An Analysis of Common Vulnerabilities and Exposures in View Of MITRE ATT&CK

Gurinder Pal Singh
Research Scholar, Computer Science and Engineering
Chandigarh University
Gharauan, Punjab
gpshpr@gmail.com

Vishal Bharti
Additional Director, Computer Science and Engineering
Chandigarh University
Gharauan, Punjab
ad.cse@cumail.in

Manish Kumar Hooda
Sci/Eng-'SF', Head-Technology Development Division
Semiconductor Laboratory
S.A.S.Nagar, Punjab
manishk@scl.gov.in

***Abstract*— Due to the ever-increasing threat posed by cyber-attacks on important cyber infrastructure, companies are focusing on expanding the knowledge base on cyber security. The Universal Vulnerabilities and Exposures (CVE), that were a selection of vulnerabilities known as the Common Vulnerabilities and Exposures that may be discovered in a wide variety of applications and hardware and which are the most commonly exploited, are the most important things to know about security. They are troublesome, though, because many vulnerabilities do not have a mechanism of dealing with them, making it hard for an attacker to take use of them. ATT&CK, a well-known cyber security risk management methodology, provides mitigation solutions for a wide range of destructive tactics, according to the MITRE Corporation. In the National Vulnerability Database (NVD), there is a collection of security defects that have been publicly revealed, which is referred to as Common Vulnerabilities and Exposures (CVEs) (CVE). In this case various figure of CVE listings, however a few of missing crucial data, like as the type of vulnerability. during this article, our techniques for used Common Vulnerabilities and Exposures data interested in weakness classes by employing a naive Bayes classifier to categories the entries. To assess the classification capabilities of the approach, a set of testing data is gathered and analyzed.**

***Keywords*— CVE, National Vulnerability Database, Cyber security, Cyber Crime, MITRE ATT&CK**

I. Introduction

Defense weaknesses innate within programming bundles preserve be handily taken advantage of for directing malignant controls. Aggressors can distinguish weak Web administrations by utilizing an Internet-wide filtering instrument and lead malevolent conduct [1] . Subsequently, security specialists should know about known weaknesses and have the option to rapidly adapt to dangers [2].

Just 57.6 percent of all CVE passages accommodate CWEs that detect various types of flaws (Figure 1). It is possible to determine which type of vulnerability is explained by a CVE [3] passage by referring to the weakness outline provided in each CVE part, and so insufficient data may be improved by using this approach. The outline text has been structuralized in a certain structure, but because the construction is not exactly the same, we want to convert this text into a structure that is more appropriate for the building. This will be accomplished through information preprocessing.

In this paper, we propose a systematic strategy for identifying the types of weaknesses highlighted in text reports, predicting the order of CVE passages. We gather CVE sections from NVD and create a weakness characterization model in view of gullible Bayes. By utilizing this strategy, we can arrange CVE sections into weakness classification, i.e., CWE.

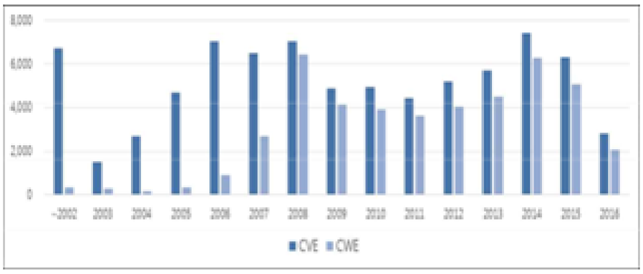


Fig. 1. Number of CVE entries and CWEs by year [3].

II. Literature Review

According to the [4] proposed a powerlessness examination strategy for Shodan-recognized Internet-associated gadgets. Minus any additional handling of the CVE passages, this program just paired them to the related gadgets. Chang et al. [5] analyzed weakness designs from 2007 to 2010 utilizing CVE sections.

They have shown vulnerabilities using the CVE and Common Vulnerability Assessment System (CVSS) ratings, respectively. This report differs from previous security trend analysis due to the addition of vulnerabilities found and reported in this year.

Neuhaus and Zimmermann [6] analyzed vulnerability patterns using topic models, such as the vulnerability classes associated with CVE submissions prior to 2009. The authors used Latent Dirichlet Allocation (LDA) to identify 28 subjects in CVE entries and allocated LDA topics to CWEsLDA has a high accuracy and review for some CWEs, for example, CWE-79 and CWE-89, however a low accuracy and review for other people, for example, CWE-310 and CWE-94.

Guo and Wang [7] made a philosophy based model of CVE weaknesses and used it to survey related weaknesses. We allude to the construction of CVE weaknesses to be utilized in this examination's order technique.

Li et al. [8] utilized text characterization and data recovery methods to evaluate the attributes of bugs and order them. In this exploration, we portray weaknesses utilizing a guileless Bayes classifier.

Introduced in 2016, the transformer model substitutes recurrent cells in well-known deep learning [9] models for text categorization (e.g., BiLSTM, LSTM) with multi-head attention mechanisms. While the first design included an encoder-decoder structure (for machine translation jobs), multi-class text categorization requires [10] simply the encoder stack.

The encoder transformer model constructs an embedding from the input, runs it through the transformer block, and produces a soft max probability score for the outputs. The embedding layer utilizes a one-hot encoding technique in conjunction with positional encodings. The transformer block is composed of a multi-head attention mechanism and feed-forward layers, and has been demonstrated to significantly enhance accuracy, [11] precision, recall, and F1-score in benchmark tasks when compared to recurrent models. Transformers have recently been utilized to generate huge Pre Training Language Models (PTLMs) that achieve state-of-the-art performance on a variety of text categorization tasks.

These models (for example, BERT, GPT-2) are frequently trained on millions of data points and have hundreds of millions of trainable parameters. While the majority of researchers do not own the [12] hardware or data used to generate their PTLM, these models may be fine-tuned and distilled for enhanced performance on particular tasks. Knowledge distillation is a relatively new concept for extracting critical information from a PTLMs parameters in order to augment the training of a specific model.

III. TECHNIQUES

We present vulnerability classification techniques based on the summary of CVE input. The conceptual map of our technique is depicted in Figure 2. We scratch NVD's CVE xml documents and parse each CVE section.

Following that, we do preprocessing to kill pointless terms from the chose outline, for example, stop words and programming item names, to build the characterization model's exactness. At long last, we make a model for weakness classification and arrange CVE sections.

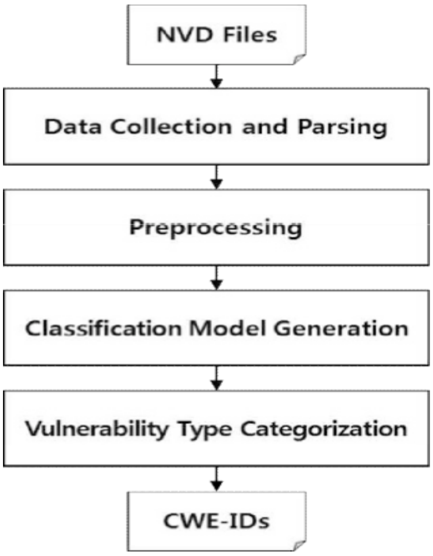

NVD Files
Data Collection and Parsing
Preprocessing
Classification Model Generation
Vulnerability Type Categorization
CWE-IDs

Fig. 2. hypothetical point of the proposed strategy.

A. Collection of the general Idea content

As found in Figure 3, a CVE passage contains of a recognizing, As a general rule, the rundown text contains the accompanying.

(whenever) 'states of the weakness event'.

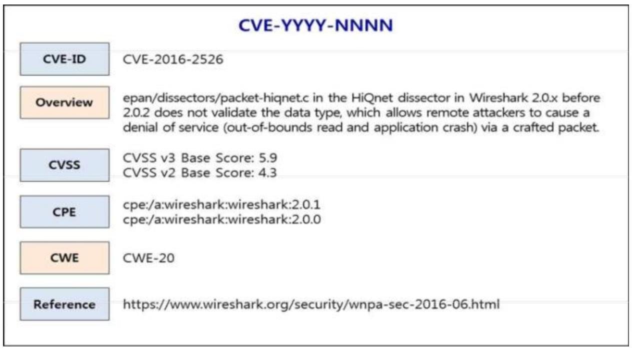
(permit) 'assailant type'.

(to) 'consequences of assault'.

(through) 'method for assault'.

(also known as) 'weakness title in the reference site'.

(an unexpected weakness in comparison to) 'other CVE-I.



CVE-YYYY-NNNN

CVE-ID	CVE-2016-2526
Overview	epan/dissectors/packet-hiqnet.c in the HiQnet dissector in Wireshark 2.0.x before 2.0.2 does not validate the data type, which allows remote attackers to cause a denial of service (out-of-bounds read and application crash) via a crafted packet.
CVSS	CVSS v3 Base Score: 5.9 CVSS v2 Base Score: 4.3
CPE	cpe:/a:wireshark:wireshark:2.0.1 cpe:/a:wireshark:wireshark:2.0.0
CWE	CWE-20
Reference	https://www.wireshark.org/security/wnpa-sec-2016-06.html

Fig. 3. Instance of a CVE access [3]

We use a piece of the outline text to portray the 'assault results,' 'assault strategy,' and 'weakness title on the reference site,' which may all be utilized to recognize weakness sorts.

In order to do this, the character string after the term 'allow to' is extracted and separated before the phrase 'a different vulnerability than'. The character string "generate a denial of service (out-of-bounds read and application crash) using a contrived packet." is used in the example shown in Figure 3. Unless otherwise noted, all phrases in the altered texts are capitalized.

Following that, we delete some terms regardless of vulnerability categorization, including stop words such as 'because' and 'with,' as well as product-related information. This step allows for the elimination of frequent terms that are irrelevant to the vulnerability category.

B. Making a Model for categorising vulnerabilities

The preparation/testing dataset comprises of the weakness synopsis text and the distinguishing proof for the weakness type (CWE-ID). We make a weakness grouping model utilizing revealed CVE sections and evaluate it utilizing extra CVE passages that were not used in the arrangement.

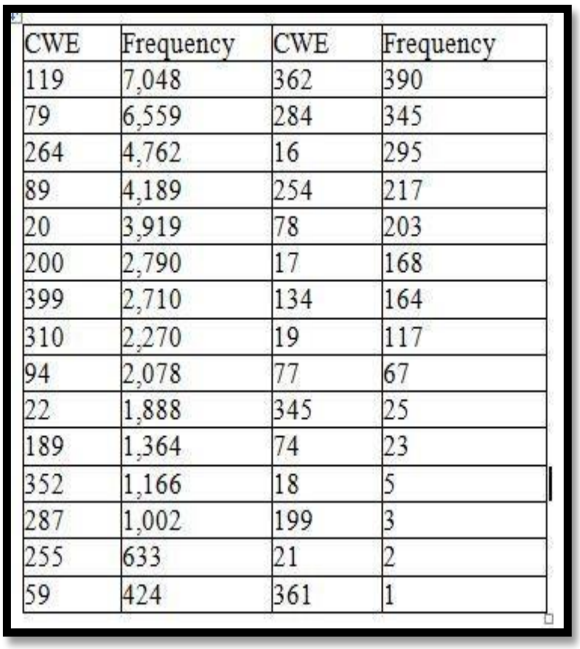
IV. CLASSIFICATION FOR THE DATA

A. Investigational Fact

Between 1999 and 2020, we gathered 77,885 CVE entries from NVD for the categorization and assessment. Among these, experimental data were derived from CVE entries including over 1,000 identified CWEs (Table I).

We categorized CWE-119 and CWE-79, which have the highest number of recognized CWEs, as well as the top ten CWEs in terms of CWE frequency, in this study. Table 2 summarizes these CWEs. For categorization and assessment, we used 500 CVE records for each category of CWE. As a result, a total of 10,000 CVE entries were used in this experiment, and they were used regardless of when they were published.

TABLE I. FIGURE OF RECOGNIZED CWES

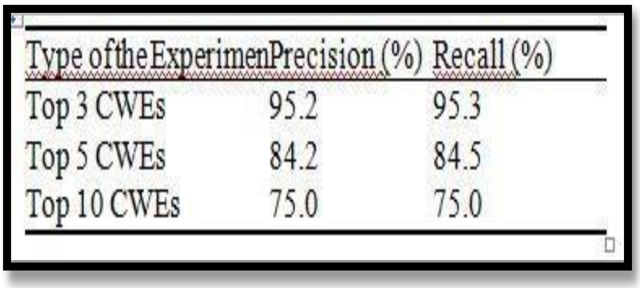


CWE	Frequency	CWE	Frequency
119	7,048	362	390
79	6,559	284	345
264	4,762	16	295
89	4,189	254	217
20	3,919	78	203
200	2,790	17	168
399	2,710	134	164
310	2,270	19	117
94	2,078	77	67
22	1,888	345	25
189	1,364	74	23
352	1,166	18	5
287	1,002	199	3
255	633	21	2
59	424	361	1

B. Investigational Outcome

To begin, we divided the experimental dataset into two groups according to the number of CWEs it contained: CWE-119 and CWE-79. In the second experiment, we divided the experimental dataset into two categories according to the number of CWEs it contained: CWE-119 and CWE-79. The categorization model was 99.8 percent correct in its classification. After that, the top three CWEs and the top five CWEs were correctly categorized with 95.1 percent and 84.5 percent accuracy, respectively, in the next experiment. A 75.5 percent accuracy rate was achieved in the most recent experiment, which categorized the top 10 CWEs. The accuracy and recall values for each trial are included in Table I of this report. Because several CWEs had a similar vulnerability overview, certain CVE entries were wrongly classified as related vulnerabilities in the top 5 and top 10 CWEs tests as a result of the similarity in vulnerability overview across the CWEs [3].

TABLE II. FIGURE OF RECOGNIZED CWES



Type of the Experimen	Precision (%)	Recall (%)
Top 3 CWEs	95.2	95.3
Top 5 CWEs	84.2	84.5
Top 10 CWEs	75.0	75.0

V. CONCLUSION

We propose and evaluate a naive Bayes-based classification method to classify CVE entries into vulnerability types. We wish to improve the accuracy of the model by systematically analyzing several vulnerabilities using similar languages and advanced functional engineering. Finally, we will examine the unidentified CVE entries.

REFERENCES

[1] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, “A Search Engine Backed by Internet-Wide Scanning,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA, Oct. 2015, pp. 542–553. doi: 10.1145/2810103.2813703.

[2] H. Booth, D. Rike, and G. Witte, “ITL BULLETIN FOR DECEMBER 2013 THE NATIONAL VULNERABILITY DATABASE (NVD): OVERVIEW,” p. 3.

[3] S. Na, T. Kim, and H. Kim, “A Study on the Classification of Common Vulnerabilities and Exposures using Naïve Bayes,” in *Advances on Broad-Band Wireless Computing, Communication and Applications*, vol. 2, L. Barolli, F. Xhafa, and K. Yim, Eds. Cham: Springer International Publishing, 2017, pp. 657–662. doi: 10.1007/978-3-319-49106-6_65.

[4] “Shodan,” *Shodan*. https://www.shodan.io (accessed Mar. 10, 2022).

[5] Y.-Y. Chang, P. Zavarsky, R. Ruhl, and D. Lindskog, “Trend Analysis of the CVE for Software Vulnerability Management,” in *2011 IEEE Third Int’l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int’l Conference on Social Computing*, Boston, MA, USA, Oct. 2011, pp. 1290–1293. doi: 10.1109/PASSAT/SocialCom.2011.184.

[6] S. Neuhaus and T. Zimmermann, “Security Trend Analysis with CVE Topic Models,” in *2010 IEEE 21st International Symposium on Software Reliability Engineering*, San Jose, CA, USA, Nov. 2010, pp. 111–120. doi: 10.1109/ISSRE.2010.53.

[7] M. Guo and J. A. Wang, “An Ontology-based Approach to Model Common Vulnerabilities and Exposures in Information Security,” p. 10, 2009.

[8] Z. Li, L. Tan, X. Wang, S. Lu, Y. Zhou, and C. Zhai, “Have things changed now?: an empirical study of bug characteristics in modern open source software,” in *Proceedings of the 1st workshop on Architectural and system support for improving software dependability - ASID ’06*, San Jose, California, 2006, pp. 25–33. doi: 10.1145/1181309.1181314.

[9] “CV-HG-2019-Official-Annual-Cybercrime-Report (1).pdf.”

[10] “TR_Dimensionality_Reduction_Review_2009 (2).pdf.”

[11] X. Gong, Z. Xing, X. Li, Z. Feng, and Z. Han, “Joint Prediction of Multiple Vulnerability Characteristics Through Multi-Task Learning,” in *2019 24th International Conference on Engineering of Complex Computer Systems (ICECCS)*, Guangzhou, China, Nov. 2019, pp. 31–40. doi: 10.1109/ICECCS.2019.00011.

[12] Furlanello, T., Lipton, Z.C., Tschannen, M., Itti, L. and Anandkumar, “Born-Again Neural Networks.”