An exploration of future challenges for crowd sourced vulnerability detection

Olivier de Casanove, Florence Sèdes

An exploration of future challenges for crowd sourced vulnerability detection

Olivier de Casanove
Institut de Recherche en Informatique de Toulouse - IRIT
Université Toulouse III - Paul Sabatier
Toulouse, France
olivier.decasanove@irit.fr

Florence Sèdes
Institut de Recherche en Informatique de Toulouse - IRIT
Université Toulouse III - Paul Sabatier
Toulouse, France
florence.sedes@irit.fr

***Abstract*—Human dimension of cybersecurity falls under different practices or strategies. Crowd sourcing the detection of vulnerabilities is one of them. A lot of intelligence is published on online social networks (OSN) and it may be hard to process everything for human analysts. There is a need for automated solutions which will process the OSN stream to extract knowledge. We will specifically talk about the methods used to mine information related to vulnerabilities in this paper and compare them to identify future challenges.**

***Index Terms*—Cybersecurity, Vulnerability, Online Social Network, OSINT, Crowdsourcing**

I. Introduction

The use of online social networks (OSN) to detect cyberattacks and cybersecurity-related events is known as "crowd sourced" detection. The users, by exchanging messages between each other, give a vision of the current threat landscape. The user-generated data are publicly available, therefore using OSN to detect cybersecurity-related events is a subset of Open-source intelligence (OSINT). OSINT is the collection and analysis of data gathered from open sources to produce actionable intelligence. In the context of crowdsourced detection, the actionable intelligence is used for three aspects:

1) Extraction of information for decision-making and treatment of security alerts. This is especially useful for analysts operating in Computer Security Incident Response Team (CSIRT) or Computer Emergency Response Team (CERT).
2) Extraction of information for cyberawarness purpose. The intelligence is used to have a general overview of the current situation. It helps to do prevention and educate the public to cybersecurity.
3) Detection of any cybersecurity-related events to make a snapshot of the threat landscape. It is useful to depict the cybersecurity environment and follow the evolution over time.

The detection of vulnerabilities is common to the first and third objectives. The CVE (Common Vulnerabilities and Exposures) website defines a vulnerability as "a weakness in the computational logic (e.g. code) found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, OR availability." [5].

In the rest of this paper, we will discuss how to detect vulnerabilities using crowd sourced detection.

The remainder of the paper is structured as follows. In Section 2 we present a state of the art on the solutions used to detect vulnerabilities using OSN as input data. In Section 3 we discuss the limitations and future challenges of such solutions. Finally, we conclude in section 4.

II. State of the art

The first vulnerability detection solution using OSN comes from Sabottke et. al. [12] in 2015. Since then, other solutions have been proposed and most of them, if not all, are using Twitter as their source of input data. Twitter provides an easily accessible and free API, which explains why researchers tend to work on this OSN. In addition, Twitter can be used by whistleblowers to disclose important information and is a place where many experts discuss. This results in a stream of messages with a lot of relevant information in it. However, these solutions are generic enough to be adapted to other OSN. There is a need to compare them, to identify which one is more adapted to the needs one can have. Altalhi and Gutub [2] proposed seven factors to compare the solutions. For each factor, they assigned a percentage score based on how much the solution helps reach a subjective goal the authors set. This score is subjective and does not help to compare the solutions, therefore it will not be discussed in the rest of the paper. We will now briefly explain the factors:

- Detection scope: how generic is the intelligence collected? Does the solution detect only vulnerabilities, cyberattacks or every security-related topics?
- Feature extraction: the technic used to extract features from text.
- Algorithm complexity: the name of this factor could be misleading, this factor is the name of the algorithm used to classify or cluster the features.
- Summarisation: how the information is restituted. Is it keywords, bulk of messages or messages extended with external sources?
- Scalability over time: this factor is about automation and how much and where we need to manually update the system.

- Performance: quantitative metrics used to compare the solution. Interestingly enough, Altalhi and Gutub [2] decided to put precision and recall in the same factor. On the other hand, we decided to split the performance in two and will have one column for precision and another for recall.

We will not keep the feature extraction and algorithm complexity factors as they do not help to identify which solution is better. We propose to add three additional factors. The first one is the security of the execution: how resilient the detection is to an attack. In other words, is it possible to make the solution detect false vulnerabilities or at the opposite, miss real vulnerabilities? We already discussed this factor further in one of our previous papers [4]. The second and third ones are the proportion of vulnerabilities detected on Twitter before their publication on an official platform like the National Vulnerability Database [1] (we will call this factor early detection) and the average time delta between detection on Twitter and official release. Obviously, if the solution detects every vulnerability, but always later than their official release, the solution is worthless.

To summarise, we will use these seven factors to compare the solutions used to extract vulnerability from OSN : detection scope, summarisation, scalability, security of the detection, precision, recall, early detection and average time delta between detection. Altalhi and Gutub [2] already proposed a review of vulnerability detection solutions [13] [12] [6] [9] [11] but they did not compare them. We propose to complete their review with the comparison of the solutions, which can be found in I. This table will help to better identify the limitations and future challenges.

III. Future challenges

In their respective articles, authors provide proof that their solutions are working. Some of them use the precision or the recall of their solution to measure how well it performs. Despite such measurements, we identified that some important technical limitations are actually either partially addressed or not addressed at all in most papers. These limitations question the usability, reliability and efficiency of vulnerability detection solution using OSN.

A. Quality of the input data

The first step of every vulnerability detection solution workflow is to gather relevant messages. It is already a challenging task as OSN have both valuable messages and promotional or spam messages. These messages decrease the overall quality of the input data. In addition, messages on OSN are written with spelling mistakes, slang and emoticons which make the messages harder to process. In the first place, basic natural language preprocessing steps can be performed to improve the quality of the input data. Naseem et. al. [10] propose a list of preprocessing steps to improve short text processing. In a second time, using a list of keywords to query the Twitter API helps to "avoid a lot of noises: we receive only tweets that contain the relevant keywords hence threat are highly probable to be relevant". [8]

"The drawback of the keyword-based collection method for cyberthreat information is that this method requires expert knowledge about cyberthreats to choose the relevant keywords. The keyword-based collection method, therefore, can easily ignore cyberthreat-related information and collects cyberthreat irrelevant information if the keywords are not carefully selected" [**?**]. It also requires a constant updating process since the relevant keywords for cybersecurity are evolving over time. It's either manually or automatically done. In the first case, this causes problems in terms of scalability; will the experts be able to process the amount of new security keywords in a reasonable period ? In the case of an automatic update, it exposes our solution to adversarial machine learning. We will go more in depth in the following subsection.

Another problem is that we collect a lot of tweets which are promotional. For example, take the two following messages: "New ransomware derived from Petya found." and "Buy our new state-of-the-art solution against ransomware". The word "ransomware" appears in both relevant tweets and promotional one. A promotional tweet is a legitimate usage of an OSN; therefore this cannot be considered as a spam, but we still need a solution to discard them. The intuitive solution would be to use a blacklist of words or train a machine learning algorithm to distinguish promotional tweets from relevant tweets. The drawbacks of this approach are the same as to the ones listed in the paragraph about list of keywords.

An alternative to this keyword-base method is to use a novelty classifier. Novelty classifier needs to be trained only with positive samples, it is a positive unlabelled (PU) learning problem [3]. A PU learning problem arises when the data cannot be entirely labelled because of a lack of resources, time, too many data or when negative examples cannot be clearly identified. This is the case in our problem. We cannot manually label all the tweets and it is hard to identify a representative set of non-relevant tweets. Therefore, Le et. al. [7] proposed to train a novelty classifier on NVD, a CVE database, and then to use it to classify tweets. The tweets similar to CVE description are the one considered as relevant, the others are not. Alternatively, we can use abstracts from newspapers as they are short texts and provide a lot of information. The drawback of this method is a bias it introduces in the format of the tweets. Tweets are informal text with slang specific to Twitter, while CVE description and press articles and rigorous and formal. There is a risk to discard relevant tweets that are not written in the usual way.

A solution often seen in the literature to improve the quality of the tweets gathered is the usage of a whitelist of users known for the quality of their tweets and the relevance regarding the subject. Once again, the usage of a list asks the question of how it is kept up to date. By doing this, we also centralise the information and we lose the possibility of discovering new cybersecurity-related events coming from new or less known users. For example, we will miss the messages

[1] https://nvd.nist.gov/

Paper	Detection Scope	Summarisation	Scalability over time	Security Measures	Recall	Precision	Early detection	Time Delta
Sabottke et. al. (2015) [12]	Real World Exploit, Proof-of-Concept Exploit	CVE number	List of users	Restricted users	Provided in the paper	Provided in the paper	Provided in the paper	Provided in the paper
Trabelsi et. al. (2015) [13]	Zero-day	Bulk of Tweets	List of keywords	Evaluation of user's trustfulness	Provided in the paper	-	Provided in the paper	Provided in the paper
Kergl et. al. (2016) [6]	Zero-day	Bulk of Tweets enriched with url	List of keywords	None	-	-	-	-
Mittal et. al. (2016) [9]	Vulnerability	Structured summary of an alert	List of keywords	None	-	-	-	-
Queiroz et. al. (2017) [11]	Software Vulnerability	Tweet	List of users	Restricted users	Provided in the paper	Provided in the paper	-	-

TABLE I
COMPARISON OF VULNERABILITY DETECTION SOLUTIONS

of whistleblowers.

To summarise, we did not identify in the literature a panacea solution to extract high quality input data. Either we give up on some reasons why we turn to OSN in the first place, whether we accept to have input data of disputable quality for which we cannot evaluate the actual quality.

B. Resiliency against adversaries

Another problem of the solutions trying to identify vulnerabilities thanks to OSN is that, most of the time, they do not take into account potential adversarial threat. In a previous work, we proposed a first contribution for a threat model for event detection algorithms, which are a specific class of algorithm extracting knowledge from OSN [4]. The identified attacks may be specific to the technology studied in the paper, but the threat actors, the hypothesis and some defence strategies identified are still relevant for any kind of algorithm taking OSN messages as input. We emphasise the fact that training and test dataset, algorithms and ground truth are, most of the time, publicly available; therefore the attacker has at least partial knowledge of how our solution works.

On the mitigation technique used in the literature, the implementation of a whitelist of users who are known to make valuable messages on security is predominant. We already explained the problem from a data-oriented perspective, but we will now develop the problem from a security-oriented perspective. The whitelist is not a problem in itself, it's more about how it is updated. If the list is manually updated, the security of the process relies on how well the updating process is defined and how well the operator applies it. If the updating process is automated thanks to machine learning, then it becomes highly vulnerable to adversarial learning [14] and it will be easy to add malicious users to the whitelist, which will go against the initial interest of the whitelist.

In addition, [1] identified multiple attacks possible against OSN text processing applications. Such concern is never discussed in the vulnerability detection solutions previously presented. It's common knowledge that implementing security features change the result of an algorithm, most of the time resulting in a decrease in the performance. The systemic absence of these security measures in the literature could mislead the community by overestimating the effectiveness of vulnerability detector using OSN.

To summarise, vulnerability detection thanks to OSN may work now, but is not at all secure by design. This is concerning because we are trying to extract highly valuable intelligence for attackers and various actors in the cybersecurity community and the better the technology becomes, the more the risk of adversarial threat will increase; therefore, defence against adversarial threat should be considered as a building block and not a feature.

C. Evaluation of the solutions

As outlined in Table I, only one solution out of five provides all the metrics needed to evaluate the solution. It is problematic as it does not allow comparison between the solutions proposed. In addition, without all the metrics, it becomes harder to argue over the performance and the relevance of the solutions. There are four metrics we need to measure to test if a vulnerability detection solution works.

The first one is the precision of the detection. It can easily be computed by dividing the number of vulnerabilities detected by the number of detection the solution raised. It is an important metrics because if the precision is too low, there will be no advantage over manually processing the messages.

The second one is the recall. It can be measured by dividing the number of identified vulnerabilities by the number of vulnerabilities posted on NVD in the same period. The recall is probably the most important metrics in our case, as it allows evaluating the coverage of the detection.

The third one is the proportion of vulnerability discovered before officials released. The official release date could be found on a website such as CVE or NVD. If the solution has a more restricted scope and only focuses on the detection of vulnerability exploited in the wild, then we can use the creation date of rules related to a vulnerability for antivirus or IDS as the "official release date" of the exploit. The fourth metrics is related to the third one and it is the average

time delta between the official release date and the detection, for the vulnerabilities discovered before officials released. Indeed, if the proportion of vulnerabilities detected before their official release date is too small, then the solution does not provide actionable intelligence. In addition, if that proportion is detected only one day or two before official release, it could be valuable, but not worth the amount of effort needed to deploy such a solution.

IV. Conclusion

The literature tends to demonstrate that vulnerability identification thanks to OSN is working and provides satisfactory results. On the other hand, we clearly identified three future challenges that are currently unaddressed by the community, which are 1) the quality of the input data, 2) the security of the solution and 3) a better evaluation method of the results. Concerning the third point, we formalised four metrics necessary to better evaluate the performance of the solutions. Without these three aspects studied, it is hard to provide proof that this technology will be reliable over time. We seek in future works to extend this analysis not only to vulnerability detectors, but all solutions which extract cybersecurity intelligence from OSN.

References

[1] Alsmadi, I., Ahmad, K., Nazzal, M., Alam, F., Al-Fuqaha, A., Khreishah, A., Algosaibi, A.: Adversarial Attacks and Defenses for Social Network Text Processing Applications: Techniques, Challenges and Future Research Directions (Oct 2021)

[2] Altalhi, S., Gutub, A.: A survey on predictions of cyber-attacks utilizing real-time twitter tracing recognition. Journal of Ambient Intelligence and Humanized Computing **12**(11), 10209–10221 (Nov 2021)

[3] Bekker, J., Davis, J.: Learning from positive and unlabeled data: a survey. Machine Learning **109**(4), 719–760 (Apr 2020)

[4] de Casanove, O., Sèdes, F.: Malicious human behaviour in information system security: Contribution to a Threat Model for Event Detection Algorithms. In: 15th Symposium on Foundations and Practice of Security (FPS 2022). Ottawa, Canada (Dec 2022), 14 pages ; Session 4: Privacy

[5] CVE: Cve numbering authority (cna) rules (Feb 2020)

[6] Kergl, D., Roedler, R., Rodosek, G.D.: Detection of Zero Day Exploits Using Real-Time Social Media Streams. In: Pillay, N., Engelbrecht, A.P., Abraham, A., du Plessis, M.C., Snášel, V., Muda, A.K. (eds.) Advances in Nature and Biologically Inspired Computing. pp. 405–416. Advances in Intelligent Systems and Computing, Springer International Publishing, Cham (2016)

[7] Le, B.D., Wang, G., Nasim, M., Babar, A.: Gathering Cyber Threat Intelligence from Twitter Using Novelty Classification. Tech. rep. (Sep 2019)

[8] Le Sceller, Q., Karbab, E.B., Debbabi, M., Iqbal, F.: SONAR: Automatic Detection of Cyber Security Events over the Twitter Stream. In: Proceedings of the 12th International Conference on Availability, Reliability and Security. pp. 1–11. ARES '17, Association for Computing Machinery, New York, NY, USA (Aug 2017)

[9] Mittal, S., Das, P.K., Mulwad, V., Joshi, A., Finin, T.: 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). pp. 860–867 (Aug 2016)

[10] Naseem, U., Razzak, I., Eklund, P.W.: A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter. Multimedia Tools and Applications pp. 1–28 (2020)

[11] Queiroz, A., Keegan, B., Mtenzi, F.: Predicting Software Vulnerability Using Security Discussion in Social Media (2017)

[12] Sabottke, C., Suciu, O., Dumitras, T.: Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits. In: 24th USENIX Security Symposium (USENIX Security 15). pp. 1041–1056. USENIX Association, Washington, D.C. (Aug 2015)

[13] Trabelsi, S., Plate, H., Abida, A., Ben Aoun, M.M., Zouaoui, A., Missaoui, C., Gharbi, S., Ayari, A.: Mining social networks for software vulnerabilities monitoring. In: 2015 7th International Conference on New Technologies, Mobility and Security (NTMS). pp. 1–7 (Jul 2015), iSSN: 2157-4960

[14] Wang, X., Li, J., Kuang, X., Tan, Y.a., Li, J.: The security of machine learning in an adversarial setting: A survey. Journal of Parallel and Distributed Computing **130**, 12–23 (Aug 2019)