Topic Modeling And Classification Of Common Vulnerabilities And Exposures Database

MounikaVanamala
Department of Computer science
North Carolina A&T State University
Greensboro,NC USA
vanamala.mounika7@gmail.com

Xiaohong Yuan
Department of Computer science
North Carolina A&T State University
Greensboro,NC USA
xhyuan@ncat.edu

Kaushik Roy
Department of Computer science
North Carolina A&T State University
Greensboro,NC USA
kroy@ncat.edu

***Abstract*—The Common Vulnerabilities and Exposures (CVEs) is a repository of publicly known cybersecurity vulnerabilities. This repository can be used in many ways including vulnerability assessment, intrusion detection, security information management, etc. Mining this repository also help us understand the trend of vulnerabilities and exposures. We have analyzed CVE entries from 2009-2019 in terms of vulnerability types defined by OWASP Top-10 risks. We extracted 121,716 unique CVEs through the twenty years. In this paper we have implemented a method to categorize these CVE entries to OWASP Top-10 risks using LDA Topic Modelling and Keyword matching technique. This method enables automatic analysis of CVE with large number of entries and contributes to the creation of vulnerability taxonomy from CVE.**

***Keywords*— Latent Dirichlet Allocation (LDA), Topic Classification, Rule Based Systems, Vulnerability analysis, document classification.**

I. Introduction

Mining software repositories is a research area which aims to understand the data repositories related to software enhancement and make intelligent use of these software repositories to support the decision-making process of software development. Software repositories include databases for bug tracking, code change management, code review, building system, releasing binaries, forums, and more to mention. There have been software repositories that support software security, i.e., the process of developing software that are robust to malicious attacks.

For example, a vulnerability database is a software repository aimed at collecting, disseminating, and maintaining information about discovered computer security weaknesses. There are different vulnerability databases like the U.S. government repository National Vulnerability Database (NVD) [1], Common Weakness Enumeration (CWE) [2], and Common Vulnerabilities and Exposures database (CVE) [3]. CVE includes information about publicly known cybersecurity vulnerabilities software engineers can benefit from.

In this paper, we present a method for analyzing the CVE database using topic modeling. Topic models take as input a set of text documents (the corpus) and return a set of topics that can be used to describe each document in the corpus [4].The Latent Dirichlet Allocation (LDA) [5] topic modeling technique has been applied to examine data provided in the CVE and identify the prevalent topics in the CVE. These topics were then mapped to The Open Web Application Security Project (OWASP) Top-10 Risks [6] using keyword based matching technique. The OWASP [6] is a non-profit organization dedicated to providing reliable knowledge about protection of applications. In 2017 the OWASP Top 10 Web Application Security Risks was revised to provide developers and security practitioners with advice on the most important vulnerabilities frequently found in web applications.

In a previous work [7], we have analyzed the 121,716 CVE entries from 2009 until the end of June 2019 using topic modeling and mapped the topics into OWASP top-10 risks manually. We have developed some rules based on the understanding of a human expert on the topics generated by topic modeling and the OWASP top 10 risks. Based on the description of OWASP top 10 risks, and OWASP Application Security Verification Standard, we have prepared a list of keywords that corresponds to a vulnerability type in OWASP top 10 which represent these rules. In this work, we used pattern matching technique to automatically map the topics produced by topic modelling to OWASP top 10 vulnerability categories. Results from the automatic mapping are compared with the results of manual mapping, and it is shown that automatic mapping results are very similar to manual mapping results. Therefore, the topic modeling and automatic mapping method can be used to analyze large CVE data sets. The method implemented in this paper can contribute to the creation of vulnerability taxonomy from CVE.

The rest of the paper is organized as follows: First, topic modelling and LDA are introduced in Section II. Then in section III the methodology used for Topic Modeling and Classification of CVE Database is presented and the results are discussed. Section IV discusses the related work and Section V concludes the paper.

II. TOPIC MODELLING

When it comes to analyzing huge amount of text data, it's too big a task to do it manually. It's also tedious, time-consuming, and expensive. Manually sorting through large

978-1-7281-6770-1/20/$31.00 ©2020 IEEE

amount of data is also more likely to lead to mistakes and inconsistencies. Topic analysis models enable a user to sift through large sets of data and identify the most frequent topics in a very simple, fast and scalable way.

Topic modeling is an unsupervised machine learning technique for analyzing large volumes of text data by clustering the documents into groups of topics [8]. It has been demonstrated to be highly effective in a wide range of tasks, including multi-document summarization, organizing large blocks of textual data, word sense discrimination, sentiment analysis, information retrieval [9], image labelling, and feature selection.

There are many approaches for obtaining topics from a text such as – Term Frequency and Inverse Document Frequency, Nonnegative Matrix Factorization techniques [5]. LDA is a content analysis technique designed to automatically arrange large document collections based on latent topics, calculated as word patterns (co-)occurrence. LDA assumes documents are produced from a mixture of topics [4]. Those topics then generate words based on their probability distribution. Given a dataset of documents, LDA backtracks and tries to figure out what topics would create those documents in the first place. The purpose of LDA is to map each document in the corpus to a set of topics which covers a good deal of the words in the document. The LDA considers each document as a set of topics. And every topic, as a set of keywords [10]. If the number of topics is given, it rearranges the distribution of topics within the documents and the distribution of keywords within the topics to obtain a composition of the distribution of topic and keywords. LDA is a popular algorithm for topic modeling with excellent implementations in the Python's Gensim package [11]. The challenge with topic modeling, is how to extract good quality topics that are clear, segregated and meaningful. This depends heavily on the quality of text preprocessing and the strategy of finding the optimal number of topics [11].

III. TOPIC MODELING AND CLASSIFICATION OF CVE DATABASE

Fig. 1 shows the flowchart for topic modeling and classification for CVE database. The process includes three phases which are explained below.

Phase-1: Data Preprcessing

First we need to remove stopwords. To remove stopwords, we need the stopwords from Natural Language Toolkit (NLTK) and spacy's en model for text pre-processing. We used the spacy model for lemmatization which converts a word to its root word. For example, the lemma of the word 'machines' is 'machine' . Likewise, the lemma of the word 'walking' is 'walk', the lemma of 'mice' is 'mouse' and so on. When data preprocessing is complete, regular expressions are used to erase emails and newline characters. After that, if the text still has special characters like empty spaces, it isn't ready for consumption by the LDA. Upon eliminating special characters, each sentence needs to be broken down into a list of words by tokenization, thus cleaning up all the punctuations and unnecessary characters in the text. The core packages used in this process are re, gensim, spacy and pyLDAvis. Besides this, we also used matplotlib, numpy and pandas for data handling and visualization [10]. Once the data cleaning is done, we move to building Bigram and Trigram models.

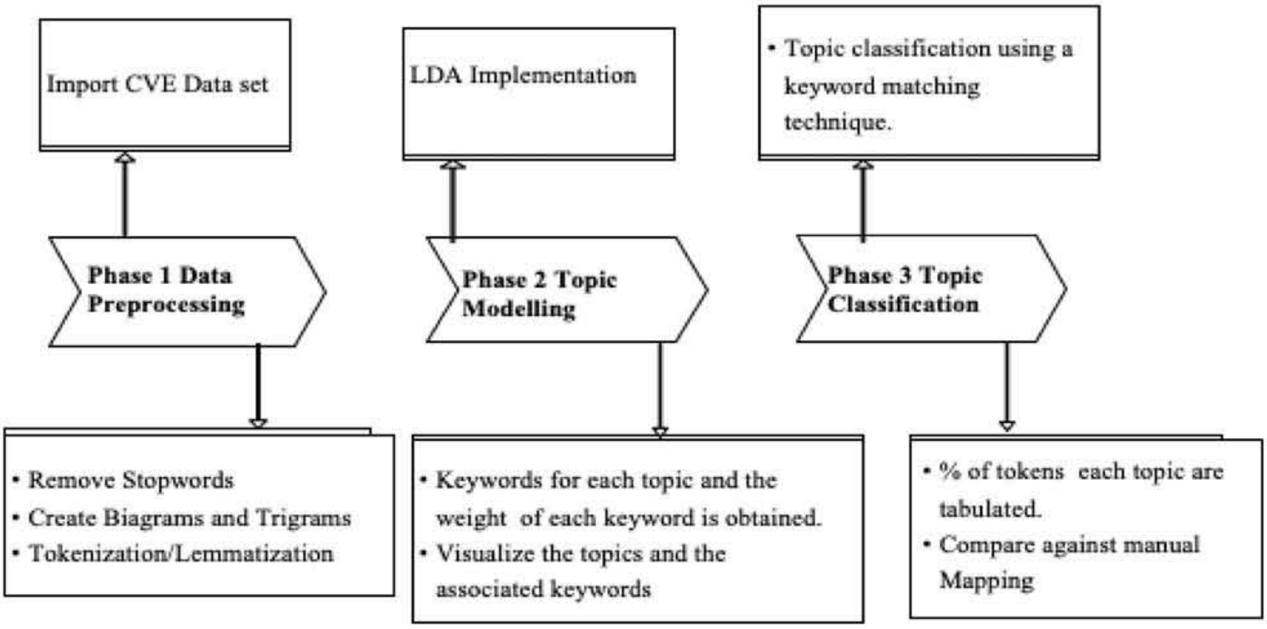
Import CVE Data set
LDA Implementation
• Topic classification using a keyword matching technique.
Phase 1 Data Preprocessing
Phase 2 Topic Modelling
Phase 3 Topic Classification
• Remove Stopwords
• Create Biagrams and Trigrams
• Tokenization/Lemmatization
• Keywords for each topic and the weight of each keyword is obtained.
• Visualize the topics and the associated keywords
• % of tokens each topic are tabulated.
• Compare against manual Mapping

Fig. 1. The Flowchart for Topic Modeling and Classification of CVE database

Phase-2: Building the Topic Model

The LDA topic model's two key inputs are the dictionary (id2word), and the corpus. The dictionary is a mapping of the terms with their integer ids. Besides the corpus and dictionary, the user also needs to give the number of topics. It will be used for LDA model preparation. After the LDA has been built we are able to view the topics in the LDA model. The LDA model was designed with 10 different topics for the CVE dataset, each topic being a combination of keywords and each keyword contributing to the topic a certain weighting. After the LDA model is built on the CVE dataset, we can use lda_model_print topics () to display the keywords for each topic and the weight (importance) of each keyword. Topic 0 is represented as **(0,'0.112*"unspecified" + 0.103*" vector " + 0.047*" relate " + 0.046*"possibly" + ' '0.045*"unknown" + 0.040*"application" + 0.037*"buffer" + 0.033*"code" + ' '0.027*"overflow" + 0.019*"base"').** It means the top 10 keywords that contribute to this topic are: **'unspecified', 'vector', 'relate', 'possibly', 'unknown', 'application', 'buffer', 'code', 'overflow', 'base',** the weightage of keyword **'unspecified'** on topic 0 is 0.112, and so on.The weights reflect how important a keyword is to that topic. Once the LDA is built we move on to Visualize the topics and the associated keywords. pyLDAvis package is an interactive visualization tool designed to work with Jupiter notebooks. Fig. 2 shows the output of pyLDAvis for data entries in CVE from 2016-2019. Each bubble represents a topic. The bigger the bubble, the more prevalent the topic is. A model with too many topics, will typically have many overlapping, small sized bubbles clustered in one region of the chart. In our case Topic 9 and 10 are overlapping which means these topics are very similar. When a bubble (topic) is selected (i.e., topic 1 in Fig. 2), the top 30 most relevant terms for that topic is shown, and the percentage of these 30 terms over all the tokens for this topic is given.

Phase-3 Rule-based Topic Classification

It's possible to build a topic classifier entirely by hand, without using machine learning methods. This is by directly programming a set of man-made rules, based on the content of the documents that a human expert actually read. The idea is that the rules represent the codified knowledge of the expert and are able to discern between documents of different topics by looking directly at semantically relevant elements of a text, and at the metadata that a document may have. Each one of these rules consists of a pattern and a prediction [12].

Based on the definition of OWASP top 10 vulnerabilities, and the OWASP Application Security Verification Standard, a list of keywords corresponding to a vulnerability has been identified which correspond to the rules for classifying topics in CVE generated through topic modeling to OWASP top 10 risks [6]. Some of the OWASP top 10 Vulnerability type and the keywords associated with them are listed below:

- A1: 2017-Injection:SQL, Injection, code_injection, Malicious, Statements, Escape characters, Query.
- A2:2017-Broken Authentication Allow: Attacker, unexpired, session, tokens, login, Credentials, Authenticate, timeout, Passwords, hashed, bypass, credentials, brute forcing, autocomplete, two-factor authentication, API keys.re-authentication.
- A3:2017-Sensitive Data Exposure: local privacy laws, regulations, organizational sensitive data, risk assessment, user's session identifiers, passwords, hashes, or API tokens, data, protection directives, URL parameters.

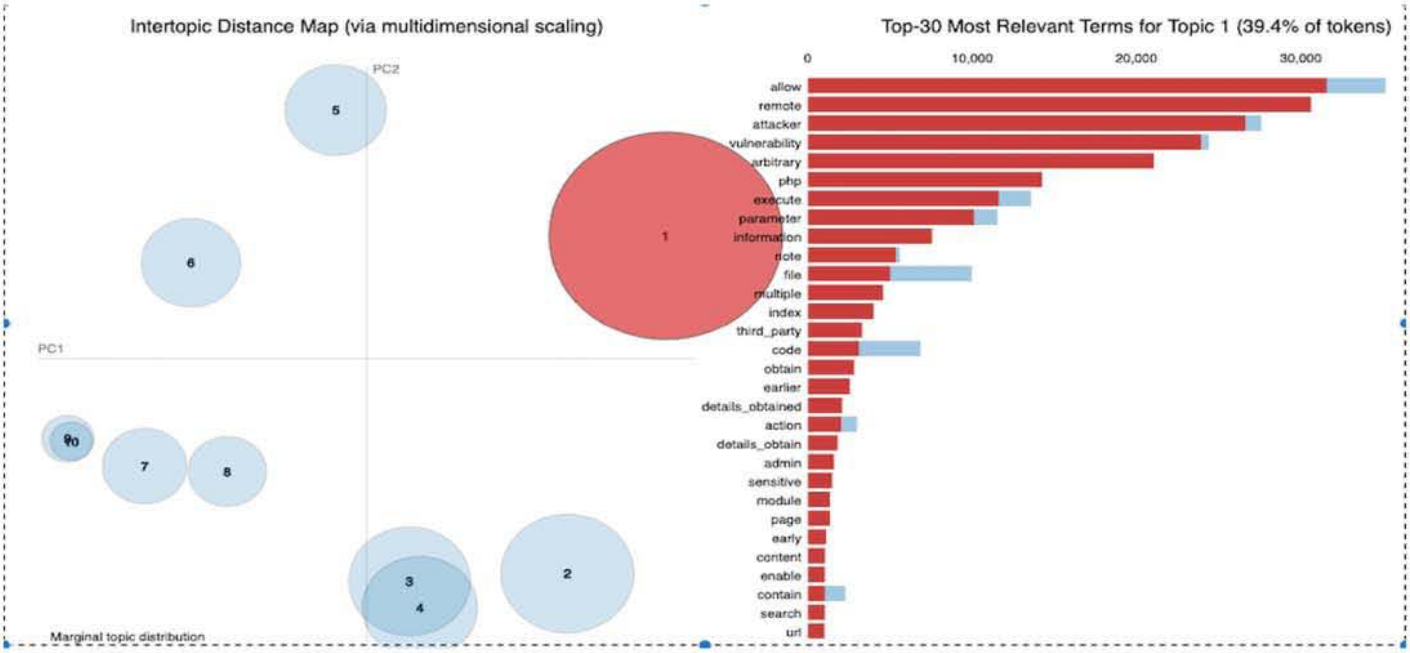

Fig. 2. The output of pyLDAvis for years 2016-2019

Based on the keywords for each OWASP top 10 vulnerability type, we mapped the topics produced by topic modeling to OWASP top 10 vulnerabilities using keyword matching technique. From Phase-2, we have 10 topics and each topic has the set of keywords obtained from LDA. We get the list of keywords corresponding to each topic, put the keywords into a list. In the same manner we have split the keywords for OWASP vulnerability types by separating them with a delimiter and removing the stop words. Then we compared the two lists. We compare the keywords from every topic to keywords for each OWASP top 10 vulnerability type. We classify a topic that has matched maximum number of keywords of an OWASP top 10 vulnerability to that OWASP vulnerability. The detailed steps applied for the keyword matching technique is described below:

Step-1: From Phase-1 and 2 described above we obtained the top 30 relevant terms/keywords for every topic as shown in Fig.2. We created a dictionary mapping topic to sets of relevant terms
Let d= {topic1: (set1 of terms), topic2: (set2 of terms), ..., topic *m*: (set *m* of terms)}

Step-2: Based on the definition of OWASP top 10 vulnerabilities, and the OWASP Application Security Verification Standard, a list of keywords corresponding to a vulnerability has been prepared. The complete mapping table is provided in [7].
Let k = {vulnerability1: (set1 of keywords), vulnerability2: (set2 of keywords), ..., vulnerability *n*: (set *n* of keywords.}

Step-3: Now we compare the set of terms for each topic in list d and the set of keywords for each vulnerability in list k. Once we get the set of keywords with highest intersection with the terms, we map the topic to the corresponding vulnerability. Let v = {topic 1: vulnerability i1, topic 2, vulnerability i2, …}

```
for topic in d:
  I = {};
  for vulnerability in k:
    if I < d[topic] ∩ k[vulnerability]
      I = d[topic] ∩ k[vulnerability]
      v[topic] --> vulnerability
```

TABLE-I COMPARISION OF MANUALLY MAPPED TOPIC AND AUTOMATICALLY MAPPED TO OWASP TOP-10

% of Tokens															
OWASP-TOP-10	**1999-2003**			**2003-2007**			**2007-2011**			**2011-2015**			**2016-2019**		
	Manual	Auto	CV	Manual	Auto	CV	Manual	Auto	CV	Manual	Auto	CV	Manual	Auto	CV
A1:2017-Injection	4.4	3.2	0.16	7	5.8	0.09	7.5	6.2	0.09	5	5.7	0.07	5.4	6.2	0.07
A2:2017-Broken Authentication	11.3	10.8	0.02	16.5	15.2	0.04	20.5	21.4	0.02	26.9	25.1	0.03	27.6	27.6	0.00
A3:2017-Sensitive Data Exposure	6.5	7.8	0.09	7	7.5	0.03	5.1	6	0.08	5.2	6	0.07	7.4	7.8	0.03
A4:2017-XML External Entities (XXE)	0	1.8	1.00	2.3	2.5	0.04	6.6	7.2	0.04	9.7	10.2	0.03	10	11.2	0.06
A5:2017-Broken Access Control	27.4	23.5	0.08	26.9	23.5	0.07	33.2	31	0.03	35.7	36	0.00	39.4	39.2	0.00
A6:2017-Security Misconfiguration	5.9	4.3	0.16	0.3	1.6	0.68	5.1	4.9	0.02	0	3.2	1.00	5.2	4.9	0.03
A7:2017-Cross-Site Scripting (XSS)	36.6	37	0.01	38.9	37.7	0.02	7.1	8.5	0.09	6.6	6.7	0.01	5	5.4	0.03
A8:2017-Insecure Deserialization	0	2.5	1.00	4	3.4	0.08	3.3	4.3	0.13	3.3	2.5	0.14	0	0	0.00
A9:2017-Using Components with Known Vulnerabilities	3.9	4.1	0.03	0	0	0.00	5.8	4.5	0.00	2.7	2.2	0.00	0	0	0.00
A10:2017-Insufficient Logging and Monitoring	4	5	0.11	3.1	2.8	0.05	5.8	6	0.02	4.8	2.4	0.33	0	0	0.00

In our previous work we have manually mapped the topics produced by topic modeling to OWASP top 10 vulnerabilities [7].Using the output generated by pyLDAvis shown in Fig. 2 we tabulated how many % of tokens are in each vulnerability type for the span of every 5 years from 1999-2019 shown in Table-I. From Fig. 2 we can see that 39.4% of the total % of tokens for 2016-2019 years contribute to Vulnerability A5. For another example, topic-3 has 13.4% and topic – 4 has 14.2% of the total % of tokens from the years 2016-2019. As these topics are overlapping, sum of both the % of tokens i.e. 27.6% topics is mapped to A2:2017-Broken Authentication. The same procedure is followed for all the years and tabulated in Table-I.

In Table-I a comparison of the manual mapping and automated mapping results are shown. Standard deviation/mean, or coefficient of variance (CV) is also calculated and shown in Table I. If CV is <1, then the data is said to have low variance, i.e., the values are close to each other. From the TABLE-I we can observe that in the data set all the CV values are <=1. The average CV values for the 5 sets of CVs are 0.27, 0.11, 0.05, 0.27, 0.02. Therefore, the automatic mapping results are very similar to manual mapping results.

IV . RELATED WORK

Earlier work regarding vulnerability assessment was proposed by Genge and Enăchescu [13] in which they presented a vulnerability assessment tool for Internet-facing devices identified by Shodan [14]. This tool matched CVE entries to the corresponding devices without any processing of entries. Our work group the CVE entries into 10 topics and map the topics to OWASP top 10 security risks. Chang et al. [15] analyzed vulnerability trends using CVE entries from 2007 to 2010. They showed the vulnerability trends through vulnerability frequency and severity by using the CVE and CVSS scores, respectively. In our work we use Topic modelling to find different vulnerabilities and use keyword matching technique to categorize these topics into OWASP vulnerabilities. Neuhaus and Zimmermann [16] used topic models to analyze vulnerability trends, such as vulnerability types of CVE entries until 2009. The authors found 28 topics in CVE entries and mapped these topics to CWEs. Our approach is similar to this work in that we also used topic modeling to analyze CVE entries. However, we analyzed the topics in CVE data from 2009 to 2019 and mapped the topics to OWASPs top 10 instead of CWE.

V. CONCLUSION AND FUTURE WORK

Topic modeling is one of the most sought-after research areas in NLP. It is used to group large volumes of unlabeled text data. In this paper we built a topic model of CVE entries using Gensim's LDA and classified the topics to OWASP top 10 vulnerabilities. We have compared the results of the manual mapping to mapping automatically using keyword matches techniques. Our results show that automatic mapping results are very similar to manual mapping results. This implies that automatic mapping can be used to replace manual mapping for analyzing large CVE data sets. The mapping of CVE entries to OWASP top 10 vulnerabilities can contribute to the creation of vulnerability taxonomy from CVE.

Our future work includes comparing keyword matching algorithms with learning-based methods for classification of CVE database. The future goal is to develop a framework to automate the process of mapping the vulnerability reports or penetration test reports to security standards using machine learning techniques.

ACKNOWLEDGMENTS

This work is partially supported by NSF under the grant CNS-1900187. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF.

REFERENCES

[1] National Vulnerability Database, https://nvd.nist.gov/

[2] CWE, Common Weakness Enumeration: *URL:* https://cwe.mitre.org/

[3] Common Vulnerabilities and Exposures (CVE), http://www.cve.mitre.org/.

[4] Zhao, D., He, J., & Liu, J. (2014, April). An improved LDA algorithm for text classification. In *2014 International Conference on Information Science, Electronics and Electrical Engineering* (Vol. 1, pp. 217-221). IEEE.

[5] Li, W., Sun, L., & Zhang, D. K. (2008). Text classification based on labeled-LDA model. *CHINESE JOURNAL OF COMPUTERS-CHINESE EDITION-*, *31*(4), 620

[6] OWASP, T. (2017). Top 10-2017 The Ten Most Critical Web Application Security Risks. *URL: owasp. org/images/7/72/OWASP_Top_10-2017_% 28en, 29.*

[7] Mounika, V., Xiaohong, Y., & Kanishka, B. (2019, December). Analyzing CVE Database Using Unsupervised Topic Modelling. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI).*IEEE.

[8] Rehurek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.*

[9] Gopalakrishna, R., & Spafford, E. H. (2005). A trend analysis of vulnerabilities. *West Lafayette: Purdue University*, *13*.

[10] Newman, D., Asuncion, A., Smyth, P., & Welling, M. (2009). Distributed algorithms for topic models. *Journal of Machine Learning Research*, *10*(Aug), 1801-1828.

[11] Srinivasa-Desikan, B. (2018). Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras. Packt Publishing Ltd.

[12] Heffley, J., & Meunier, P. (2004, January). Can source code auditing software identify common vulnerabilities and be used to evaluate software security?. In *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the* (pp. 10-pp). IEEE.

[13] Genge, B., & Enăchescu, C. (2016). ShoVAT: Shodan-based vulnerability assessment tool for Internet-facing services. *Security and communication networks*, *9*(15), 2696-2714.

[14] Shodan, https://www.shodan.io/.

[15] Chang, Y. Y., Zavarsky, P., Ruhl, R., & Lindskog, D. (2011, October). 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). IEEE.

[16] Neuhaus, S., & Zimmermann, T. (2010, November). Security trend analysis with cve topic models. In *2010 IEEE 21st International Symposium on Software Reliability Engineering* (pp. 111-120). IEEE.