I. J. Education and Management Engineering, **2021, 3, 11-19**
Published Online June 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijeme.2021.03.02

A New Vulnerability Reporting Framework for Software Vulnerability Databases

Hakan Kekül
University of Fırat, Institute of Science, Elazığ Turkey.
Sivas Information Technology Technical High School, Diriliş Mahallesi Rüzgarli Sokak No 21 Sivas, Turkey.
Email: hakankekul@gmail.com

Burhan Ergen
University of Fırat, Faculty of Engineering, Computer Engineering Department, Elazığ Turkey.
Email: bergen@firat.edu.tr

Halil Arslan
University of Sivas Cumhuriyet, Faculty of Engineering, Computer Engineering Department, Sivas Turkey.
Email: harslan@cumhuriyet.edu.tr

Received: 07 January 2021; Accepted: 07 March 2021; Published: 08 June 2021

Abstract: Cyber security is one of the fundamental research areas of software engineering. The systems that make up today's information systems infrastructure have been developed largely with software support. Security vulnerabilities in the software used in these systems may cause undesirable results. It is very important to manage software vulnerabilities correctly. In addition, an effective communication mechanism and certain standards should be established among those working in this field. The importance of the subject has been understood in recent years and the studies in this area have gradually increased. The use of machine learning algorithms is increasing in recent studies in this area. Although there is a large data set accumulated in vulnerability databases, there is often the problem of unstructured data. Vulnerability databases and security reports are created in natural language that people can understand and interpret. These reports are difficult to read and understand by machines. Our study focuses on the difficulties of this unstructured and natural language system. In order to investigate this problem, firstly, up-to-date and accessible databases used in scientific research were examined and evaluated. Then, a three-stage security framework was proposed, consisting of the use of vulnerabilities by machines to assist experts from the notification stage to the reporting stage. The rules and flow charts of each stage are defined. In order to increase the usability of different databases in their own systems, the framework rules are defined as a guideline containing flexible directions, not rigid items. The point of consideration is not the methods and tools used, but the definition of outputs as common and similar attributes.

Index Terms: Software Security, Software Vulnerability, Vulnerability Databases, Cyber Security, Information Security.

1. Introduction

Cyber security emerges as an important problem since the day information systems entered our lives. Cyber security is basically in the literature; It is defined as the abuse of intellectual or actual property and personal information through acts that violate the working principles of the resources, processes and structures of systems with cyber space features [1]. In addition, cyber security is a science. Although it is considered as a sub-branch of computer science, it concerns a multi-disciplinary field. Its main purpose is to protect information systems and software under attack by using people and machines against hostile attacks [2]. Information systems have become an indispensable part of our social, economic and commercial life. This situation has caused the subject of cyber security to be one of the important research areas of today. One of the basic components of cyber security is software. Therefore, software has a very important place in the field of cyber security. Problems that may occur in any software system can cause many irreparable problems, including the loss of human life [3]. A lot of research is being done to find new techniques for vulnerabilities [4]. In this way, it can be determined which software are more vulnerable to risk [5].

Platforms where the detected security vulnerabilities are listed and shared with the public are called vulnerability databases. Security vulnerability databases have been created by different research groups that include software

vulnerabilities detected. The transactions until the software vulnerability is detected, reported, scored and published are carried out manually by experts from the moment the first record is kept.

There are different databases created by public support or private working groups. Although databases basically use the same dictionary, different data collection and reporting methods are used. When the vulnerability reports are examined, it is seen that it consists of identification number, product information affected by the vulnerability, a description describing the vulnerability, dates, author information, solution suggestions, exploit codes, references and the severity score of the vulnerability [6].

The reason behind the publication of this information is to ensure that application developers and users take action. Of course, published reports also carry the risk of abuse. However, it is an important resource for researchers and developers. Due to the increasing and accumulating security vulnerability data, unstructured structure and data size, it makes it difficult for people to make meaningful results [7].

It has become an important requirement to determine the security levels of software systems and to predict their vulnerabilities. There is a need for a road map to be followed for the development of secure software systems. In this sense, software vulnerabilities and security vulnerabilities detected in the past should be analyzed well. In this way, the same mistakes with past experiences will not be repeated. In this context, it has been observed that despite the increasing data size, a regular reporting and reporting standard could not be achieved. For this reason, the main purpose of the study was determined as to suggest a usable reporting and reporting framework for all databases by minimizing the problems caused by the human factor. For this purpose, we focus on the question of whether a common security vulnerability database framework can be created. This study brings about an innovation that will be an alternative to manual procedures, which is a big problem in the field. In addition, it will provide a guideline for the application of machine learning and data mining techniques in this particular problem.

Other parts of the study are organized as follows. In the second section, the literature guiding the study was examined in detail, in the third section, the research methodology and vulnerability database framework was presented in the third section, in the following section, the threats to validity were summarized, the results of the study were presented and future studies were expressed.

2. Related Works

The importance of software security vulnerabilities has been increasingly recognized due to the increased risks recently [8]. It is very important to manage detected software vulnerabilities. Correct coordination and information sharing between the researchers and stakeholders of the field is essential. In this way, attacks can be prevented by taking preventive measures against cyber attack [9]. Studies confirm that individuals who perceive themselves as vulnerable to a security threat are more likely to adopt an IS security innovation [10].

When any security vulnerability is detected, it is requested to be reported to the relevant organizations. This notification can be made by individuals, companies or institutions. The notification process can be performed through any vulnerability provider. Officially, the procedures provided by MITER company, which is funded and authorized by the United States Department of Homeland Security, are followed [11]. The official process is initiated by making the notification process through Common Vulnerabilities and Exposures (CVE), which is affiliated to the same company and is the main reference source for vulnerabilities.

CVE was established in 1999. It has an international activity on software security vulnerabilities. The CVE glossary provides a descriptive list of cyber security vulnerabilities. It defines itself as a dictionary rather than a database. All known large databases are basically created on the basis of the lists published in the CVE [12]. CVE lists are used by many academic and empirical studies [10,11].

Vulnerabilities in CVE database lists are analyzed by the National Vulnerability Database (NVD) and new attributes are added. NVD is a database created in 2000, affiliated with the National Institute of Standards and Technology (NIST). It is supported by the National Cyber Security Division of the US Department of Homeland Security. The data published in the NVD database mainly includes related impact metrics, (Common Vulnerability Rating System CVSS), types of vulnerability (Common Weakness Enumeration CWE) and other relevant metadata with applicability statements (Common Platform Enumeration - CPE) [15].

After the security vulnerabilities are published in the NVD database, the official notification is completed. However, the criticism that the definitions in the reports do not adequately express the relevant weakness is emphasized by the experts in the field [16]. With these criticisms, security vulnerability databases have been published by different organizations due to the problems caused by the institutional structures of CVE and NVD. These databases are the ones used in academic studies that can be accessed up-to-date and actively; Exploit-DB [17], SecurityFocus [18], Rapid7 [19], Snyk [20] and SARD [21] databases.

Moore et al. [22], reveals that concerns about the abuse of security vulnerabilities have increased recently. As a result, they state that it is necessary to quickly determine whether a security vulnerability can be exploited. In this way, any openness can be prevented from being exploited. This situation is possible by accelerating the evaluation processes of security vulnerabilities. The proposed method will contribute to this situation.

Kobek [23], It estimates that by the end of 2021, the damage from cybercrime will reach $ 6 trillion. An open source and customizable management is required to prevent damages caused by security vulnerabilities. One of the main goals of our study is to present the first conceptual suggestion on this path.

Russo et al. [24], emphasizes that publicly published vulnerability reports are written in natural language and can only be understood by experts. With these aspects, security vulnerabilities cannot be interpreted automatically by machines. This situation causes time delays and errors. It is aimed to accelerate the processes by supporting the experts with the proposed method.

Ruohonen [7], states that the number of security vulnerabilities is increasing day by day. He emphasizes that the size of the archive formed as a result of this makes it difficult to evaluate with the statistical methods used by researchers. The rate of increase in data size will accelerate further and the need for new approaches to processing these data will arise.

Theisen et al. [25], notes that software vulnerabilities have accumulated over the years and become a huge chunk of unstructured data. In this case, they state that the most accurate method for interpreting the data is machine learning algorithms that have been successfully applied in different problems. They emphasize that comprehensive analysis of the data cannot be made because these methods are not used for comprehensive analysis of the data. They state that this means that there will be points in the data that cannot be discovered as a result. The proposed method will contribute to the association and better interpretation of the information contained in security vulnerabilities.

Ghaffarian et al. [26], have provided a comprehensive review of many different academic studies using machine learning and data mining techniques that fall under software vulnerability analysis and discovery. The authors who dealt with the studies in the specified scope presented a detailed literature summary and made some suggestions in this context. The most striking thing here is that they emphasize the need for a common data set and feature engineering studies to be obtained and applied over this data set.

As can be seen in the studies examined, it was clear that there was a need to report and archive especially the accumulated and continuously available data with a new structure. Recently, researchers have recommended that feature engineering studies be carried out on data sets that can be used in machine learning algorithms [23, 24]. Machine learning algorithms are recommended to be used in this area. However, structured data sets with necessary attributes are needed to obtain successful results from studies. Because the reports published by databases are written in natural language, they are not suitable for direct use in machine learning algorithms.

3. Research Methodology

An accurate archiving and reporting method is essential in the face of the rapidly increasing number of vulnerabilities with its importance. In this part of the study, the methods used by existing databases are examined and an archiving and reporting method suitable for today's conditions is proposed. Thus, we tried to establish the answer to the question of whether a common security vulnerability database framework can be created, which we aimed within the scope of the study.

At this stage of our study, a security vulnerability database reporting framework that can be used directly with machine learning algorithms and understood by machines is not recommended. Within our framework, both old data can be converted and new data can be stored in a structured way. In this way, a system that can make much deeper statistical analysis can be created from an increasingly unstructured and difficult to understand structure.

3.1. Recommended Vulnerability Framework

To encourage information providers to write structured reports, there is a need to identify vulnerabilities and flaws, integrate the entire reporting and evaluation process. Taking into account the necessity and importance of a roadmap or framework for developing databases with basic and desired security features, an integrated and prescriptive framework is proposed here. We tried to make the proposed framework highly applicable and prescriptive in nature.

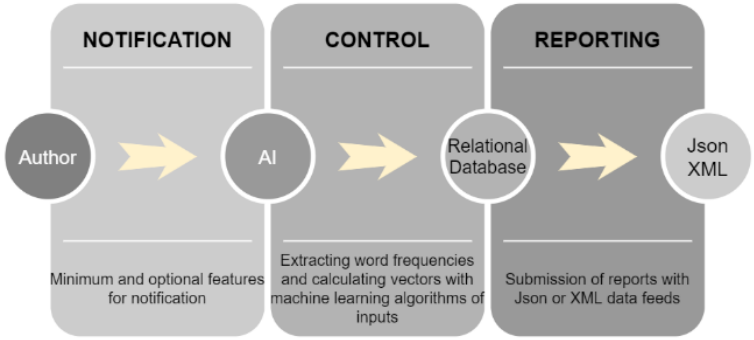

Fig. 1. An overall view of the proposed framework.

The vulnerability framework consists of three different phases. Actions to be taken at each stage are bound by a certain rule. However, in order to increase the acceptability of different databases under their own principles, complementary and basic components for a database have been proposed. Figure 1 shows an overview of the proposed

framework. The first stage is the reporting of vulnerabilities. The second phase is the control phase where vulnerabilities and flaws are classified and prioritized for identified vulnerabilities. In the third stage, which is the reporting stage, it is the layer where all the stored data can be presented to the end components in the form of JSON or XML instructions. A brief explanation of each is provided below to symbolically represent the vulnerability framework concepts, articulate their rules, and describe their stages. Figure 2 shows a general flow chart of the proposed framework.

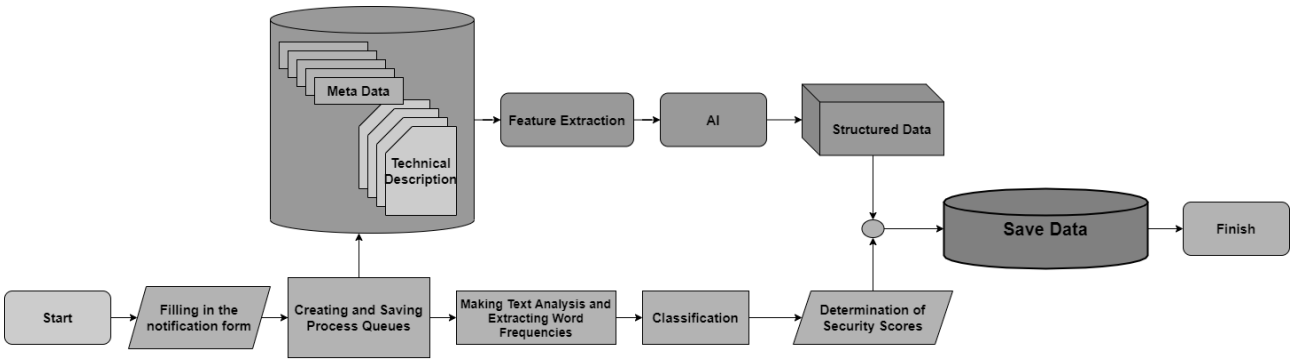

Fig. 2. Overall View of the Proposed Framework

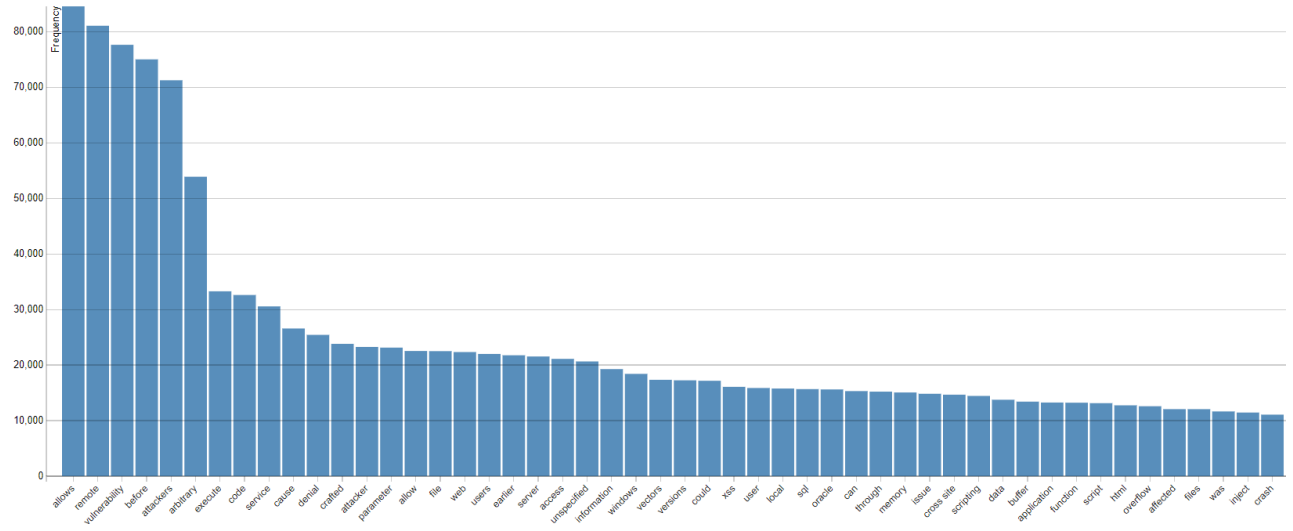

Fig. 3. Word Frequency List [28]

3.2. Notification Phase

This phase starts with the sending of a vulnerability warning notification to a database provider by individuals, institutions and companies that detect the vulnerability. This is done using a form filling or email template, although it is different for each database provider. After this stage, the relevant notification is evaluated by the authorized experts of each provider and the process continues. For example, CVE Numbering Authorities (CNA) have been defined for this process in the CVE database. The CNA analyzes the vulnerable functions identified in the notification and detects false notifications.

While we require a common notification feature list at the notification stage of our framework, whether or not different features are requested is left to the discretion of the providers. The steps for the execution and monitoring of the notification process are shown in Figure 4. Here, the basic fields specified in the vulnerability notification form detected by a database provider are defined with extra fields if required, provided that they are mandatory. With this definition, the notifying identifier can assign a priority that expresses the risk status of the related vulnerability. The priority value can be a metric value between 1 and 10, and this definition can be normalized with the high-normal-low range. The new vulnerability defined according to the determined priority value is transferred to the system via the relevant priority queue.

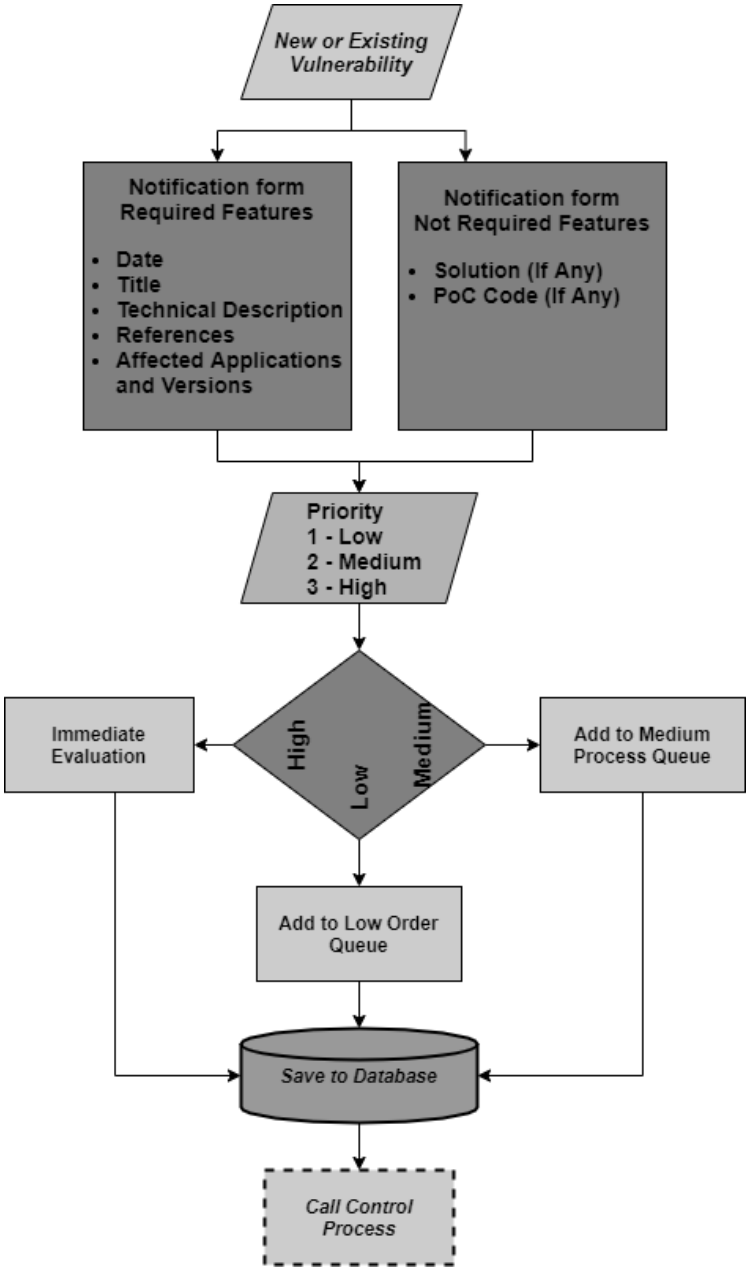

Fig. 4. Notification Stage

3.3. Control Phase

All of the security reports provided by current providers are written in natural language for people to understand. It is not possible for them to be understood as such by machines. Today, advances in artificial intelligence algorithms and natural language processing techniques can be applied to structure data. Especially for security vulnerabilities, the most important information is the technical explanations that experts enter while processing notification reports. In the first step of the control phase, we recommend adding the technical explanations to the data set as a new feature by removing the word frequencies. In this process, we believe that it would be appropriate to use the list of words provided by NVD, which includes the word frequencies of all technical explanations. Thus, we aim to ensure that scores can be predicted more accurately and experts are motivated to enter more specific descriptions by using a common word set for all database providers. Figure 3 shows NVD word frequencies.

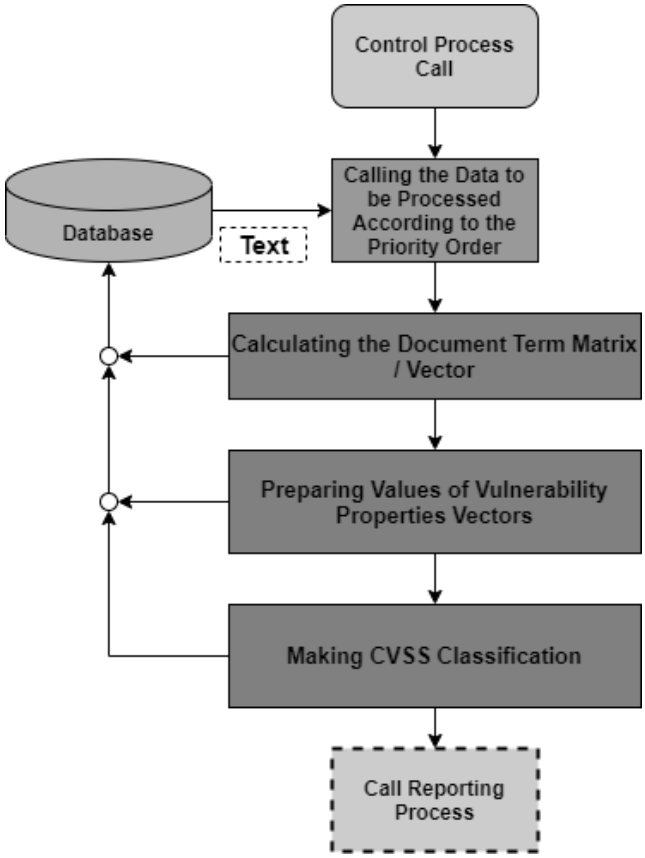

Fig. 5.Control Phase

In addition, the Common Vulnerability Scoring System (CVSS) is another important feature. However, not every database contains this information. The generally accepted standard for grading importance scores is CVSS. In addition, the Weighted Impact Vulnerability Scoring System (WIVSS) method, in which security vulnerabilities are calculated by weighting, has also been proposed in the literature [29]. The CVSS standard has been developed by NVD and is constantly updated. Currently versions 2.0 and 3.1 are officially used. CVSS 2.0 and WIVSS use the same set of metrics. Vulnerability metrics are used to determine severity. The values of these metrics are also determined manually by experts. Due to the manual determination of metrics, the accuracy of these values cannot be guaranteed due to their nature. At this stage, our framework proposes a support system created with machine learning algorithms to assist decision-making experts. In this structure, different algorithms can be used for classification purposes, or a multi-layered classification can be preferred according to their performance. The processes in the control phase are shown in Figure 5.

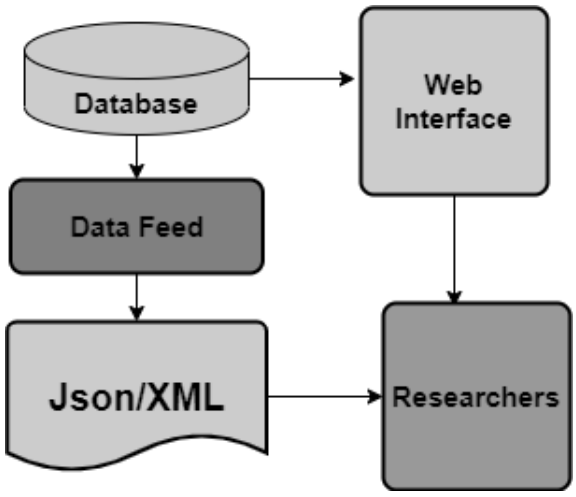

Fig. 6. Reporting Phase

3.4. Reporting Phase

A notification entering the system should be published on the web with a basic data feed technology after the basic features required in the control phase are calculated. At this point, the important thing is to present the calculated values together with the vector values and importance scores assigned by the experts. In this way, it will be possible to avoid errors caused by manual processes. In addition, it will be possible to make semantic inferences about a vulnerability by machines by presenting term matrices together with the explanation texts. At this stage, access to all data will be facilitated by using a data feeding technology. It will be possible to present data in JSON or XML format here. At this stage, with relational databases, data can be offered to users through an interface. The processes at the reporting stage are presented in Figure 6.

4. Threats to Validity

There may be points that we overlook in terms of the data we examine. It should be noted that all database information is extracted from official websites. The proposed framework was revealed with the information obtained from the researched databases and literature, and it was presented as a theoretical study. The fact that it has not been tested in this respect means that its deficiencies are not revealed clearly.

5. Conclusions

Natural language safety reports are structures created by people with their past experiences and knowledge. This situation results in missing important points or not fully understanding the deficit. With the proposed framework, a framework that will make sense of the views of all experts operating in the system and evaluate new deficits according to the results of past deficits is proposed. The most important point of the system is the ability of the security reports, which reach half a million in total, to evaluate all gained experiences. In addition, the fact that the data is presented by structuring will contribute to increase the performance with different machine learning algorithms. While the framework was proposed, the stages were created with basic principles rather than defined by strict rules. The reason for this is to allow different database providers or researchers to integrate their ideas into the system. In this way, it will provide motivation to increase performance among databases. In addition, the purpose of the framework is to ensure that the printouts of security reports are highly understandable by machines. For this reason, no limitation has been determined for the methods used in the control phase. The proposed framework is open to development and contribution with its customizable structure. It can also be updated due to the open source principle. The model offers a foresight that will form the basis for the work to be done in this field. It will contribute to future studies due to its testable nature with different feature extraction methods and classification algorithms.

6. Future Works

In our future work, we intend to implement and verify the proposed framework with different algorithms. In this way, we can identify the shortcomings of our frame. First of all, we plan to present our study, which includes the analysis of all available and up-to-date security vulnerability databases. Next, we will present models that will perform feature extraction and classification processes that will validate and test the proposed framework. We aim to improve these processes by adding statistical and deep learning-based current methods.

CRediT authorship contribution statement

Hakan KEKÜL: Conceptualization, Methodology, Validation, Formal analysis, Resources, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization. **Burhan ERGEN:** Conceptualization, Methodology, Validation, Formal analysis, Writing - Review & Editing, Supervision, Project administration. **Halil ARSLAN:** Conceptualization, Methodology, Validation, Formal analysis, Writing - Original Draft, Writing - Review & Editing, Supervision

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding Sources

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] D. Craigen, N. Diakun-Thibault, and R. Purse, "Defining cybersecurity," *Technol. Innov. Manag. Rev.*, vol. 4, no. 10, 2014.
[2] F. Chang, "Guest Editor's Column," *Next Wave*, vol. 4, no. 19, pp. 1–2, 2012.
[3] B. S. Cruz and M. de Oliveira Dias, "CRASHED BOEING 737-MAX: FATALITIES OR MALPRACTICE?," *GSJ*, vol. 8, no. 1, pp. 2615–2624, 2020.
[4] M. M. A. Muhammad Noman Khalid, Muhammad iqbal, Kamran Rasheed, "Web Vulnerability Finder (WVF): Automated Black- Box Web Vulnerability Scanner," *Int. J. Inf. Technol. Comput. Sci.*, vol. 12, no. 4, pp. 38–46, 2020.
[5] C. P. T. Pubudu K. Hitigala Kaluarachchilage, Champike Attanayake, Sasith Rajasooriya, "An Analytical Approach to Assess and Compare the Vulnerability Risk of Operating Systems," *Int. J. Comput. Netw. Inf. Secur.*, vol. 12, no. 2, pp. 1–10, 2020.
[6] S. Zhang, X. Ou, and D. Caragea, "Predicting Cyber Risks through National Vulnerability Database," *Inf. Secur. J. A Glob. Perspect.*, vol. 24, no. 4–6, pp. 194–206, 2015.
[7] J. Ruohonen, "A look at the time delays in CVSS vulnerability scoring," *Appl. Comput. Informatics*, vol. 15, no. 2, pp. 129–135, 2019.
[8] A. Kuehn and M. Mueller, "Shifts in the cybersecurity paradigm: Zero-day exploits, discourse, and emerging institutions," in *Proceedings of the 2014 New Security Paradigms Workshop*, 2014, pp. 63–68.
[9] O. Bozoklu and C. Z. Çil, "Yazılım Güvenlik Açığı Ekosistemi Ve Türkiye'deki Durum Değerlendirmesi," *Uluslararası Bilgi Güvenliği Mühendisliği Derg.*, vol. 3, no. 1, pp. 6–26, 2017.
[10] C. W. Samuel Ndichu, Sylvester McOyowo, Henry Okoyo, "A Remote Access Security Model based on Vulnerability Management," *Int. J. Inf. Technol. Comput. Sci.*, vol. 12, no. 5, pp. 38–51, 2020.
[11] "Mitre Corporation," 2020. [Online]. Available: https://www.mitre.org. [Accessed: 25-Jul-2020].
[12] CVE, "CVE," *Common Vulnerabilities and Exposures*, 2020. [Online]. Available: https://cve.mitre.org. [Accessed: 25-Jul-2020].
[13] G. Schryen, "Security of open source and closed source software: An empirical comparison of published vulnerabilities," *AMCIS 2009 Proc.*, p. 387, 2009.
[14] G. Schryen, "Is Open Source Security a Myth?," *Commun. ACM*, vol. 54, no. 5, pp. 130–140, May 2011.
[15] NVD, "NVD," *National Vulnerability Database*, 2020. [Online]. Available: https://nvd.nist.gov. [Accessed: 25-Jul-2020].
[16] Y. Fang, Y. Liu, C. Huang, and L. Liu, "Fastembed: Predicting vulnerability exploitation possibility based on ensemble machine learning algorithm," *PLoS One*, vol. 15, no. 2, pp. 1–28, 2020.
[17] ExploitDB, "Exploit Database," 2020. [Online]. Available: https://www.exploit-db.com. [Accessed: 25-Jul-2020].
[18] SecurityFocus, "SecurityFocus," 2020. [Online]. Available: https://www.securityfocus.com. [Accessed: 25-Jul-2020].
[19] Rapid7, "Rapid7," 2020. [Online]. Available: https://www.rapid7.com/db/. [Accessed: 25-Jul-2020].
[20] Snyk, "Snyk," 2020. [Online]. Available: https://snyk.io. [Accessed: 25-Jul-2020].
[21] SARD, "SARD-Software Assurance Reference Dataset Project," 2020. [Online]. Available: https://samate.nist.gov. [Accessed: 25-Jul-2020].
[22] T. W. Moore, C. W. Probst, K. Rannenberg, and M. van Eeten, "Assessing ICT Security Risks in Socio-Technical Systems (Dagstuhl Seminar 16461)," *Dagstuhl Reports*, vol. 6, no. 11, pp. 63–89, 2017.
[23] L. P. Kobek, "The State of Cybersecurity in Mexico: An Overview," *Wilson Centre's Mex. Institute, Jan*, 2017.
[24] E. R. Russo, A. Di Sorbo, C. A. Visaggio, and G. Canfora, "Summarizing vulnerabilities' descriptions to support experts during vulnerability assessment activities," *J. Syst. Softw.*, vol. 156, pp. 84–99, 2019.
[25] C. Theisen and L. Williams, "Better together: Comparing vulnerability prediction models," *Inf. Softw. Technol.*, vol. 119, no. August 2019, 2020.
[26] S. M. Ghaffarian and H. R. Shahriari, "Software vulnerability analysis and discovery using machine-learning and data-mining techniques: A survey," *ACM Comput. Surv.*, vol. 50, no. 4, 2017.
[27] G. Spanos and L. Angelis, "A multi-target approach to estimate software vulnerability characteristics and severity scores," *J. Syst. Softw.*, vol. 146, pp. 152–166, 2018.
[28] "Description Summary Word Frequency," 2021. [Online]. Available: https://nvd.nist.gov/general/visualizations/vulnerability-visualizations/vuln-description-summary-word-frequency. [Accessed: 02-Jan-2021].
[29] G. Spanos, A. Sioziou, and L. Angelis, "WIVSS: A New Methodology for Scoring Information Systems Vulnerabilities," in *Proceedings of the 17th Panhellenic Conference on Informatics*, 2013, pp. 83–90.

Authors' Profiles

Hakan Kekül is currently working as a teacher at Sivas Information Technology Technical High School. In 2018, he received his undergraduate degree from Sakarya University, Department of Electronics and Computer Education. In 2018, he received his bachelor's degree in Computer Engineering from Cumhuriyet University. In 2017, he received his Master's degree from Cumhuriyet University. Since 2018, he is a PhD candidate at Fırat University, Department of Computer Engineering.

Burhan Ergen is currently Asst. Prof. in Department of Computer Engineering at Fırat University. He received his BS degree in Electronics Engineering from Karadeniz Technical University in 1993. He received his master's degree from Karadeniz Technical University in 1996 and his doctorate degree from Fırat University in 2004. He is currently working at Fırat University.

Halil Arslan is currently a faculty member at Sivas Cumhuriyet University Computer Engineering Department. He received his undergraduate, graduate and doctorate degrees from Sakarya University Electronics and Computer Education Department in 2006, 2008 and 2016, respectively. He is currently working at Sivas Cumhuriyet University

How to cite this paper: Hakan Kekül, Burhan Ergen, Halil Arslan, " A New Vulnerability Reporting Framework for Software Vulnerability Databases", International Journal of Education and Management Engineering (IJEME), Vol.11, No.3, pp. 11-19, 2021. DOI: 10.5815/ijeme.2021.03.02