics framework for effective disaster management," in *2022 IEEE Delhi Section Conference (DELCON)*. IEEE, 2022, pp. 1–6.

[19] K. Jayaram and K. Sangeeta, "A review: Information extraction techniques from research papers," in *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. IEEE, 2017, pp. 56–59.