Automated Analysis of Topics on Security in Research Papers

Paul POCATILU[1], Alin ZAMFIROIU[1,2]*, Vlad APOSTOL[1]

[1] Bucharest University of Economic Studies, Romania

[2] National Institute for Research and Development in Informatics, Romania

ppaul@ase.ro, alin.zamfiroiu@csie.ase.ro (**Corresponding author*), vladdapostol@gmail.com

Abstract: Cybersecurity is a major concern in the field of information and communication technology. Topics like risk, vulnerability, exploit, attack, threat are related to security, thus revealing its complex nature. In order to take better security measures, vulnerabilities have to be identified. These topics are the subject of research papers in journals, and of Master's, doctoral and postdoctoral theses. This paper proposes a model and several metrics for the analysis of topics related to vulnerabilities in scientific papers. This model has been validated based on an automated tool by analyzing over 400 research papers. The Common Vulnerabilities and Exposures (CVE) database was used as the main reference source for the existing vulnerabilities.

Keywords: Security, Vulnerabilities, CVE, Correlations, Automated Analysis.

1. Introduction

The vulnerability of a system could be described as a weakness or a flaw that allows black-hat hackers to violate the system's security in order to deliver their malicious intent or white-hat hackers to reveal the exposure to the general public or to the companies that own the system (Wang et al., 2020). Thus, it is important to analyze these vulnerabilities and action in the interest of providing countermeasures against them.

There are two main institutes that are continuously investing in this domain, the MITRE Corporation and NIST (National Institute of Standards and Technology). The MITRE Corporation (Massachusetts Institute of Technology Research & Engineering) is the first CNA (CVE Numbering Authority) and it has authorized over 120 other CNAs in 21 countries including big companies like Oracle, Adobe, Microsoft Corporation, Google and others. CNAs are able to identify vulnerabilities within their proprietary systems and publicly assess a CVE identifier for that certain weakness in the system. An entry in CVE database represents a way of identifying a certain vulnerability which was registered by a certain CNA and contains an identification code, a description, some references and other details regarding the weakness. The field of common vulnerabilities will evolve in a proper direction as along with the evolution of the IT industry, the number of flaws in the system will increasingly grow. NIST is part of the United States government and it owns a public database of CVEs, called NVD (National Vulnerability Database), together with the MITRE Corporation.

In order to further understand how the research in the field of vulnerabilities is conducted, first it should be specified how a common vulnerability is scored, which are the main factors that influence the score of a CVE item from the database, how that score became a standard for the IT industry and why there is a certain opposition against a common standard that was set with the purpose of clarifying and facilitating the process of scoring a flaw within a system.

This paper presents the results of a study that aims to highlight how the security-related topics are approached by the researchers in this field.

This paper is structured as follows. Section 2 presents the background of this research and the major studies related to topic of this paper. Section 3 sets forth the employed methodology and datasets. Section 4 discusses the results and findings. Finally, Section 5 includes the conclusion and the proposed future work.

2. Background

The CVSS (Common Vulnerability Scoring System) is a standard that has been adopted by many organizations and companies so that if a CVE item has been discovered, it can also be scored. The CVSS works by taking

https://doi.org/10.24846/v29i4y202007

into account multiple factors that define the subject vulnerability. The base score is then computed depending on multiple metrics with CVSS calculator (FIRST, Inc., 2019), the main ones being:

- *access vector* - describes the ways in which an individual can take advantage of the vulnerability, either remotely (for example, by using a packet sniffer) or locally, physically (which require the attacker to connect a device to the system, such as a network tap, or to gain access from an adjacent network, for example through Bluetooth or a wireless local area network), the remote access generates the highest possible risk;

- *access complexity* - which measures the difficulty of exploiting the respective weakness;

- *authentication method* - which assesses how many times the person who exploits the vulnerability should authenticate; this can be without this possibility, once or several times;

- *confidentiality impact*; a complete confidentiality impact is the worst as it exposes the entire system to the perpetrator, while a partial impact would only provide access to a part of the files of the system, and most probably random ones; confidentiality impact can be either partial or complete;

- *integrity impact* considers the possible modification of files of the targeted system, similarly to the confidentiality impact, it can have a value like partial or complete;

- *availability impact* could lead the system to a complete shutdown, a partial outage (like a DDoS attack), or not affect the system at all.

All of the terms described above, evaluated together, give us the ability to record the flaw found in the system with a base score. Even though the base score is the most representative one for understand the risk of an identified vulnerability, there is also a temporal and environmental score.

The temporal score is computed based on the remediation level, the exploitability score and the report confidence, while the environmental score has certain defined metrics, but it basically depends on the system of each vendor that scores a certain vulnerability, as every company and organization is different, and so are the needs and characteristics of their systems. This could lead to some challenges for the scoring system.

Having a common scoring system for vulnerabilities provides a great benefit, as it gives someone the ability to easily comprehend the risk that comes with the respective. However, there are certain factors that prevent the common scoring system to work as an open industry standard that could be adopted by many entities.

First of all, one of the issues is that the vendors don't want to report their identified vulnerabilities, as this would be a general risk for their company and would also put pressure on solving that vulnerability before it gets disclosed to the general public, even though they will get a grace period in order to find a solution for the vulnerability. Furthermore, some companies perceive the disclosure of weaknesses in their systems as a general brand risk that may affect their image. To illustrate this, let's consider a smartphone company that uses a proprietary operating system for their devices. A large number of vulnerabilities identified within their system and disclosed to the public would certainly affect the image of their brand, resulting in a lower brand trust within their target market.

In addition to the problematic factors of a common scoring system, there is also the environmental factor stated before. The environmental factor is a problematic issue for a standard because each company and their systems function in a unique way, and would need special metrics for vulnerabilities found within their systems and the risks that come with them. For this reason, the CVSS chose to have the environmental score not integrated within the base score of a vulnerability, so the companies that adopt this scoring system have the freedom to identify their vulnerabilities by making abstraction of the environmental factor and treating it as a separate object. Still, some organizations favor the USA for their in-house vulnerability scoring system against a common standard, even though in the long run a common standard would benefit all

the parties as companies tend to have the same level of technology used within them and more thorough analyses could be conducted, giving everyone a better understanding of the evolution of the technology threats around us.

Thirdly, another factor that needs to be taken into consideration regarding the problems specific to an open standard scoring system would be the biased base score related to certain vulnerabilities. This problem arises from having multiple vulnerabilities that behave differently, but which end up having the same base score. Some of the metrics of the vulnerabilities could be at opposite ends on the intervals and in the end still provide the same base score. One solution for this problem may be splitting the base score into multiple more concentrated scores, however this could lead to some observers to confusion, and lower the adoption rate of the scoring system. Returning to the subject of biased scores, the human bias must also be taken into consideration, as the experience of different security experts might differ from one individual to another.

Some of the recent researches carried out on the subject of CVEs (Chang et al., 2011; Glanz et al., 2015; Pham & Dang, 2018) reveal different opinions regarding the current scoring system that is used as a standard and the wide differences in the analysis of the vulnerabilities among all the databases that the community of the researchers in this field offers (Allodi et al., 2020). There are several voices stating that particular vulnerability databases have a large number of questionable scores upon further investigation, as they do not represent the real risk that the vulnerabilities involve. There are some recent researchers who have analyzed the way on how the research papers have been published and the software used to manage and support scholarly publications such as Boja et al. (2018), Sandu et al. (2019) or Li et al. (2020).

Johnson et al. (2016) conducted a research, using a Bayesian analysis, concerning the most used vulnerability databases which considers the reliability and validity of the data each database entity provides for their vulnerability scores. A Bayesian analysis is expected to deliver results about some unknown parameters, the base score in this case, by using probabilistic statements. The analysis considered only the base scores of some vulnerabilities found in each database and concluded that a couple of the most reliable databases would be NVD and Cisco. Consequently, considering the current state of the databases that provide information about common vulnerabilities, the presented data will rely almost entirely on the data administered by the U.S. National Vulnerability Database.

There are also other researches on the matter of CVE, namely (Wu et al., 2020), and one can understand that they have used the CVE entries to design an automatic tool for conducting a large-scale dataset for security bug report prediction.

Considering the amount of research that has been carried out in the field of vulnerabilities and their scoring system, there are still some doubts in the community that affect the overall trust in the CVSS standard. Most importantly, the CVSS was introduced as a standard designed to suggest the severity that a certain vulnerability poses and the threats that come with it (Ruohonen, 2019). However, some parts of the general public insist on promoting the idea that a standard scoring system should also express the amount of times a vulnerability has been or will be exploited without understanding that such a matter would be in fact almost impossible or very difficult to estimate and predict. Given the above, it can be stated that the current CVSS has its advantages and disadvantages as it keeps evolving and getting updated with a better documentation, subsequently being a step in the right direction with regard to the mitigation of security issues around the world, raising awareness of the software engineers about the threats that they may encounter when developing a system and providing faster responses to zero-day vulnerabilities.

In (Sauerwein et al., 2019) a comprehensive analysis and classification of public information security data sources used in research and practice is provided. This analysis aims to study the vulnerabilities of keywords from research papers in correlation with the subject of those research papers, which is the topic of this paper.

A framework for supporting the analysis of logs produced by security attacks prevention and detection tools is provided by Guzzo et al. (2020). An algorithm and models for the identification of the targets are proposed and presented in this paper. The authors have also presented the implementation issues and an experimental evaluation of this algorithm.

In (Williams et al., 2020) a framework that uses Topically Supervised Evolution Model (TSEM) model to discover relationships between vulnerabilities and predisposition of software products to attacks and risks is proposed. This work presents the labels of each vulnerability that was analyzed. Similarly, the same keywords from this research paper will be used in this paper to analyze the existing vulnerabilities.

Other proposed solutions like (Angelini et. al., 2018), (Tan et al., 2019) (Elbaz et al., 2019) and (Zamfiroiu et. al., 2020) deal with creating tools that would enable an automated analysis of risks or existing vulnerabilities for some topics or solutions.

3. Methodology and Datasets

3.1 Methodology

The proposed methodology consists in an analysis of certain electronic documents, based on the frequency of occurrence of certain key terms related to security in order to achieve a hierarchy of words for each document. The Document Impact Score (DIS) indicator as the average of Impact Scores of the vulnerabilities identified for some keywords from each document was proposed. In (1) it is shown how a set of keywords is calculated for a paper (KS^i):

$$KS^i = \left\{ k_1^i \quad k_2^i \quad \dots \quad k_j^i \quad \dots \quad k_m^i \right\}, i = \overline{1:n} \quad (1)$$

where:

n – the number of the analyzed documents;

i – current number of the analyzed paper;

m – the number of terms used for the analysis;

k_j^i – the term j with the highest frequency of occurrence in document i.

The terms from the KS^i set are used for the automatic analysis of vulnerabilities that are related to these terms.

Accordingly, for each paper it is obtained a set similar with the KS^i and for the terms from this set the vulnerabilities identified on CVE platform are analyzed.

Vulnerabilities are then identified for each term and the average Impact Score (*IS*) is calculated for all vulnerabilities of a given key term. Thus, for the KS^i set containing the key terms identified in document *i* the set IS_{KS^i} is obtained:

$$IS_{KS^i} = \left\{ MIS_{k_1^i} \quad MIS_{k_2^i} \quad \dots \quad MIS_{k_j^i} \quad \dots \quad MIS_{k_m^i} \right\}, i = \overline{1:n} \quad (2)$$

where $MIS_{k_j^i}$ is the average value of the impact score obtained for the vulnerabilities identified for the term k_j^i. That means that if for a term only one vulnerability was identified the $MIS_{k_1^i}$ will be exactly the impact score of that vulnerability. For those terms for which more vulnerabilities were identified, an average was calculated. The Impact Score for each vulnerability is calculated with the newer versions of the CVSS calculator (version 2 or version 3). In this way two values will be obtained for this indicator: DIS^2 and DIS^3. In this analysis only the DIS^3 will be used.

CVSS Version 2 (CVSSv2) has been launched in 2007 and CVSS version 3 (CVSSv3) has been released in 2015. According to (Risk Based Security, 2017), the most important change was that the environmental metrics in version 2 were replaced with a Modified Base Score. Essentially, each of the Base metrics may be modified by a certain organization so as to reflect differences between its situation and environment vs those of others.

For this set, the Document Impact Score (*DIS*) is proposed as the average vulnerability score for the document *i* that it is calculated based on the following formula:

$$DIS_i = \frac{\sum_{j=0}^{m} MIS_{k_j^i}}{m} \quad (3)$$

Based on this value, trends will be determined regarding the publication of scientific papers and the correlation between these materials and the existing security.

In parallel with this analysis, the proposed solution analyzes the words in the title of scientific papers and its correlation with security concepts. As such, a new indicator will be introduced: Title Security Keywords Score (TSKS). In order to calculate this indicator, a list of security-related keywords was made with each keyword being assigned a score. The platform analyzes all the words from the title and gathers the number of points obtained for all the keywords identified in that title. The security-related keywords and the associated scores are presented in Table 1.

Table 1. Security-related keywords and their importance

Keyword	Points	Keyword	Points
security	100	secure	100
vulnerability	90	vulnerabilities	90
risk	70	hack	70
attack	70	exploit	50
threat	80	breach	60
countermeasure	40	malware	40
critical	40	defend	30
malicious	40	firewall	40
virus	60	antivirus	60
protection	40	ransomware	70
spyware	60	phishing	60
trojan	60	worm	50
encryption	70	decryption	70
CVE	80	password	80
injection	80	privacy	30
botnet	30	denial	40
DDoS	60	rootkit	30
authentication	80	authorization	80
surveillance	30	identity	50
secrecy	60	secret	60

This list from Table 1 was built based on the lists of the most searched words using search engines like Google as well as on the words used in SEO (Search Engine Optimization) provided by Mondovo, Inc. (2020) and WordStream (2020).

The values obtained for the paper title will be correlated with the values previously obtained for the paper content. This will show the correlation between the paper title and the content in terms of security and vulnerabilities identified for the key terms in the text.

For each article a report will be generated with the following content: article title, link to the PDF file, five keywords identified for the analysis of existing vulnerabilities, the number of vulnerabilities identified for each keyword, and the Average Base Score (*AverageBaseScore*) for vulnerabilities identified for each keyword and for the whole paper (based on the five terms).

One uses both versions of the CVSS computer to calculate the average base score. These values represent DIS^2 and DIS^3, and they are calculated according to equation (3). Also, the keywords in the title of the paper according to Table 1 were determined, along with the scores obtained automatically for TSKS indicator.

3.2 Datasets

This analysis can be performed for any type of documents, such as: Doctoral theses, Master theses, Bachelor theses, articles published in journals etc. For this analysis the articles published in a scientific journal were chosen and the evolution of publishing trends in the field of information security was shown.

In order to validate the proposed model all papers published in "Informatica Economică" journal (http://revistaie.ase.ro) between 2010 and 2020 were analyzed. According to its website, "Informatica Economică" is an open access journal that covers various topics regarding the research, practice, and education in economic informatics field, like: digital economy, applied informatics in economy, ICT security, information and computer-based systems, education and research in economic informatics, qualitative and quantitative models applied in computer science. The first issue of "Informatica Economică" journal was published in 1997. The papers were written in Romanian until the last issue of 2006. Starting with 2007, the journal language is English. "Informatica Economică" journal is published four times a year.

The analyzed period is of 11 years (2010 – 2020) and it includes 42 issues (2 issues from current year) with an average of 11 papers per issue.

3.3 Solution Architecture

The proposed solution is Web-based with a RESTFul Web service that allows to follow the flow from the REST API to the front-end and so,

to the user. This flow is presented in Figure 1. The application architecture is based on Model View Controller (MVC) architectural design pattern.

The server acts as a REST API and it also handles some of the workflow of the back-end, like handling the requests from the clients and responding to them, as it shown in Figure 2 through a Progressive Web Application (PWA). The RESTful web service aims to provide an independent data source for the client, by taking advantage of the HTTP/ HTTPS protocol. Most importantly, the main idea behind the back-end was that it must be a scalable solution that would allow the expansion of the functionalities within the application while being able to maintain and organize the source code easier.

The core components of the solution are represented by two Node.js scripts that have the purpose of fetching all the data regarding the common vulnerabilities and exposures provided by the NVD data feeds and of populating/updating the MongoDB database with them.

The modules implementation, which allows the user to see the latest vulnerabilities or search for specific vulnerabilities, was done so the application offers the required functionality which can be further expanded upon. The

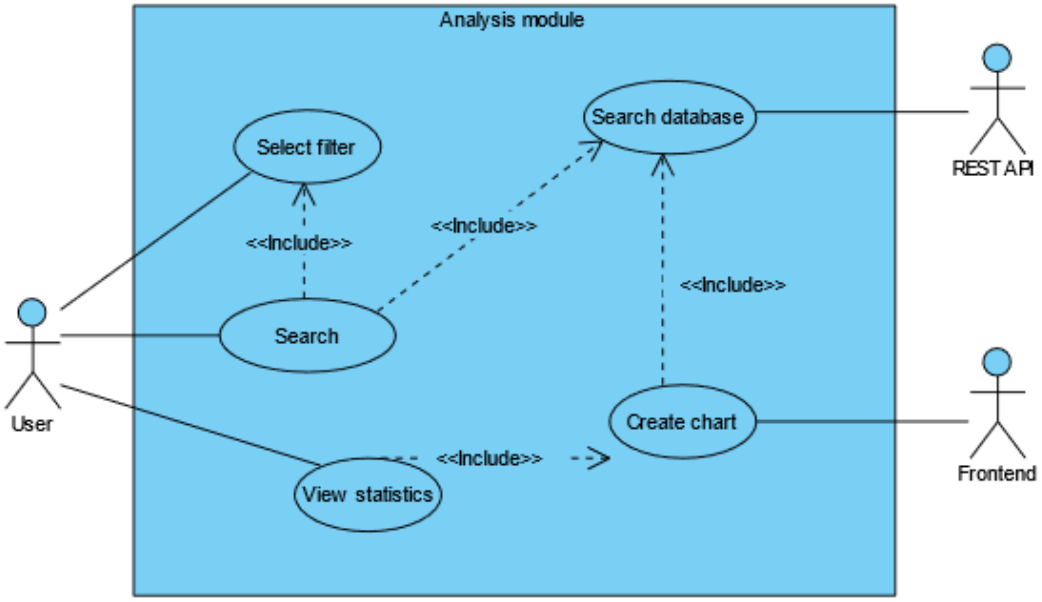

Figure 1. The application flow

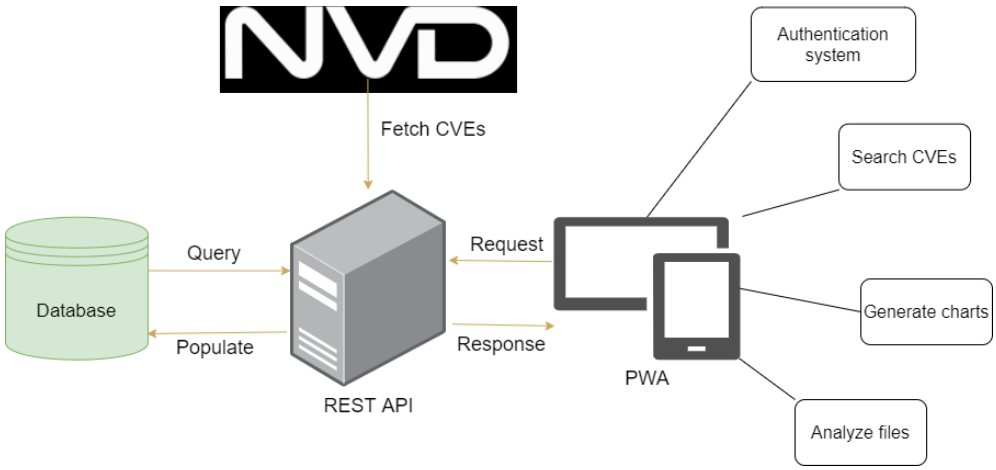

Figure 2. Solution architecture

Mongoose object data modeling (ODM) is used for creating complex queries to the MongoDB database in order to get the desired data, as it is illustrated in Figure 3.

```
const avgV3 = await Cve.aggregate([
  {
    $match: {
      $and: query,
      publishedDate: {
        $gte: new Date(body.startDate),
        $lte: new Date(body.endDate),
      },
    },
  },
  {
    $group: {
      _id: null,
      avg: { $avg: '$impact.baseMetricV3.cvssV3.baseScore' },
    },
  },
]);
```

Figure 3. Mongoose average base score query

Indeed, as it was suggested before, the aim of this solution is to offer an automated way of obtaining the results needed for the analysis of certain common disclosed vulnerabilities.

4. Results

4.1 Outcomes

A total of 470 articles published between 2010 and 2020 were analyzed and for each article several characteristics were stored (title, the link to the PDF file, calculated average scores etc.). Because of the limit of 5 keywords per paper, a total of 2,350 keywords were obtained. From among these keywords, only 284 terms are unique, all other 2,066 terms are duplicates of the initial ones. For example, the term "IT" has been encountered 250 times, being the term with the highest frequency of occurrence. This can be explained by the fact that the journal addresses the IT field in its topics. Figure 4 shows the keywords identified according to the frequency of their occurrence in scientific papers.

A clustering of the papers was made according to the year in which they were published. Table 2 includes the number of published papers

Table 2. Average Base score for all papers published by year

Year	Number of published research papers	Number of security-related research papers	V2	V3
2010	73	9	5.718904	7.084849
2011	70	7	5.750657	7.119714
2012	61	8	5.818951	7.169311
2013	54	7	5.791852	7.188481
2014	48	3	5.799583	7.166708
2015	27	1	5.849111	7.140222
2016	29	3	5.863586	7.189586
2017	29	2	5.906966	7.174
2018	33	1	5.898121	7.217515
2019	30	4	5.8634	7.2
2020	16	3	5.920125	7.18075

Figure 4. Keywords from the content of the research papers

in the journal, the number of security-related papers and the average score for all the articles written during the analyzed period. The number of published papers related to security domain is determined based on the number of keywords related to security from the title of these papers.

The number of published papers has decreased over time, and for 2020 there are only 16 papers, from the first two issues of the journal. The values obtained for Average Base Score for version 2 and version 3 of CVSS calculator are graphically represented in Figure 5.

There is a slightly increasing trend regards to the obtained score for the vulnerabilities identified for both versions of the CVSS calculator. This may be due to the fact that over time the authors have attached greater importance in the scientific papers to solutions and tools with vulnerabilities already identified by the community of researchers and specialists in this field.

Figure 6 shows that the number of published security-related research papers represents a small part of the total published research papers.

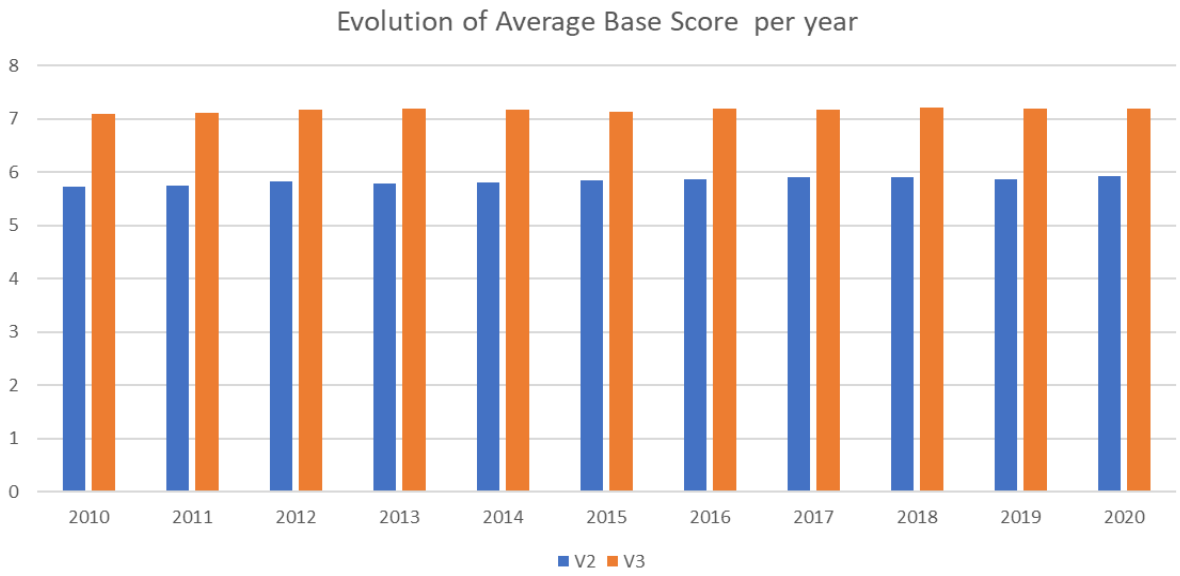

Figure 5. Average Base score for all papers published by year

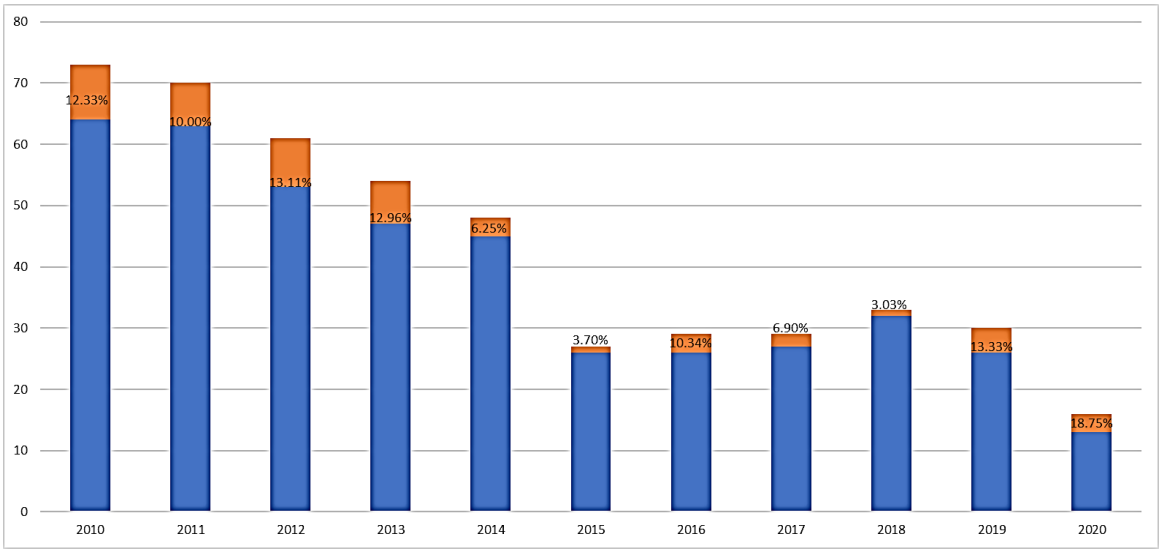

Figure 6. Percentage of published security-related papers

4.2 Discussion and Interpretation

Out of the 470 scientific papers analyzed, 48 contain security-related keywords according to Table 1. Obviously the most common term is "security", with a frequency of 23. The resulting representation, as cloud words, based on security-related keywords from the titles of scientific papers is shown in Figure 7.

An interesting analysis is given by the fact that some of the keywords found in the title of the research papers were not included in the analysis of the words contained in the text of the papers. These words have a low frequency in the text and were not included in the set of five terms analyzed for each scientific paper. These five keywords that are not in the first list are highlighted along with their frequencies in Table 3.

By using the points for each keyword from Table 1, the Title Security Keywords Score (TSKS) indicator is calculated or scores for the titles of research papers. These scores were compared with Document Impact Score (DIS) obtained for those papers. This analysis was performed only for those papers that contain keywords in the title. Because the values obtained by using version 3 of the CVSS computer are higher, these values were used in this analysis, and the values for the score obtained based on the analysis of the keywords in the title were normalized for the interval [0, 10].

Table 3. List of keywords from title that are not analyzed in the content of research papers

Keyword	Frequency in title	Analyzed in the content of papers	Frequency in the content
ransomware	1	NO	-
secure	4	NO	-
security	23	YES	40
risk	16	YES	24
vulnerabilities	1	NO	-
critical	1	NO	-
protection	1	YES	2
malicious	1	NO	-
identity	1	YES	2
phishing	1	YES	1
privacy	4	YES	2
threat	1	YES	2
vulnerability	1	YES	1

Figure 7. Keywords from the titles of research papers

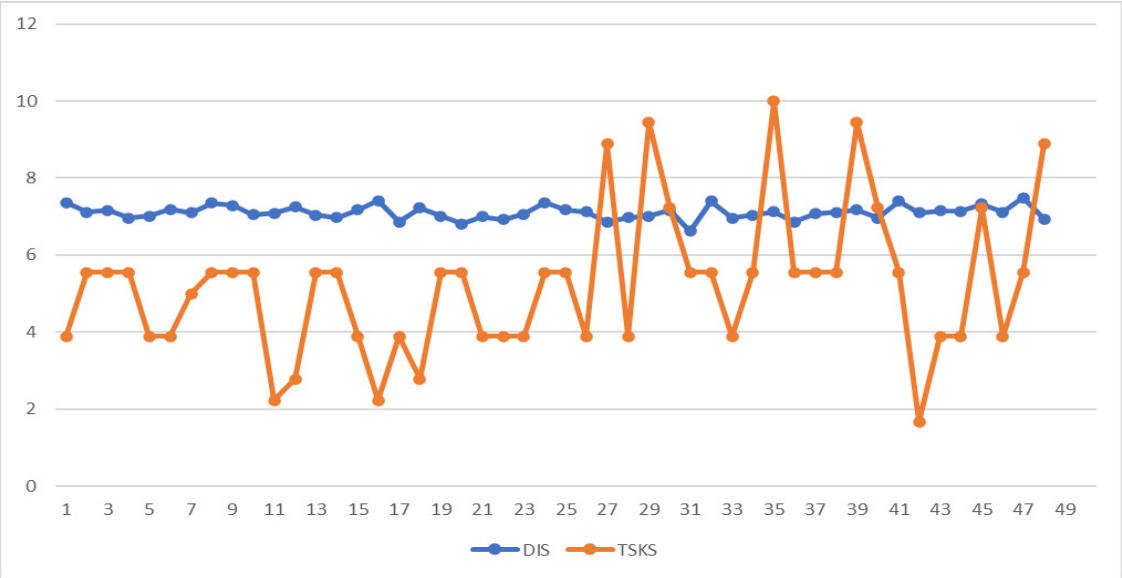

Figure 8. Correlation between DIS and TSKS

Figure 8 shows that the values obtained for DIS have a much smaller variation than the values obtained for TSKS. You can also see some peaks of the values obtained for TSKS. T hese peaks are also due to the fact that in the title of research papers were identified two key terms that contributed to obtaining TSKS score.

5. Conclusions and Future Work

This paper proposed several indicators and a web-based solution for the automated analysis of vulnerability-related topics in research papers and they were tested using published scientific papers from a selected journal. The results depict the status of this research in this field.

The results also show a high interest in research in the security-related fields and the desire to be up to date with the latest topics.

There are certain risks in the successful development of the application that supports this research, as many factors in the future could influence the overall results of the system. For example, many of the CNAs could simply stop identifying their vulnerabilities according to the international standard and making them public, therefore incapacitating the application to analyze more recent vulnerabilities. As another example, there may be legal issues because "CVE" acronym is a registered trademark of the MITRE which can result in the database becoming unusable by the general public.

Another future aim is to use this tool for other scientific papers in order to perform a comprehensive analysis of this important topic. Another objective is to improve the process of keyword selection and to automate this process.

The current version of the solution is results-oriented rather than user-friendly. A specific development objective is to create a publicly available platform.

Another objective would be to develop other indicators that could be employed in order to better describe the research in this important field of computer security.

Acknowledgements

This paper was co-financed through the Human Capital Operational Program 2014-2020, project number POCU/380/6/13/125245 no. 36482/23.05.2019 "Excellence in interdisciplinary PhD and post-PhD research, career alternatives through entrepreneurial initiative (EXCIA)", coordinated by the Bucharest University of Economic Studies".

REFERENCES

Allodi, L., Cremonini, M., Massacci, F. & Shim, W. (2020). Measuring the accuracy of software vulnerability assessments: experiments with students and professionals, *Empirical Software Engineering*, *25*(2), 1063-1094.

Angelini, M., Blasilli, G., Catarci, T., Lenti, S. & Santucci, G. (2018). Vulnus: Visual vulnerability analysis for network security, *IEEE Transactions on Visualization and Computer Graphics*, 25(1), 183-192.

Boja, C. E., Herţeliu, C., Dârdală, M. & Ileanu, B. V. (2018). Day of the week submission effect for accepted papers in Physica A, PLOS ONE, Nature and Cell, *Scientometrics*, *117*(2), 887-918.

Chang, Y. Y., Zavarsky, P., Ruhl, R. & Lindskog, D. (2011). Trend analysis of the CVE for software vulnerability management. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing* (pp. 1290-1293).

Elbaz, C., Rilling, L. & Morin, C. (2019). Towards Automated Risk Analysis of "One-day" Vulnerabilities. In *RESSI 2019 - Rendez-vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information*, Erquy, France (pp. 1-3).

FIRST, Inc. (2019). *CVSS v3.1 Specification Document – Revision 1*. Available at: <https://www.first.org/cvss/v3.1/specification-document>.

Glanz, L., Schmidt, S., Wollny, S. & Hermann, B. (2015). A vulnerability's lifetime: enhancing version information in CVE databases. In *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business* (pp. 1-4).

Guzzo, A., Ianni, M., Pugliese, A. & Saccà, D. (2020). Modeling and efficiently detecting security-critical sequences of actions, *Future Generation Computer Systems*, *113*, 196-206.

Johnson, P., Lagerström, R., Ekstedt, M. & Franke, U. (2016). Can the common vulnerability scoring system be trusted? a Bayesian analysis, *IEEE Transactions on Dependable and Secure Computing*, *15*(6), 1002-1015.

Li, Y., Xu, Z., Wang, X. & Filip, F. G. (2020). Studies in Informatics and Control: A Bibliometric Analysis from 2008 to 2019, *International Journal of Computers Communications & Control*, *14*(6), 633-652.

Mondovo, Inc. (2020). *The Most Searched Internet Security Keywords in Google*. Available at: <https://www.mondovo.com/keywords/internet-security-keywords>.

Pham, V. & Dang, T. (2018). CVExplorer: Multidimensional visualization for common vulnerabilities and exposures. In *2018 IEEE International Conference on Big Data (Big Data)* (pp. 1296-1301).

Risk Based Security (2017). *CVSSv3: When Every Vulnerability Appears to Be High Priority*. Available at: <https://www.riskbasedsecurity.com/2017/05/02/cvssv3-when-every-vulnerability-appears-to-be-high-priority/>.

Ruohonen, J. (2019). A look at the time delays in CVSS vulnerability scoring, *Applied Computing and Informatics*, *15*(2), 129-135.

Sandu, I. E., Smada, D. M. & Dumitrache, M. (2019). An affordable Web-based Grant Management Software Designed to Support Romanian Scholarly Publications, *Studies in Informatics and Control*, *28*(1), 95-103. DOI: 10.24846/v28i1y201910

Sauerwein, C., Pekaric, I., Felderer, M. & Breu, R. (2019). An analysis and classification of public information security data sources used in research and practice, *Computers & Security*, *82*, 140-155.

Tan, T., Wang, B., Tang, Y., Zhou, X. & Han, J. (2019). ICVSS: A New Method for Vulnerability Quantitative Grading, *CMC: Computers, Materials & Continua*, *61*(2), 629-641.

Wang, W., Shi, F., Zhang, M., Xu, C. & Zheng, J. (2020). A Vulnerability Risk Assessment Method Based on Heterogeneous Information Network, *IEEE Access*, *8*, 148315-148330.

Williams, M. A., Barranco, R. C., Naim, S. M., Dey, S., Hossain, M. S. & Akbar, M. (2020). A vulnerability analysis and prediction framework, *Computers & Security*, *92*, 101751.

WordStream (2020). *Internet Security Keywords*. Available at: <https://www.wordstream.com/popular-keywords/internet-security-keywords>.

Wu, X., Zheng, W., Chen, X., Wang, F. & Mu, D. (2020). CVE-assisted large-scale security bug report dataset construction method, *Journal of Systems and Software*, *160*, 110456. DOI: 10.1016/j.jss.2019.110456

Zamfiroiu, A., Pocatilu, P. & Capisizu, S. (2020). CrawVulns - A Software Solution for Vulnerabilities Analysis, *Informatica Economica*, *24*(1), 38-47.