2023 3rd International Conference on Electronic Information Engineering and Computer Science (EIECS) | 979-8-3503-1667-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/EIECS59936.2023.10434284

Tracing Vulnerability to Attack Patterns Using Text Similarity

Mengyuan Pan
Sun Yat-sen University
School of Systems Science and Engineering
Guangzhou, China
panmy6@mail2.sysu.edu.cn

Bin Li
State Grid
State Grid Henan Electric Power Research Institute
Zhengzhou, Henan
flyover586@sina.com

Yiwei Zou
Sun Yat-sen University
School of Systems Science and Engineering
Guangzhou, China
zouyw3@mail2.sysu.edu.cn

Wen Yang
State Grid
State Grid Henan Electric Power Research Institute
Zhengzhou, Henan
yangwwww@126.com

Dan Wang
State Grid
State Grid Henan Electric Power Research Institute
Zhengzhou, Henan
hndlzdh@126.com

Tao Zhang*
Sun Yat-sen University
School of Systems Science and Engineering
Guangzhou, China
zhangt358@mail.sysu.edu.cn

***Abstract*—In order to quickly discover attack patterns that may be used against vulnerabilities and provide security personnel with more security knowledge and reference information, we propose a Weighted-SBERT (WSBERT) based text similarity calculation method to trace vulnerability to attack patterns. The WSBERT method take representative keywords into consideration to get more accurate sentence embeddings and calculate the similarities between attack patterns and vulnerabilities using sentence embeddings. To demonstrate the effectiveness of our proposed method, we compare the WSBERT method to the TF-IDF similarity method, and conduct ablation experiments. The results indicate that our WSBERT method performs better than the unimproved SBERT method, has similar performance to the TF-IDF method, and possesses stronger generalization ability and the ability to process long texts.**

Keywords- network security, knowledge fusing, vulnerabilities, attack patterns

I. INTRODUCTION

To better study and address potential network threats and help security professionals better remediate vulnerabilities and defend systems, several public security knowledge bases have been proposed and rapidly developed. Currently, the three most commonly used security knowledge bases are CVE, CWE, and CAPEC. CVE [1] catalogues known network security-related vulnerabilities worldwide and provides detailed and standardized descriptions, with the number of vulnerabilities recorded increasing every day. CWE [2] catalogs weaknesses that exist in different types of software and hardware, where a weakness refers to an error, negligence, or other issues that have not been effectively addressed in the hardware, software deployment, design, or construction process, and if exploited, would be observed as CVE vulnerabilities. CAPEC [3] records common attack patterns to help users understand how attackers exploit weaknesses in a system to achieve their attack goals and provides some detection and mitigation recommendations. These three knowledge bases describe threats from different perspectives. With the use of these knowledge bases, security personnel can better comprehend the potential attack behaviors of threat actors, the causes of system vulnerabilities, and apply them for system repair, attack simulation [4–6], and anomaly detection [7–9]. However, different knowledge bases cannot be completed fused and utilized due to the missing linkage between them.

To predict the relationships between CVE and CAPEC, Navarro et al. [10] utilized doc2vec to vectorize CVE and CAPEC and calculated similarity to discover their connections. They combined this method with host log analysis to detect potential threats in the system. However, Navarro et al.'s research focused on how to utilize prior knowledge for threat detection and did not discuss the accuracy of their discovered connections. Dang et al. [11] utilized text-based, graph-based, and recommendation-based approaches to predict the relationships between CVE and CWE, CWE and CAPEC and link these three knowledge bases. Similarly, Das et al. proposed VWC-MAP to connect CVE and CAPEC indirectly by integrating CWE and CAPEC relationship prediction into V2W-BERT [12]. However, neither Dang et al. nor Das et al. discussed the accuracy and feasibility of such indirect connections. Kanakogi et al. [13,14] used TF-IDF, USE (Universal Sentence Encoder), and Sentence-BERT (SBERT) [15] to calculate text similarity between CVE and CAPEC descriptions to identify potential CAPEC attack patterns corresponding to CVE vulnerabilities. Through experiments, Kanakogi et al. found that TF-IDF yielded the best performance, but they did not consider that different words may have varying contribu-

tions to the semantic meaning of the sentence in a specific domain when calculating sentence embeddings.

Currently, progress has been made in predicting relationships between CVE and CAPEC. To the best of our knowledge, there are two main research approaches for predicting relationships between CVE and CAPEC. One is to directly calculate text similarity between CVE and CAPEC to discover potential connections, and the other is to indirectly trace CVE to CWE and find the CAPEC corresponding to the CWE, thus the linkage between CAPEC and CVE can be established.

In this paper, we propose a Weighted SBERT (WSBERT) method based on SBERT, a NER model using BERT-BiLSTM-CRF, and TF-IDF weighting to extract representative information from text to calculate text similarity between CVE and CAPEC. We select 63 CVE vulnerabilities defined as related to CAPEC attack patterns as test data and evaluate our method using the Recall@n. Experimental results show that our WSBERT method has superior performance and outperforms unweighted SBERT methods.

The rest of this paper is organized as follows: Section 2 introduce a motivating example to discuss the feasibility of our work. We discuss our method in detail in Section 3. In Section 4, our experiments are illustrated and discussed. Finally, we make a conclusion in Section 5.

II. SIMILARITIES BETWEEN CVE AND CAPEC DESCRIPTION

Some CVE vulnerabilities may appear as example instances in the description of a corresponding CAPEC attack pattern, which can be understood as a concrete manifestation of the CVE vulnerability in the CAPEC attack pattern. Taking CAPEC-31 (Accessing/Intercepting/Modifying HTTP Cookies) as an example, it corresponds to two example instances, namely CVE-2010-5148 and CVE-2016-0353.

CVE-2016-0353 is caused by two CWE weaknesses, namely CWE-254 (7PK-Security Features) and CWE-200 (Exposure of Sensitive Information to an Unauthorized Actor). However, neither of these weaknesses are linked with CAPEC-31. Therefore, if linkage between CVE and CAPEC are discovered indirectly, the linkage between CVE-2016-0353 and CAPEC-31 cannot be accurately established. Additionally, as CWE weaknesses are abstractions and summaries of multiple vulnerabilities with similar features, the mapping between vulnerabilities and weaknesses is not a one-to-one relationship, and multiple vulnerabilities with very different exploitation methods may be caused by the same weakness, for instance, XSS and SQL injection can both be caused by CWE-20 (Improper Input Validation), however, they are two different kinds of vulnerabilities. Furthermore, a single weakness can be linked with numerous attack patterns, for example, CWE-200 is linked with 59 attack patterns. Such a large number of attack patterns has limited usefulness for understanding and mitigating vulnerabilities, and we require more precise related attack patterns with vulnerabilities. Therefore, linking CVE and CAPEC indirectly may not be a sufficiently rational approach.

Some description such as “session cookie” and “http session” in CVE-2016-0353, these words also appear in CAPEC-31, these words could make great contributions when calculating similarities between texts and therefore it is possible to discover linkage between the two by utilizing text similarity methods. In fact, it is reasonable to generalize this conclusion to all CVE vulnerabilities since security practitioners tend to use common cybersecurity terminology when describing vulnerabilities. Therefore, it can be concluded that utilizing text similarity to discover linkage between CVE and CAPEC has better rationality.

III. WSBERT

A. *Focusing on representative keywords*

When calculating sentence embedding from word embedding, SBERT uses mean-pooling and treats all word embedding as equally important. However, this approach may lead to the submersion of critical information in a large amount of irrelevant information when sentences are lengthy, resulting in inaccurate representation of sentence semantics. To address this issue, a natural idea is to assign larger weight for those words that contribute more to the semantics of sentences which in this paper we call them representative keywords during the pooling process. This approach enables the preservation of low-importance information semantics while focusing on critical information semantics, leading to more accurate representation of sentence semantics.

For CVEs, most descriptions follow a fixed pattern of expression which is a Weakness in a Product that can be exploited in a specific way, eventually leading to a Consequence. The Weakness represents a flaw in software or hardware and how an attacker may exploit that flaw, while the Consequence represents the potential outcomes that could arise from exploiting the weakness.

An example of such a CVE description is CVE-2007-1544, illustrated in Figure 1, which is one of the example instances mentioned in Section 2. This vulnerability is linked with CAPEC-92 (Forced Integer Overflow), a common attack pattern that describes how to exploit an integer overflow weakness and the potential consequences that may result from it. In this case, the “Network Audio System” product contains an “Integer overflow” weakness that can be exploited through a “large max_samples value,” eventually leading to “denial of service (crash)” and “execute arbitrary code.” Similarly, CAPEC also includes a detailed description of “Integer overflow” weakness, the attack patterns that exploit these weakness, and the potential consequences of such attacks. Therefore, in the process of calculating textual similarity, the above-mentioned description are the most significant semantic contributions.

Integer overflow in the ProcAuWriteElement function in server/dia/audispatch.c in Network Audio System (NAS) before 1.8a SVN 237 allows remote attackers to cause a denial of service (crash) and possibly execute arbitrary code via a large max_samples value .

Figure 1. Representative keywords in CVE-2007-1544

B. Structure of WSBERT

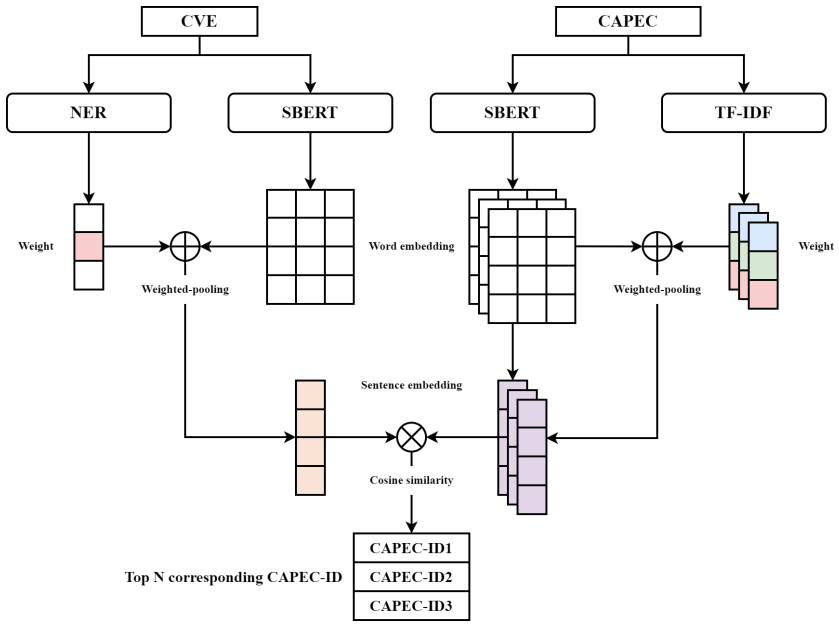

Figure 2. WSBERT

To better represent the semantic knowledge, and discover the linkage between CVE and CAPEC, this paper proposes a WSBERT method to calculate similarities between CVE and CAPEC description. Our WSBERT approach first feeds the CVE and preprocessed CAPEC data into the SBERT to obtain word embeddings. Next, the word embeddings are weighted-pooled to get thee sentence embedding. Then the cosine similarity is calculated between the CVE and various CAPECs, thus obtaining the most likely corresponding CAPEC-ID for the CVE. The WSBERT process is illustrated in Figure 2.

The data flow of Weighted SBERT is as follows:

Data pre-processing For CAPEC documents, stop words and non-alphabetic characters were filtered out to eliminate redundant information unrelated to the semantic of the sentence, and lemmatization was applied to eliminate the error introduced by different word forms in calculating the TF-IDF values. For CVE descriptions, subsequent sentence vectorization using SBERT does not have strict requirements on word forms as in the TF-IDF method. Moreover, filtering out redundant information such as stop words can disrupt sentence structure, making it difficult for NER model to learn the sentence pattern. Therefore, no data preprocessing was performed on CVE descriptions.

Sentence embedding The preprocessed data was tokenized using AutoTokenizer and input into SBERT to obtain word embeddings. For CAPEC documents, we adopt the TF-IDF weighted-pooling method, which involves inputting the CAPEC documents into a TF-IDF vectorizer to obtain the TF-IDF score of each word. Then, the TF-IDF score is multiplied by the corresponding word embedding and averaged to obtain the sentence embedding. As for the CVE descriptions, we first extract representative keywords from the descriptions using a pretrained BERT-BiLSTM-CRF NER model. We then assign weights to the keywords, and in order to reduce the difficulty of NER extraction, we merge the Weakness and Consequence keywords into Weak-Cons entity and assign it a weight of 10 in subsequent processing, while the weight of non-representative keywords is set to 1. Finally, we use the obtained word weight to perform weighted-pooling by multiplying it with the word embedding and taking the average to obtain the sentence embedding.

Similarity calculation For the CVE sentence embedding and CAPEC sentence embeddings, we compute the cosine similarity between them and sort the results in descending order. This process yields the Top-N CAPEC-IDs corresponding to the given CVE description.

IV. EXPERIMENTS

A. Evaluation settings

As explained in Section 2, there are 63 CVE vulnerabilities that of example instances in CAPEC descriptions and are associated with CAPEC attack patterns. These vulnerabilities are considered by security experts to be related to their corresponding attack patterns. Therefore, we selected these 63 CVE vulnerabilities as our test dataset to evaluate the performance of our proposed WSBERT method. The average number of words in the descriptions of these 63 vulnerabilities is 38.

We use Recall@n, n ranging from 1 to 15 to evaluate our CVE-CAPEC linking prediction performance. Recall@n is a commonly used evaluation metric in information retrieval and recommendation systems. It measures the proportion of relevant items in the top n items recommended by a system.

We make comparison experiments with the TF-IDF method proposed by Kanakogi et al [14]. And for ablation study, we compare our proposed method with other three methods

which are the SBERT without weighted-pooling, SBERT with TF-IDF weighted pooling for both CAPEC documents and CVE description, SBERT with TF-IDF weighted-pooling for only CAPEC documents.

B. *Tracing performance evaluation*

The ablation study results are illustrated as Figure 3, where **A** represents SBERT without weighted-pooling, **B** represents SBERT with TF-IDF weighted-pooling for CAPEC documents only, and **C** represents SBERT with TF-IDF weighted-pooling for both CAPEC documents and CVE descriptions. Experiment results demonstrate that all three weighted average methods for SBERT show significant performance improvement compared to the unweighted SBERT method. Our proposed WSBERT method exhibits the best performance, followed by the SBERT method with TF-IDF weighted-pooling for CAPEC documents only. However, the performance obtained by TF-IDF weighted-pooling for both CVE descriptions and CAPEC documents is slightly lower than that of TF-IDF weighted-pooling for CAPEC documents only.

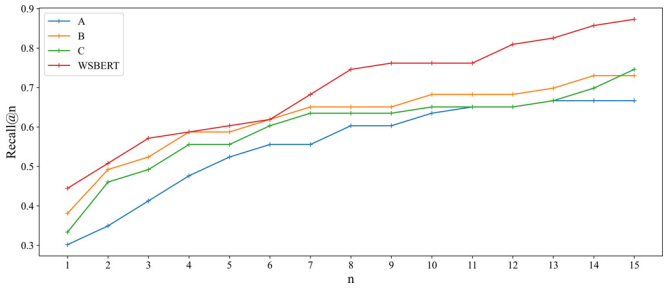

Figure 3. Ablation study

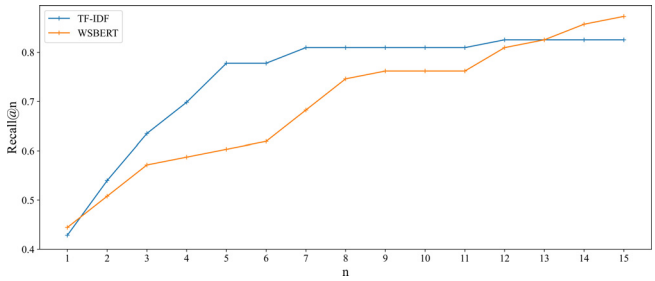

Figure 4. Comparison result

The performance comparison between WSBERT and TF-IDF is illustrated in Figure 4. The experimental results demonstrate that our proposed WSBERT method exhibits the best performance at Recall@1, but is gradually surpassed by the TF-IDF method afterwards. Eventually, the performance of WSBERT method catches up with that of TF-IDF at Recall@13 and surpasses it afterwards.

Although the TF-IDF method performs slightly better than our proposed WSBERT method at Recall@2 to Recall@12, it is known that the TF-IDF method represents different words using different dimensions in the vector space, whereas the WSBERT method maps different words onto the same vector space. When two texts with the same semantic meaning are described using the same vocabulary, the TF-IDF method can obtain good performance and represent the similarity between texts accurately. However, when the two texts use different vocabularies, the method cannot calculate text similarity well, which means that its ability to handle synonyms and related words is weak and its generalization ability is poor. Furthermore, the TF-IDF method considers the contribution of each word in the semantic meaning based on its TF-IDF value, but this value cannot effectively represent the importance of words in calculating the similarity between CVE and CAPEC. Consequently, when facing long texts, representative keywords in CVE descriptions can be easily buried by redundant information. On the other hand, the WSBERT method is trained with a large corpus, which allows it to better represent the similarity between words in the vector space, obtain word semantics and mitigate the problem of synonyms and related words. Therefore, it has a stronger generalization ability, although this might introduce errors caused by synonyms. Additionally, the WSBERT method uses NER to extract the information that contributes most to semantic meaning in CVE descriptions and assigns them a higher weight, which enables the sentence's semantic meaning to be well represented even when facing long texts.

C. *Beyond the selected 63 CVE vulnerabilities*

To demonstrate the generalization ability of the TF-IDF and WSBERT methods, we created the following test case, assuming that a vulnerability description includes the term "global positioning system" or "GPS", it is obvious that they express the same meaning, but the latter is more commonly used in security-related descriptions. In CAPEC, there is an attack pattern related to CAPEC-627 (Counterfeit GPS signals), but the description of this attack pattern does not include the phrase "global positioning system". We input this phrase into both models to test if they can correctly match the CAPEC-627 attack pattern in top 5 results. The result shows that the WSBERT method correctly matches CAPEC-627, while the TF-IDF method failed to do so. In addition, the WSBERT method also matches CAPEC-559 (Orbital Jamming), an attack pattern that also affects GPS by disrupting its normal transmission through satellite links. Therefore, it can be concluded that our proposed WSBERT method has stronger generalization ability.

To demonstrate the advantages of the WSBERT method over the TF-IDF method in handling long texts, we selected CVE-2017-5219 as an input. The description of this vulnerability, as shown in Figure 5, contains 129 words, which is much greater than the average word count of 38 in the 63 CVE descriptions in the test set. The vulnerability is related to the possibility of web shell upload. When feeding this vulnerability into the TF-IDF and WSBERT models, the results show that the WSBERT method match the CAPEC-650 (Upload a Web Shell to a Web Server) attack pattern, while TF-IDF failed to match this attack pattern.

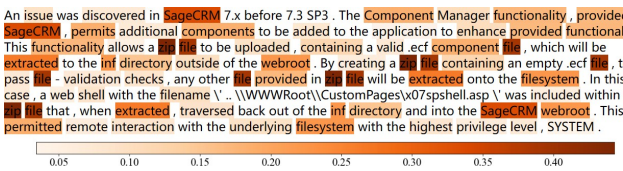

Figure 5. TF-IDF value heat map of CVE-2017-5219

As shown in the TF-IDF value heat map in Figure 5, it can be seen that in the description of CVE-2017-5219, keywords such as “zip file” and “filesystem” have larger TF-IDF values, but keywords such as “web shell” have smaller TF-IDF values, greater TF-IDF value contributes more when calculating text similarity. This explains why the TF-IDF method failed to correctly match attack patterns related to “web shell” but matches files-related attack patterns such as CAPEC-177 (Create files with the same name). This problem becomes more severe as the text length increases, as more redundant information with high TF-IDF values may be introduced. Our proposed WSBERT method, on the other hand, avoids this problem by identifying representative keywords with maximum semantic contribution in matching attack patterns and assigning them with large weights, thus avoiding them getting drowned out in redundant information.

V. Conclusion

In this paper, we propose a WSBERT method for discovering CAPEC attack patterns that may be linked with CVE vulnerabilities by calculating text similarity. We compare the WSBERT method with the TF-IDF method, and conduct ablation experiments. The results show that the proposed WSBERT method performs well, and can find linkage between vulnerabilities and attack patterns by ranking them based on the text similarity of their descriptions, providing effective references for security experts and assisting them in rapidly identifying potential attacks. However, the ability of our proposed WSBERT method to accurately identify linkage needs improvement. In the future, we will attempt to fine-tune the WSBERT model using CVE and CAPEC data, to better represent semantics in network security, improve text similarity calculation accuracy, and discover more accurate linkage.

Acknowledgement

This work is supported by “Research on Intelligent Judgment and Endogenous Defense Technology of APT Attack in Power Monitoring System”, 5108-202324069A-1-1-ZN, a science and technology project of State Grid Co., Ltd in 2023.

References

[1] CVE - CVE. https://cve.mitre.org/.

[2] CWE - Common Weakness Enumeration. https://cwe.mitre.org/.

[3] CAPEC - Common Attack Pattern Enumeration and Classification (CAPEC™). https://capec.mitre.org/.

[4] Ferda ..zdemir S..nmez, Chris Hankin, and Pasquale Malacaria. Attack Dynamics: An Automatic Attack Graph Generation Framework Based on System Topology, CAPEC, CWE, and CVE Databases. Computers & Security, 123:102938, December 2022.

[5] Igor Kotenko and Elena Doynikova. The CAPEC based generator of attack scenarios for network security evaluation. In 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), pages 436–441, Warsaw, Poland, September 2015. IEEE.

[6] Weilin Wang, Huachun Zhou, Kun Li, Zhe Tu, and Feiyang Liu. Cyber-Attack Behavior Knowledge Graph Based on CAPEC and CWE Towards 6G. In International Symposium on Mobile Internet Security, pages 352–364. Springer, 2021.

[7] Kabul Kurniawan, Andreas Ekelhart, and Elmar Kiesling. An ATT&CK-KG for Linking Cybersecurity Attacks to Adversary Tactics and Techniques. 2021.

[8] Sandeep Nair Narayanan, Ashwinkumar Ganesan, Karuna Joshi, Tim Oates, Anupam Joshi, and Tim Finin. Early detection of cybersecurity threats using collaborative cognition. In 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), pages 354–363. IEEE, 2018.

[9] Wei Wang, Rong Jiang, Yan Jia, Aiping Li, and Yi Chen. KGBIAC: Knowledge Graph Based Intelligent Alert Correlation Framework. In Sheng Wen, Wei Wu, and Aniello Castiglione, editors, Cyberspace Safety and Security, volume 10581, pages 523–530. Springer International Publishing, Cham, 2017.

[10] Julio Navarro, Véronique Legrand, Sofiane Lagraa, Jérôme François, Abdelkader Lahmadi, Giulia De Santis, Olivier Festor, Nadira Lammari, Fayçal Hamdi, Aline Deruyver, Quentin Goux, Morgan Allard, and Pierre Parrend. HuMa: A Multi-layer Framework for Threat Analysis in a Heterogeneous Log Environment. In Abdessamad Imine, José M. Fernandez, Jean-Yves Marion, Luigi Logrippo, and Joaquin Garcia-Alfaro, editors, Foundations and Practice of Security, volume 10723, pages 144–159. Springer International Publishing, Cham, 2018.

[11] Quang-Vinh Dang and Jérôme François. Utilizing attack enumerations to study SDN/NFV vulnerabilities. In 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), pages 356–361. IEEE, 2018.

[12] Siddhartha Shankar Das, Edoardo Serra, Mahantesh Halappanavar, Alex Pothen, and Ehab Al-Shaer. V2w-bert: A framework for effective hierarchical multiclass classification of software vulnerabilities. In 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA), pages 1–12. IEEE, 2021.

[13] Kenta Kanakogi, Hironori Washizaki, Yoshiaki Fukazawa, Shinpei Ogata, Takao Okubo, Takehisa Kato, Hideyuki Kanuka, Atsuo Hazeyama, and Nobukazu Yoshioka. Comparative Evaluation of NLP-Based Approaches for Linking CAPEC Attack Patterns from CVE Vulnerability Information. Applied Sciences, 12(7):3400, 2022.

[14] Kenta Kanakogi, Hironori Washizaki, Yoshiaki Fukazawa, Shinpei Ogata, Takao Okubo, Takehisa Kato, Hideyuki Kanuka, Atsuo Hazeyama, and Nobukazu Yoshioka. Tracing cve vulnerability information to capec attack patterns using natural language processing techniques. Information, 12(8):298, 2021.

[15] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084, 2019.