REVS: A Vulnerability Ranking Tool for Enterprise Security

Igor Forain[a], Robson de Oliveira Albuquerque[b] and Rafael Timóteo de Sousa Júnior[c]
Professional Program in Electrical Engineering (PPEE), Dept. of Electrical Engineering (ENE), University of Brasília (UnB), Brasília, Brazil

Keywords: Cybersecurity, Vulnerabilities, Pentest, NVD, CNVD, TOPSIS.

Abstract: Information security incidents currently affect organizations worldwide. In 2021, thousands of companies suffered cyber attacks, resulting in billions of dollars in losses. Most of these events result from known vulnerabilities in information assets. However, several heterogeneous databases and sources host information about those flaws, turning the risk assessment difficult. This paper proposes a Recommender Exploitation-Vulnerability System (REVS) with the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) to rank vulnerability-exploit. The REVS is a dual tool that can pinpoint the best exploits to pentest or the most sensitive vulnerabilities to cybersecurity staff. This paper also presents results in the GNS3 emulator leveraging data from the National Vulnerability Database (NVD), the China National Vulnerability Database (CNVD), and Vulners. They reveal that the CNVD, despite data issues, has 23,281 vulnerabilities entries unmapped in the NVD. Moreover, this work establishes criteria to link heterogeneous vulnerability databases.

1 INTRODUCTION

Today, most organizations need to provide services in computing environments. Beyond the impact on the healthcare industry worldwide, the COVID-19 epidemic has also had a disruptive effect on the way businesses operate (Ferreira et al., 2021). Once performed in person, services and processes had to undergo an almost instantaneous digital re-adaptation to remote access (Khan et al., 2020).

The digitalization of processes has been underway since the Internet became ubiquitous. However, it increased the attack surface (Pimenta Rodrigues et al., 2017; Chowdhary et al., 2020; Gualberto et al., 2020). The rush to provide services based on Cloud Computing has been accompanied by the introduction of a series of vulnerabilities in the IT environments of organizations, especially with the advent of services based on the Internet of Things (IoT) (Liu et al., 2019; Thamilarasu and Chawla, 2019).

Although there is a plethora of data regarding information security, several heterogeneous databases and sources host those data, turning the risk assessment difficult (Du et al., 2019). Besides, most organizations still lack effective methods to choose the best option and path in real scenarios (Bertoglio and Zorzo, 2017). The National Vulnerability Database (NVD) from the National Institute of Standards and Technology (NIST) provides reliable information about software and hardware flaws (Mavroeidis and Bromander, 2017; Hemberg et al., 2020). However, there might be a delay between NVD and other open sources (Rodriguez et al., 2018). For this reason, new databases like China National Vulnerability Databases (CNVD) (CNCERT/CC, 2021), Metasploit, and Vulners (VULNERS, 2021) can help achieve better situational awareness of enterprise risks.

Also, picking the optimal attack action and exploit in the wild is still an open question (Kanakogi et al., 2021a). To this end, Vulnerability Assessment (VA) and Penetration Testing (PT) are essential steps (Shah and Mehtre, 2015; Yaqoob et al., 2017; Ghanem and Chen, 2020; Zhou et al., 2021). However, choosing the most critical and threatening outcomes provided by those steps is a decision process subjected to several constraints.

There are several approaches to solve decision process issues. They can use from Markov chain to Deep Learning algorithms (Awiszus and Rosenhahn, 2018). However, the VA is bounded by several attributes from the Common Vulnerability Scoring System (CVSS) (Cheng et al., 2012). So, it becomes

[a] https://orcid.org/0000-0002-9965-8077
[b] https://orcid.org/0000-0002-6717-3374
[c] https://orcid.org/0000-0003-1101-3029

Forain, I., Albuquerque, R. and Sousa Júnior, R.
REVS: A Vulnerability Ranking Tool for Enterprise Security.
DOI: 10.5220/0011068600003179
In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) - Volume 2*, pages 126-133
ISBN: 978-989-758-569-2; ISSN: 2184-4992

an optimization problem that can leverage operations research methods like the Multi-Criteria Decision-Making (MCDM) (Dožić, 2019). The Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) is an algorithm of the MCDM family used for the cybersecurity of 5g networks (Kholidy, 2022), power control systems (Liu et al., 2010) and Intrusion Detection System (IDS) (Alharbi et al., 2021)

Based on the described open issues, this work aims to optimize the vulnerability-exploit choosing process and integrate new vulnerability databases. To this end, this work proposes the Recommender Exploitation-Vulnerability System (REVS) based on a MCDM approach. It leverages data from the NVD, CNVD, and Vulners to create a vulnerability-exploit ranking using the TOPSIS algorithm. REVS is a dual tool that can pinpoint the best exploits to attackers or the most sensitive vulnerabilities to cybersecurity staff. The experimental results reveal that the CNVD, despite data issues, has 23,281 vulnerabilities entries unmapped in the NVD. Moreover, this work establishes criteria to link heterogeneous vulnerability databases.

This work is structured as follows. Section 2 presents background and related work. Section 3 describes the proposed architecture. Section 4 uses GNS3 and open source tools to implement the architecture. Finally, section 5 concludes this article.

2 LITERATURE REVIEW

This section presents a literature review related to this work. It presents vulnerabilities and exploits and their available databases. Besides, it reviews approaches regarding attack path prediction and pentest automation. Finally, it shows state-of-the-art recommendation algorithms applied in the cybersecurity field.

2.1 Vulnerabilities and Exploits

Wang and Guo proposed an Ontology for Vulnerability Management (OVM) to achieve knowledge representation of the Common Vulnerabilities Exposures (CVE) of the NVD (Wang and Guo, 2009). It was the first attempt to create a knowledge base (KB) of vulnerabilities, but it did not consider the exploits possibility, different from this work. GAO et al. created a taxonomy and ontology for network attack classification using description logic (DL) (Gao et al., 2013). Kanakogi et al. used natural language processing (NLP) to match CVE to Common Attack Pattern Enumerations and Classifications (CAPEC) (Kanakogi et al., 2021a; Kanakogi et al., 2021b). These works guided the CVE data model that REVS uses for vulnerability databases.

Householder et al. evaluated a systematic analysis of the relation between vulnerabilities and exploits. By the end of 2019, only around 4.1% of the vulnerabilities exposed after 2013 had an exploit publication (Hu et al., 2020). It supported this work in giving more weight to vulnerability features than exploit.

Rodriguez et al. showed that vulnerability disclosure delay between the NVD and other open sources, e. g. in SecurityFocus, may reach 244 days (Rodriguez et al., 2018). Rytel et al. compared several databases of vulnerabilities, like the NVD and CNVD, but only regarding the IoT devices (Rytel et al., 2020). They presented the necessity of using more vulnerability databases besides the NVD.

The works presented in this subsection proposed vulnerability ontologies or evaluated vulnerability database assessment. Different from them, this work leverages and integrates those databases for pentest and security assessment.

2.2 Attack Path and Pentest

Valea and Oprisa proposed pentest automation using Nmap and Metasploit. Unlike this work, it covered only vulnerabilities regarding a root shell with Meterpreter and used decision trees to avoid overfitting (Valea and Oprişa, 2020). REVS leverages the TOPSIS algorithm. Polatidis et al. (Polatidis et al., 2017; Polatidis et al., 2020) proposed an attack path prediction algorithm to achieve an information risk assessment. Their work used only CVE data from the MITRE Corporation (MITRE, 2021) on IT maritime infrastructure to generate the attack graphs. This work uses more databases with a different optimization approach.

Huo et al. used the multi-host Multi-Stage Vulnerability Analysis (MulVAL) algorithm to generate the attack tree. It leveraged a Deep Q-Learning Network (DQN) to choose the best attack path based on the CVSS of the CVE as a reward function. Their experiment evaluated a small topology without specification about the information assets and exploits only CVE-2012-0053 (Hu et al., 2020). It showed the high computational cost of deep learning approaches for extensive networks and sometimes convergence issues. This work uses an operation research method and improves the integration between Nmap and external data sources.

2.3 Recommender Systems

Pawlicki et al. wrote a comprehensive survey about recommendation systems for Cybersecurity (Pawlicka et al., 2021). It showed to this work that type of systems as a promising approach to the vulnerability-exploit recommendation. Polatidis et al. used a recommendation system with a multi-level collaborative filtering method (Polatidis et al., 2018; Polatidis and Georgiadis, 2017) to Microsoft Windows. REVS leverages TOPSIS targeting any platform.

Some works treated the recommendation problem with an MCDM approach. It relies on minimizing or maximizing the geometric distance from an ideal solution like the classical recommendations systems. Most of them leveraged the TOPSIS for security assessment and Intrusion Detection System (IDS), but not for pentesting (Kholidy, 2022; Alharbi et al., 2021; Liu et al., 2010).

Table 1: Comparison with Related Works.

Work	*Data Sources*	*Algorithm*	*Scope*
(Pawlicka et al., 2021)	Doesn't apply	Doesn't apply	Survey
(Polatidis et al., 2018)	Mitre CVE	CF	MS Windows
(Polatidis and Georgiadis, 2017)	Mitre CVE	CF	MS Windows
(Kholidy, 2022)	NVD, Metasploit	TOPSIS	5G networks
(Alharbi et al., 2021)	Doesn't apply	TOPSIS	IDS attributes
(Liu et al., 2010)	Private	TOPSIS	Power Systems
This work	NVD, CNVD, Metasploit, Vulners	TOPSIS	IPv4, IPv6

Table 1 presents an outline comparison between the works cited in this subsection and this work. It shows that it leverages more vulnerability database sources to a broader network scope.

3 PROPOSED SYSTEM

This section describes the proposed architecture of the Recommender Exploitation-Vulnerability System (REVS). Figure 1 presents the system diagram, which has four main modules.

Four modules comprise the REVS: database, scanner, matcher, and recommender. The following subsections describe these four modules.

3.1 Database

The database module comprises two main sets: vulnerabilities and exploits. Their function is to provide numeric features to feed the TOPSIS decision matrix. REVS gathers the first set from two sources: the NVD and CNVD. The second set comes from Metasploit's database. This layer is a batch process to set up the

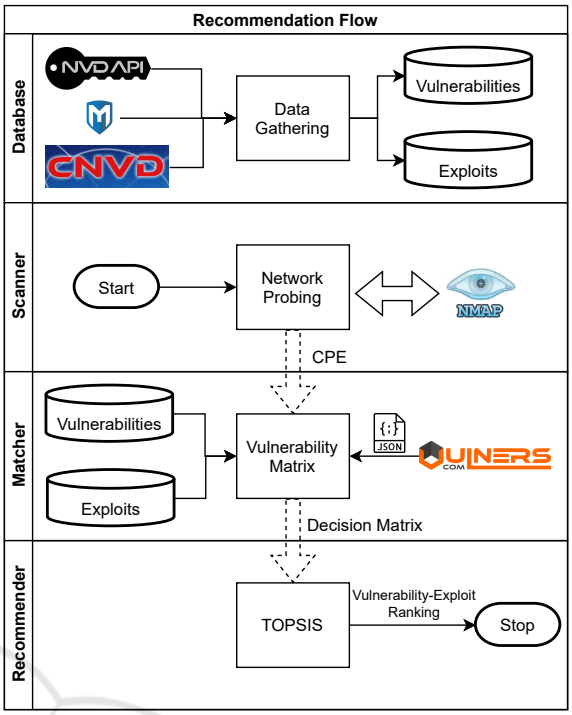

Figure 1: REVS Diagram Process.

databases before the Matcher step. REVS uses the NVD and CNVD in an ensemble, matching registers with the same CVE number and joining the scores of the two databases.

3.1.1 NVD

The NIST from the USA supports the NVD. It is a well-known and de facto authority regarding system vulnerabilities for the research community (NIST, 2021). That database comprises several features: Common Weakness Enumeration (CWE), Common Platform Enumeration (CPE), vendors, and ratings (MITRE, 2021). The NVD uses the CVSS to rate the characteristics and severity of system vulnerabilities. The CVSS has two versions, v2.0 and v3.0, with a different range for similar attributes. Table 2 shows the features from CVSS used by REVS.

Table 2 presents the six CVSS features that comprise the criteria used by the TOPSIS decision matrix.

3.1.2 CNVD

The National Computer Network Emergency Response Technical Team/Coordination Center of China (CNCERT/CC) sponsors the CNVD. China has been a global actor in the information security field, making the CNVD a relevant source of vulnerabilities (CNCERT/CC, 2021). However, it did not provide an interface for data feeding, nor did it provide

Table 2: CVSS Features.

CVSSv2	*CVSSv3*
accessComplexity	attackComplexity
accessVector	attackVector
availabilityImpact	availabilityImpact
confidentialityImpact	confidentialityImpact
integrityImpact	integrityImpact
authentication	privilegesRequired

any documentation. REVS implements a data crawler to get XML files from the CNVD. Moreover, it is supposed to use CVSS v2.0.

3.1.3 Metasploit

Metasploit works as a framework integrating reconnaissance tools, exploits, and payloads in the same environment. The metasploit source is the Exploit-DB maintained by the Offensive Security (Security, 2021). Moreover, Metasploit implements only a subset of exploits from Exploit-DB, most of them based on Ruby programming language. Besides, REVS is responsible for matching the framework's exploits with the gathered vulnerabilities because Metasploit is not entirely CVE oriented.

3.2 Scanner

REVS implements two scanning methods: network hosts status and CPE for the target services and OSs. The former uses the Nmap function to detect the network hosts' situation (up or down). The latter also leverages Nmap to identify CPEs of the opening services in each up machine. The CPE is the key used by the Matcher step to search for vulnerabilities and exploits in the gathered database. The more detailed the CPE information, the more likely the chance to match vulnerabilities and exploits for the target.

3.3 Matcher

This module uses the results from the Scanner to search for security data in the following databases: Vulnerabilities, Exploits, and Vulners API. The last is a web service provided by Vulners (VULNERS, 2021) an Information Security Company from Russia. The other two are databases collected by the first module described in subsection 3.1.

There is an interface to Vulners database embedded in Nmap and coded in Lua programming language (vulns script). The work of (Valea and Oprişa, 2020) leveraged it to carry out the Metasploit automation. However, REVS requires more vulnerability features to create the decision matrix for the TOPSIS algorithm. So, this work creates a new wrapper to Vulners API (https://vulners.com/api/v3/burp/software?) through Python classes.

3.4 Recommender

This work approach chooses the best vulnerability-exploit pair using a decision-making method. In this case, REVS uses the TOPSIS algorithm based on the optimization technique described in (Hwang and Yoon, 1981; Chen and Hwang, 1992; Opricovic and Tzeng, 2004). The attack is a one-layer problem of picking up the best vulnerability-exploit pair, i.e., the lowest cost and highest impact. So, the decision matrix has m vulnerabilities (rows) compared to n features (columns).

$$\mathbf{A} = (a_{ij}) \quad i = 1,2,...,m; j = 1,2,...,n \tag{1}$$

Equation 1 shows that the a_{ij} is the jth feature value of the ith vulnerability. After that, the $\mathbf{A}$ is normalized and weighted by columns to its $\mathbf{X}$ form:

$$\mathbf{X} = (x_{ij}) \quad x_{ij} = \lambda_j * a_{ij} / \sqrt{\sum_{j=1}^{n} a_{ij}^2} \tag{2}$$

The technique requires choosing the best and worst options from the attacker role.

$$\mathbf{Z}^+ = (z_1^+, z_2^+, z_3^+, ..., z_n^+) \tag{3}$$

$$\mathbf{Z}^- = (z_1^-, z_2^-, z_3^-, ..., z_n^-) \tag{4}$$

It is essential to understand if the column feature is a benefit or a cost. If it is a benefit, the z_j value for the best option is the maximum column value. Otherwise, it must be the minimum value. Equations 5 e 6 indicate the calculus for benefit and cost features, respectively.

$$z_j^+ = max(x_{ij}) \qquad z_j^- = min(x_{ij}) \tag{5}$$

$$z_j^+ = min(x_{ij}) \qquad z_j^- = max(x_{ij}) \tag{6}$$

Now, there are m vectors with dimension n, which are the rows of the matrix $\mathbf{X}$. There are also two new vectors with size n, Z^+ and Z^-, which are the ideal solutions. The TOPSIS requires calculating the Euclidean distance between the m vectors and the ideal solutions.

$$d_i^+ = \sqrt{(Z_i - X_+)^2} \quad (7)$$

$$d_i^- = \sqrt{(Z_i - X_-)^2} \quad (8)$$

Finally, calculate the performance ratio to rank each one of the vulnerabilities, from highest to lowest.

$$p_i = d_i^+ / (d_i^- + d_i^+), \qquad i = 1, 2, 3, ..., m \quad (9)$$

4 PRELIMINARY RESULTS

REVS is an ongoing project with preliminary results regarding the following modules: database, scanner, and matcher. This section presents the test environment based on the GNS3 and discusses those results.

4.1 Test Environment

This work emulates a medium enterprise Security Operation Center (SOC) using Graphical Network Simulator-3 (GNS3). It uses Quick Emulator (QEMU) on kernel-based virtual machines (KVM) in Linux Ubuntu 20.04. Furthermore, KVM in Linux performs better than type 2 hypervisors like VirtualBox and VMware because of hardware acceleration and kernel embedded commands. Figure 2 presents the test environment.

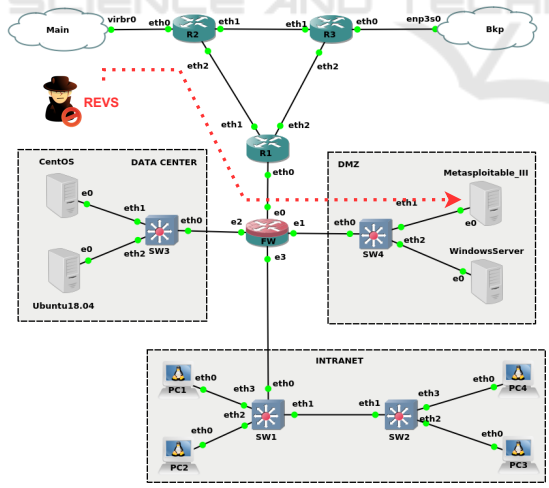

Figure 2: Emulated SOC with QEMU/KVN.

The GNS3 makes it possible to build an environment with different Operating Systems (OSs): Kali Linux, Ubuntu 18.04, CentOS 8, Windows Server, and Metasploitable III (Rapid7, 2021) machine. This last is a vulnerable public VM based on Ubuntu 14.04. Table 3 lists the environment hardware and VMs.

Table 3: Environment hardware and VMs.

Machine	***Description***
Host	Ryzen 7 4800h / 16gb RAM
Emulator	GNS3 2.2.28
Metasploitable III	Ubuntu 14.04
Firewall	pfSense 2.5.2
Routers	VyOS 1.1.8
Switches	Open vSwitch 2.4.0

4.2 Database Results

This work implements two data scrapers to collect vulnerabilities data from the NVD and CNVD. It downloads the two databases simultaneously on January 24, 2022, for a fair comparison. The first scraper uses the REST API supported by NIST, which provides vulnerability registers with unique CVE id and CVSS v2 and v3 scores. Besides, some of these registers also contain CPEs and CWEs related to the vulnerability. REVS makes use of the OPENCVE tool (Crocfer, 2020) to retrieve data from the REST API and store it locally in a PostgreSQL relational database.

The second scraper requires implementation from scratch. The CNVD does not provide any API to return vulnerability data. Otherwise, it provides a set of XML files lacking schema definition with part of the data scored by CVSS v2. Moreover, CNCERT/CC generates a new XML file every Monday 18h:00 (CST) with vulnerability registers from the past week. Different from the work of (Rytel et al., 2020), REVS uses the python requests library with custom "User-Agent" tag and cookie parameters (*__jsluid* and *__jsl_clearance_s*) to bypass the CNVD blocking system. Table 4 presents a summary of the NVD and CNVD.

Table 4: Summary of the National Databases.

Feature	***NVD***	***CNVD***
Vulnerability	178,906	99,261
Missing Weeks	6	0
Duplicated ID	0	88
Without CVE	0	23,281
Duplicated CVE	0	108
Nonexistent CVE	0	193

Table 4 shows that the CNVD dataset has the fol-

lowing issues: six missing week files, 23,281 vulnerabilities without corresponding CVE, 88 vulnerabilities with duplicated id, 108 CVEs linked to more than one vulnerability, and 193 entries linked to nonexistent CVEs. They look like malformed CVEs during string copy from other databases. Moreover, preceding files are not up-to-date; they started in January 2015, and most of the XML files have scape characters. REVS corrected these last issues and stored the CNVD data in a relational database on PostgreSQL. Figure 3 compares the NVD and the CNVD regarding the CVSS, CPE, and external references.

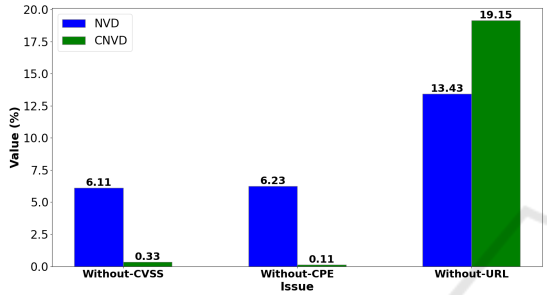

Figure 3: Databases Comparison.

Figure 3 shows that despite being the main responsible for CVE Mitre framework implementation, the NVD has 6.11% of the entries without any CVSS metric, while the CNVD has 0.33%. Moreover, the NVD and the CNVD have 6.23% and 0.11% of the entries without CPE, respectively. This last issue prevents a comprehensive identification of the vulnerable hardware or software. The NVD carries more information regarding external URL references, with 13.43% of the entries without this information, while the CNVD shows 19.15% missing this feature.

This work uses two approaches for exploits gathering: parsing the entire Metasploit source code and scraping the Exploit-DB website. The former executes a string pattern search looking for CVE mentions in the Metasploit source code structure. The latter scrapes the Exploit-DB website to get the relations between exploits and CVEs. After that, REVS also stores the results in PostgreSQL. Moreover, it leverages the Vulners API on the fly to search for vulnerabilities and exploits.

4.3 Scanner and Matcher Results

Figure 2 shows that the attack scenario uses REVS outside the topology against the Metasploitable III VM on the DMZ. In this scenario, REVS runs in the host machine against the guest VM emulated by GNS3 using the NVD and the CNVD as vulnerability database and Vulners as exploit source. It finds the results listed in table 5.

Table 5: Vulnerabilities and Exploits per CPE.

CPE	*NVD*	*CNVD*	*Exploits*
proftpd:proftpd:1.3.5	9	8	2
linux:linux_kernel	6	1	0
apache:http_server:2.4.7	44	38	53
samba:samba	1	1	0
apple:cups:1.7	6	0	1
mysql:mysql	2	0	0
	68	48	56

Table 5 shows that most of the vulnerabilities belong to the Apache webserver. The table also presents the same situation about the exploits available in the Metasploit framework. Table 5 also shows that REVS found 48 vulnerabilities using the CVE id as the search key in the CNVD. The 68 vulnerabilities detected by REVS using the NVD include these 48.

The 20 unmatched vulnerabilities in the CNVD are before 2015. It explains why they are not in the CNVD. However, a future approach is a new search for these unmatched results using CPE as the search key in the CNVD. Furthermore, a second official database, the China National Vulnerability Database of Information Security (CNNVD), also requires comparison against NVD and CNVD.

Figure 4 presents the data returned from REVS-Vulners interface regarding the CPE proftpd:proftpd:1.3.5.

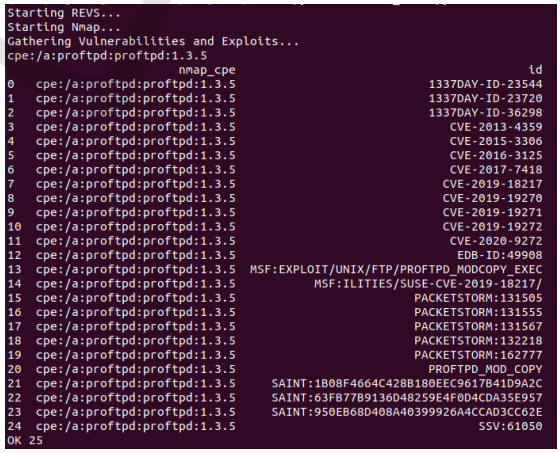

Figure 4: REVS and Vulners Interface Results.

Figure 4 shows that REVS-Vulners interface returns 9 CVEs and two exploits from Metasploit. Besides, it returns data regarding obsolete zero-days and forum messages.

5 CONCLUSIONS

The CNVD has several protection mechanisms to prevent downloads despite being a public database. Also, there are missing data files and 23,281 vulnerabilities without CVE mapping. It indicates process issues or the existence of vulnerabilities known only to the Chinese community because most of the text data are in mandarin.

REVS is an ongoing project that is working on the results of the Recommender module. It has already downloaded and normalized the NVD and CNVD databases. Furthermore, REVS integrated those two national vulnerability databases, Nmap and Vulners, using CPE and CVE as the search keys. The vulnerability assessment against the VM behind a SOC emulated in GNS3 showed the NVD as a more comprehensive database than the CNVD.

As Future work, the authors suggest using the TOPSIS fuzzy version with attack paths calculated by the MULVAL algorithm in the recommender module. The batch translation to English of the already downloaded database and an automatic one for new data are necessary improvements for REVS. This research will go further into integrating with the CNNVD and NLP approaches to vulnerability search.

ACKNOWLEDGEMENTS

The authors acknowledge the support of ABIN grant 08/2019. R.d.O.A. gratefully acknowledges the General Attorney of the Union - AGU grant 697.935/2019 and the General Attorney's Office for the National Treasure - PGFN grant 23106.148934/2019-67. R.T.S.J. gratefully acknowledges the support from EC Horizon 2020 HEROES project grant 101021801, CNPq grants 465741/2014-2 and 312180/2019-5, the Administrative Council for Economic Defense grant 08700.000047/2019-14, and the National Auditing Department of the Brazilian Health System SUS (Grant 23106.118410/2020-85).

REFERENCES

Alharbi, A., Seh, A. H., Alosaimi, W., Alyami, H., Agrawal, A., Kumar, R., and Khan, R. A. (2021). Analyzing the impact of cyber security related attributes for intrusion detection systems. *Sustainability*, 13(22):12337.

Awiszus, M. and Rosenhahn, B. (2018). Markov chain neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2180–2187.

Bertoglio, D. D. and Zorzo, A. F. (2017). Overview and open issues on penetration test. *Journal of the Brazilian Computer Society*, 23(1):1–16.

Chen, S.-J. and Hwang, C.-L. (1992). Fuzzy multiple attribute decision making methods. *Fuzzy multiple attribute decision making*, pages 289–486.

Cheng, P., Wang, L., Jajodia, S., and Singhal, A. (2012). Aggregating cvss base scores for semantics-rich network security metrics. In *2012 IEEE 31st Symposium on Reliable Distributed Systems*, pages 31–40. IEEE.

Chowdhary, A., Huang, D., Mahendran, J. S., Romo, D., Deng, Y., and Sabur, A. (2020). Autonomous security analysis and penetration testing. In *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*, pages 508–515. IEEE.

CNCERT/CC (2021). China National Vulnerability Database (CNVD). https://www.cnvd.org.cn/shareData/list. [Online; accessed 24-January-2022].

Crocfer, N. (2020). CVE Alerting Platform. https://github.com/opencve/opencve. [Online; accessed 06-December-2021].

Dožić, S. (2019). Multi-criteria decision making methods: Application in the aviation industry. *Journal of Air Transport Management*, 79:101683.

Du, G., Long, C., Yu, J., Wan, W., Zhao, J., and Wei, J. (2019). A real-time big data framework for network security situation monitoring. In *ICEIS (1)*, pages 167–175.

Ferreira, A., Cruz-Correia, R., et al. (2021). Covid-19 and cybersecurity: finally, an opportunity to disrupt? *Jmirx med*, 2(2):e21069.

Gao, J.-b., Zhang, B.-w., Chen, X.-h., and Luo, Z. (2013). Ontology-based model of network and computer attacks for security assessment. *Journal of Shanghai Jiaotong University (Science)*, 18(5):554–562.

Ghanem, M. C. and Chen, T. M. (2020). Reinforcement learning for efficient network penetration testing. *Information*, 11(1):6.

Gualberto, E. S., De Sousa, R. T., Thiago, P. D. B., Da Costa, J. P. C., and Duque, C. G. (2020). From feature engineering and topics models to enhanced prediction rates in phishing detection. *Ieee Access*, 8:76368–76385.

Hemberg, E., Kelly, J., Shlapentokh-Rothman, M., Reinstadler, B., Xu, K., Rutar, N., and O'Reilly, U.-M. (2020). Linking threat tactics, techniques, and patterns with defensive weaknesses, vulnerabilities and affected platform configurations for cyber hunting. *arXiv preprint arXiv:2010.00533*.

Hu, Z., Beuran, R., and Tan, Y. (2020). Automated penetration testing using deep reinforcement learning. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 2–10. IEEE.

Hwang, C.-L. and Yoon, K. (1981). Multiple attribute decision making: a state of the art survey. *Lecture Notes in Economics and Mathematical Systems*, 186(1).

Kanakogi, K., Washizaki, H., Fukazawa, Y., Ogata, S., Okubo, T., Kato, T., Kanuka, H., Hazeyama, A., and Yoshioka, N. (2021a). Tracing capec attack patterns

from cve vulnerability information using natural language processing technique. In *Proceedings of the 54th Hawaii International Conference on System Sciences*, page 6996.

Kanakogi, K., Washizaki, H., Fukazawa, Y., Ogata, S., Okubo, T., Kato, T., Kanuka, H., Hazeyama, A., and Yoshioka, N. (2021b). Tracing cve vulnerability information to capec attack patterns using natural language processing techniques. *Information*, 12(8):298.

Khan, N. A., Brohi, S. N., and Zaman, N. (2020). Ten deadly cyber security threats amid covid-19 pandemic.

Kholidy, H. A. (2022). Multi-layer attack graph analysis in the 5g edge network using a dynamic hexagonal fuzzy method. *Sensors*, 22(1):9.

Liu, N., Zhang, J., Zhang, H., and Liu, W. (2010). Security assessment for communication networks of power control systems using attack graph and mcdm. *IEEE Transactions on Power Delivery*, 25(3):1492–1500.

Liu, X., Qian, C., Hatcher, W. G., Xu, H., Liao, W., and Yu, W. (2019). Secure internet of things (iot)-based smart-world critical infrastructures: Survey, case study and research opportunities. *IEEE Access*, 7:79523–79544.

Mavroeidis, V. and Bromander, S. (2017). Cyber threat intelligence model: an evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. In *2017 European Intelligence and Security Informatics Conference (EISIC)*, pages 91–98. IEEE.

MITRE (2021). The MITRE Corporation. https://cve.mitre.org/. [Online; accessed 06-December-2021].

NIST (2021). National Vulerability Database (NVD). https://nvd.nist.gov/vuln/data-feeds. [Online; accessed 24-January-2022].

Opricovic, S. and Tzeng, G.-H. (2004). Compromise solution by mcdm methods: A comparative analysis of vikor and topsis. *European journal of operational research*, 156(2):445–455.

Pawlicka, A., Pawlicki, M., Kozik, R., and Choraś, R. S. (2021). A systematic review of recommender systems and their applications in cybersecurity. *Sensors*, 21(15):5248.

Pimenta Rodrigues, G. A., de Oliveira Albuquerque, R., Gomes de Deus, F. E., De Oliveira Júnior, G. A., García Villalba, L. J., Kim, T.-H., et al. (2017). Cybersecurity and network forensics: Analysis of malicious traffic towards a honeynet with deep packet inspection. *Applied Sciences*, 7(10):1082.

Polatidis, N. and Georgiadis, C. K. (2017). A dynamic multi-level collaborative filtering method for improved recommendations. *Computer Standards & Interfaces*, 51:14–21.

Polatidis, N., Pavlidis, M., and Mouratidis, H. (2018). Cyber-attack path discovery in a dynamic supply chain maritime risk management system. *Computer Standards & Interfaces*, 56:74–82.

Polatidis, N., Pimenidis, E., Pavlidis, M., and Mouratidis, H. (2017). Recommender systems meeting security: From product recommendation to cyber-attack prediction. In *International Conference on Engineering Applications of Neural Networks*, pages 508–519. Springer.

Polatidis, N., Pimenidis, E., Pavlidis, M., Papastergiou, S., and Mouratidis, H. (2020). From product recommendation to cyber-attack prediction: generating attack graphs and predicting future attacks. *Evolving Systems*, 11(3):479–490.

Rapid7 (2021). The Metasploitable 3 Linux VM. https://github.com/rapid7/metasploitable3. [Online; accessed 10-January-2022].

Rodriguez, L. G. A., Trazzi, J. S., Fossaluza, V., Campiolo, R., and Batista, D. M. (2018). Analysis of vulnerability disclosure delays from the national vulnerability database. In *Anais do I Workshop de Segurança Cibernética em Dispositivos Conectados*. SBC.

Rytel, M., Felkner, A., and Janiszewski, M. (2020). Towards a safer internet of things—a survey of iot vulnerability data sources. *Sensors*, 20(21):5969.

Security, O. (2021). The Exploit Database. https://www.exploit-db.com/. [Online; accessed 10-January-2022].

Shah, S. and Mehtre, B. M. (2015). An overview of vulnerability assessment and penetration testing techniques. *Journal of Computer Virology and Hacking Techniques*, 11(1):27–49.

Thamilarasu, G. and Chawla, S. (2019). Towards deep-learning-driven intrusion detection for the internet of things. *Sensors*, 19(9):1977.

Valea, O. and Oprişa, C. (2020). Towards pentesting automation using the metasploit framework. In *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 171–178. IEEE.

VULNERS, I. (2021). Vulners, inc. https://vulners.com/. [Online; accessed 10-January-2022].

Wang, J. A. and Guo, M. (2009). Ovm: an ontology for vulnerability management. In *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*, pages 1–4.

Yaqoob, I., Hussain, S. A., Mamoon, S., Naseer, N., Akram, J., and ur Rehman, A. (2017). Penetration testing and vulnerability assessment. *Journal of Network Communications and Emerging Technologies (JNCET) www. jncet. org*, 7(8).

Zhou, S., Liu, J., Hou, D., Zhong, X., and Zhang, Y. (2021). Autonomous penetration testing based on improved deep q-network. *Applied Sciences*, 11(19):8823.