I. J. Engineering and Manufacturing, **2022, 4, 1-14**
Published Online August 2022 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijem.2022.04.01

Comparison and Analysis of Software Vulnerability Databases

Hakan KEKÜL
University of Fırat, Institute of Science, Elazığ Turkey
Sivas Information Technology Technical High School, Diriliş Mahallesi Rüzgarli Sokak No 21 Sivas, Turkey.
E-mail: hakankekul@gmail.com

Burhan ERGEN
University of Fırat, Faculty of Engineering, Computer Engineering Department, Elazığ Turkey.
E-mail: bergen@firat.edu.tr

Halil ARSLAN
University of Sivas Cumhuriyet, Faculty of Engineering, Computer Engineering Department, Sivas Turkey.
E-mail: harslan@cumhuriyet.edu.tr

Received: 26 April 2022; Accepted: 27 May 2022; Published: 08 August 2022

Abstract: In order to protect information systems against threats and vulnerabilities, security breaches should be analyzed. In this case, analysts primarily conduct intelligence research through open source systems. In particular, vulnerability databases stand out as the most preferred references at this stage. At this point, our study will be the main reference for the verification of vulnerability analysis. It will assist in the planning of testing processes, patches and updates in the development of software. Moreover, it will create a perspective in this field, enabling readers to understand the concept of software security and databases. In addition to unique advantages of this diversity, this has also led to some disadvantages. Our study focused on the reasons behind the creation of different databases. In addition, its advantages and disadvantages have been clearly demonstrated. First, the databases used were determined by examining the academic studies in the field of software security vulnerabilities. Twelve different databases used in the literature were identified. However, among these, the ones that are current and accessible to researchers were selected. As a result of this screening process, seven different databases were included in this study. The determined databases were examined in detail and explained. Then, databases were compared according to certain criteria. The data obtained as a result of the comparison are presented in detail. In this study, a systematic review of up-to-date and accessible vulnerability databases that are widely used in the literature is presented to help researchers decide which database to use.

Index Terms: Software Security, Software Vulnerability, Vulnerability Databases, Information Securty, Cyber Security.

1. Introduction

Cyber security is one of the most important research areas in the IT and software world today. Software systems play a very important part in our modern daily life. In many cases, problems that may occur in the software system can cause unfavourable results. For this reason, it has become an important requirement to be able to determine software security. Building safer systems has emerged as one of the crucial objectives guiding all software engineering efforts in the recent years [1]. In this sense, it has led to the emergence of concepts such as error prediction, reusability, aging prediction, information security and software product line. Intensive scientific studies are carried out by considering these important research areas [2]. Many of these studies use vulnerability databases created by different research groups that include software vulnerabilities. Software vulnerabilities and databases containing them are determined, classified, categorized and scored manually by experts.

Vulnerability databases are platforms where detected security vulnerabilities are shared with the public. They contain all the data that can be accessed about a software-related vulnerability. There are different databases created by public or private organizations. Vulnerabilities are provided as security reports. When the contents of the reports are examined, the first thing to see is the identification number they have. In addition, it is seen that it consists of product information affected by the vulnerability, a description defining the vulnerability, dates, author information, solution suggestions, abuse codes and references. In addition, some databases include scoring information that provides information about the severity of the vulnerability [3].

Cybercrime, which emerges as the abuse of security vulnerabilities, can cause great harm. It is estimated that the damage caused by cybercrime will cost 6 trillion dollars in 2021 [4]. It is very important to determine the extent to which the detected vulnerabilities violate the security policy. There has been increasing concern about the abuse of security vulnerabilities lately. Therefore, it is necessary to classify the vulnerabilities without exploiting them [5]. The number of security vulnerabilities is increasing rapidly. Studies show that the large-size archive cannot be evaluated with statistical techniques and this problem is increasing. According to empirical results based on regular regression analysis of over eighty thousand archived vulnerabilities, the effort spent to calculate CVSS values has a statistically negative impact on time delays [6]. The inability to conduct a comprehensive analysis of the data in this dimension indicates the existence of unexplored points [7]. This problem is growing. It is important to keep accurate data in a specific configuration and make it accessible to researchers. In particular, security experts need access to information with accurate references [8].

Although there is a risk of misuse of published vulnerability data, it is an important resource for researchers of this field. At this point, many studies are carried out using security vulnerability data. However, these studies generally use a single database. Also, differences in databases have not yet been taken into account [9].

This study examines the data in different databases, suggests the differences of each database and guides the experts in the field in their studies. It reveals the consistency of data presented by databases and the importance of databases apart from generally accepted database providers. Henceforth, we will focus on the following research questions:

RQ1: What are the vulnerability databases open to researchers' access and widely used in the literature, and what types of data are used in the data collection and reporting processes of these databases?

We researched how software vulnerability has been detected and defined since the first study in the literature. We examined the definitions in different studies. Finally, we have created an understandable diagram of the official definition. In this way, we try to ensure that the terms that cause confusion in definitions are better-understood. In our study, 12 different databases that were found important in the literature were examined. The databases used in academic studies are especially preferred. These 12 databases were examined in detail. Two selection criteria have been determined to decide which databases will be included in our study and which ones will be excluded. These two criteria are up-to-dateness and its convenience to provide open access for researchers. Databases that meet these two basic selection criteria have been examined in our study. As a result of this selection, 7 different databases with no access problems and up-to-date data were selected. 5 of these 12 databases were excluded as they did not comply with the criteria.

RQ2: What are the main differences of databases and what advantages do they provide for researchers?

The databases included in the study were examined thoroughly. The basic differences are identified and they are comparatively specified. The indicated advantages of databases in the literature are given within the results of the research. In addition, the frequency of the databases used in researches has also been determined. In this way, the database trends and preferences of the researchers today could be understood better.

Çalışmanın diğer bölümleri şu şekilde organize edilmiştir. İkinci bölümde, çalışmaya yön veren literatür ayrıntılı olarak incelenmiş, üçüncü bölümde yazılım zafiyeti kavramı açıklanmış, dördüncü bölümde zafiyet veri tabanları tanıtılmış ve beşinci bölümde sistematik karşılaştırmaları sunulmuştur, altıncı bölümde bölümünde elde edilen sonuçların tartışılması yer almaktadır. Son bölümde ise çalışmanın sonuçları sunulmakta ve gelecekte yapılacak çalışmalar ifade edilmektedir.

2. Related Works

Studies that estimate and discover scores in addition to giving the analysis of software security vulnerabilities have been on the increase in recent years. First of all, software metrics used to determine software quality values are clearly described in the literature. Moreover, software vulnerability analysis is identified as a field of academic interest. Much as traditional approaches were applied in the first studies, the results were not satisfactory. The motivation behind the use of machine learning and data mining techniques in the problem of security vulnerabilities of software components lies behind the serious success of these algorithms in different problems. To that end, various studies have categorically been conducted using machine learning and data mining techniques in order to analyse and detect the problem of security vulnerability of software components [10].

Ghaffarian et al. [10], provide a comprehensive review of many different studies using machine learning and data mining techniques in the field of software vulnerability analysis and discovery. By examining different study categories in this field, they point out both the advantages and drawbacks, and also refer the challenges and some undiscovered layers in the field. The authors suggest carrying out feature engineering studies that can improve the performance of machine learning systems, and whose content is also rich in terms of engineering as well as being effective and having distinctive features on different software vulnerabilities. This study will give researchers a perspective on which database and data they will use.

Wu et al. [11], suggest an approach to create large-scale datasets for machine learning-based security error report prediction. In their related study, they created the initial version of the OpenStack dataset, which contains approximately 80 thousand error reports. As a result, they recommend developing a dataset building approach by including other methods (such as feature selection, deep learning) in order to improve the quality of the data sets. This study emphasizes the need for the creation of quality data sets. Our study will help researchers who will create a new dataset by comparing different vulnerability databases.

Williams et al. [7], states that the security vulnerability data accumulated over the years have become a large, unstructured data group. They emphasize that this situation is often undiscovered due to the lack of testing the tools and algorithms necessary to conduct a comprehensive analysis of the data. As a result of their study, they found out that there is a significant gap in vulnerability trends, transformations and interactions, and in general output sensitivity to vulnerabilities. Thus, they stated that understanding the important features of vulnerability data will provide significant benefits for researchers and field experts in developing secure systems in the future, reducing the problems arising from security vulnerabilities, and revealing new academic fields of study. Our study will contribute to the authors' insights into new research areas, helping researchers decide which database to use.

Fang et al. [12] in their study, they stated that only a small fraction of the vulnerabilities were exploited by attackers. For this reason, they emphasized the importance of distinguishing non-exploitable vulnerabilities from others in terms of efficient use of limited resources. It has been stated that the publication of the identified security vulnerabilities in the system takes time and the National Vulnerability Database (NVD) is insufficient due to the deficiencies arising from the institutional structure of the database, so databases created by different communities contain more efficient features.

Yang et al. [13] noted that about half of the software vulnerabilities were exploited within two weeks of the vulnerability was announced. In addition, it is stated that only 20% of the declared deficits were exposed to misuse. Therefore, the importance of accurately predicting the vulnerability scores and prioritizing them are highly emphasized.

Raducu et al. [14] emphasize that different machine learning techniques have emerged and developed to detect security vulnerabilities. However, they point out that the performance of these algorithms require data driven engines that rely on processing large amounts of data known as data sets.

As seen in the studies examined, with the reporting of software vulnerabilities, there has been an increasing interest in this field in the academic circles. Due to the importance of the field, studies are carried out with government support, especially in developed countries. After the successful application of machine learning algorithms in many problems, it is seen that they have been used in the solution to this particular problem since 2012 [10]. Recent studies suggest the use of machine learning-based approaches. However, as can be understood from the literature reviews, it is clear that in order to achieve high performance, it is necessary to use structured and extracted data sets in studies. [7,10,11,14,15]. The NVD data set, which dominates the field, cannot ensure this due to its structure. This situation was tried to be solved by the research circles by creating data sets with different features. These newly-created and different data sets have added newer and wider information into the same security vulnerabilities. The main problem of these data sets is that they are created in natural language and structures that only the experts can understand themselves. It is not suitable to use machine learning algorithms directly.

3. The Concept of Software Vulnerability

Many researchers have defined the concept of software vulnerability. However, there is no standard definition of this term. For this reason, it remains a challenge to clearly discern the situations that can be categorized as the concept of vulnerability [16]. At this point, we accept the definition by The Institute of Electrical and Electronics Engineers (IEEE). While defining the concept of software vulnerability, it is seen that the definitions of Krsul and Ozmen are acknowledged when IEEE Standard Software Engineering Terminology Dictionary is taken into consideration [12]. It has been defined by Krsul [17] as "an example of an error in the description, development or configuration of the software, which may violate the security policy in the operation of the software", and by Ozment as "A software vulnerability is an example of a mistake made in the technical specifications, development, or configuration of the software to violate the implicit or explicit security policy of the application" [18]. The main difference between these two definitions is that the word “error” has been replaced with the word “false”. These definitions are acknowledged in the IEEE Standard Software Engineering Terminology Dictionary [19].

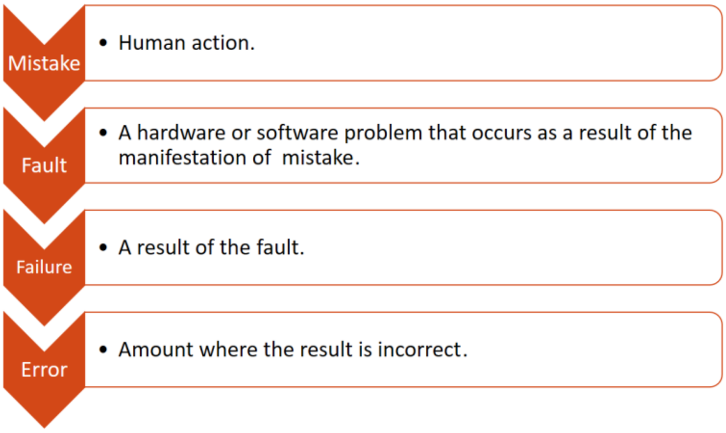
Mistake
• Human action.
Fault
• A hardware or software problem that occurs as a result of the manifestation of mistake.
Failure
• A result of the fault.
Error
• Amount where the result is incorrect.

Fig. 1. Terms used to describe software vulnerabilities and their relationships

When IEEE Software Engineering Terminology Dictionary is examined, it is understood that the four key terms in Figure-1 are important. The summary of the relationship between these terms is as follows; “Mistake is a human action”, “Fault is a hardware or software problem that occurs as a result of the manifestation of this mistake”, “Failure is the result of the Fault”, “Error refers to the amount where the result is incorrect” [19].

Based on these definitions, it is stated that the appropriate key term to be used in a software vulnerability definition could be the word “fault” [10].

According to the generally accepted definition, software vulnerability is defined as follows; "A software vulnerability is an example of a flaw caused by an error in the design, development, or configuration of the software in a way that can be used to violate some explicit or implicit security policy" [10]. Figure 1 provides a useful perspective for making sense of vulnerability terms.

Considering this definition, it is understood that it is not possible for any software component to be free from any fault (bugs). Therefore, in the event that a fault violating the security protocols of software products is detected, it has become a routine process to report and disclose it. It is important for researchers to examine the basic database of these reports Common Vulnerabilities and Exposures (CVE) and other databases developed on them. Figure 2 marks the focus of our current research. As vulnerabilities that could not be detected or reported could not be observed, research has often focused on disclosed vulnerabilities.

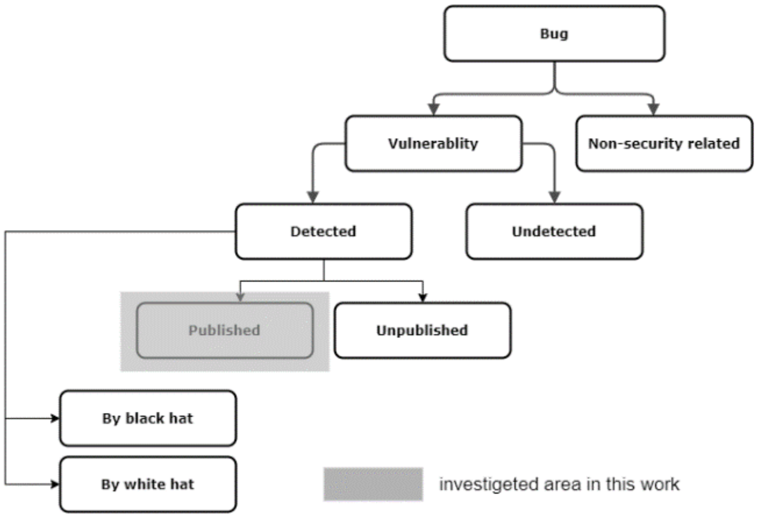

Fig. 2. Classification of software faults and vulnerabilities [20]

4. Software Vulnerability Databases

Security vulnerabilities pose an important risk in today's society which is equipped with information technologies. The importance of this risk has recently begun to be recognized [21]. For this reason, effective information sharing and coordination among the stakeholders of the subject is essential. Important problems can be avoided by taking preventive measures. This is why handling software vulnerabilities is very important [22]. It is important to know what data is collected and how it is reported. We will try to find an answer to RQ1 in this section.

Individuals, companies and other organizations that find that a flaw in any software violating security policies will inform about this by filling out a security report. This notification can be made through any vulnerability database provider. However, when a vulnerability is detected, the procedure, which is an international standard and funded by the US Department of Homeland Security, is implemented to officially declare it. The institution authorized for this procedure is the non-profit MITRE company [23]. Security vulnerabilities are registered into the Common Vulnerabilities and Exposures (CVE) database by many of the the Computer Emergency Response Teams (CERT) of the countries which are members of this organization so that the official process can begin.

A. CVE - Common Vulnerabilities and Exposures

This list was established in 1999 with the large security organizations brought together by MITER to bring an international standard to the detected security vulnerabilities. It is a descriptive list for commonly known cyber security vulnerabilities. The use of CVE entries provides a unique trust in the security of software with international reliability [24]. CVE entries are used by many emprical studies and by many product and service providers on information security such as Adobe, Apple, IBM or Microsoft [20,25].

The CVE is basically a list of vulnerabilities aiming at providing a descriptive and standardized definition for a vulnerability as well as providing public and free data approved by the industry. Its main mission is to establish the same standards for different databases and tools, to improve interoperability and the security coverage of the IT ecosystem. It describes itself as an identifier rather than a database. All large databases that are known today are basically created on the basis of the lists published in the CVE [24]. In spite of providing standard and reliable information, it contains raw information and additional information is not included in the lists. The features and descriptions in the CVE database security report are presented in Table 1.

Table 1. CVE Database Features

Feature	Description
CVE-ID	Unique ID assigned by CVE
Description	Technical expert opinion on the vulnerability
References	External links with information about the vulnerability
Assigning CNA	Notifying authority or Author information
Date Entry Created	Date the entry was created

B. NVD - National Vulnerability Database

It was created in 2000 under the National Institute of Standards and Technology (NIST). It is a database that includes the management, scoring and compliance of vulnerability data. NVD contains information on security checklist references, security-related software defects, misconfigurations, product names, and impact metrics. Backed by the National Cyber Security Division of the US Department of Homeland Security[26].

The main task of NVD employees is to analyze the vulnerability lists published in the CVE entries. They make use of all the explanations and references in the CVE and all the additional data they are able to collect. Associated impact metrics (Common Vulnerability Scoring System - CVSS), types of vulnerability (Common Weakness Enumeration - CWE) and applicability statements (Common Platform Enumeration - CPE) and other relevant metadata are all added into the data published by the NVD database. However, NVD does not perform vulnerability testing for the entity properties it assignes. According to new findings, the CVSS scores and applicability expressions of the data may be modified [26]. Field experts state the popular critisim that descriptions of the data in the database cannot clearly identify the vulnerabilities [12]. The features and descriptions in the NVD database security report are given in Table 2.

Table 2. NVD Database Features

Feature	Description
CVE-ID	Unique ID assigned by CVE
Current Description	The current explanation of the vulnerability
Analysis Description	Post-analysis vulnerability disclosure
References to Advisories, Solutions, Tools	Recommended external links with information about the vulnerability, solution methods and tools
Severity	CVSS V.3.X CVSS V2.0 scores and vulnerability vectors.
Weakness Enumeration	Category, number, and source of the Vulnerability. (CWE-ID - CWE Name, Source)
Known Affected Software Configurations	Software and versions known to be affected
Change History	Historical background of major deficit activities

C. Exploit-DB

It was created by the Offensive Security community in 2004 as a public service and non-profit project. It is compatible with the lists published in the CVE dictionary. Its main purpose is to serve the most comprehensive archive of exploits and present them in a freely accessible and easy-to-navigate database. The Exploit database provides PoC codes (Proof of Concept Code) that show the exploitability of vulnerabilities published in CVE lists rather than their definitions. PoC is a simple piece of code that explains how an attacker can exploit the vulnerability. This feature makes the database a valuable resource for those who need instantly actionable data. However, data without a PoC code is ignored [27]. The features and descriptions in the Exploit-DB database security report are presented in Table 3.

Table 3. Exploit-DB Database Features

Feature	Description
CVE-ID	Unique ID assigned by CVE
Exploit Title	Affected software name and vulnerability type
Exploit Author	Notifying authority or Author information
Date	Date the entry was created
Version	Software and versions known to be affected
Poc Code	Abuse Code
Vendor Homepage	External links including security vulnerabilities
Tested On	Operating system that the vulnerability is tested on
Author Contact	Contact information of the authority or author who reported the vulnerability

D. SecurityFocus

It was founded in 1999 by a community formed by independent security experts. SecurityFocus Vulnerability Database is based on the CVE lists and aims to provide the most up-to-date information about vulnerabilities on all platforms and services for security professionals. It publishes newsletters, technical articles and essays. Their mailing lists allow to discuss security issues with its members around the world [28]. SecurityFocus is one of the most important and most respected vulnerability databases. According to the descriptions in the NVD database, the descriptions in the SecurityFocus lists explain the impact and exploitability of the vulnerability more specifically [12]. The features and descriptions in the SecurityFocus database security report are given in Table 4.

Table 4. SecurityFocus Database Properties

Feature	Description
CVE-ID	Unique ID assigned by CVE
BUGTRAQ ID	ID defined by SecurityFocus
Info	Vulnerability general information
Discussion	Detailed information about the vulnerability
References	External links with information about the vulnerability
Solution	Solution suggestions for the deficit
Exploit	Abuse Code
Dates	Dates of Publication and Updates
Credit	Notifying authority or author information
Vulnerable	Software and versions known to be affected
Class	Name of the Vulnerability category

E. Rapid7

Rapid7, a security company providing united security management solutions, was founded in 2000. It is a database containing technical details for vulnerability and exploitation for security professionals and researchers to review. It is compatible with CVE lists. All exploit codes published in this database are included in the Metasploit (commercial penetration testing framework) framework. The database of vulnerability and exploits is frequently updated, and contains the most recent security researches. As a public policy, it has adopted working with governments, companies, non-profit organizations and experts to shape policies, standards and legislation that are also beneficial for consumers and advocate responsible cybersecurity professionals. [29]. The features and explanations in the Rapid7 database security report are given in Table 5.

Table 5. Rapid7 Database Features

Feature	Description
CVE-ID	Unique ID assigned by CVE
Title	Affected software name and vulnerability type
Description	Technical expert opinion on the vulnerability
References	External links with information about the vulnerability
Solution(s)	Solution suggestions for the deficit
Severity	CVSS V2.0 scores and vulnerability vector
Dates	Published, Created, Added and Modified dates

F. Snyk

The Snyk database was established by a commercial company that provides free code evaluation tools for open source projects. It has taken it as their mission to support the development of open source projects and help them keep safe. Snyk helps protect more than 25,000 applications by monitoring vulnerabilities in more than 800,000 open source packages. 83% of Snyk users stated that they found security vulnerabilities in their applications. New vulnerabilities are regularly disclosed. Snyk database is structured on four basic principles. These are to find, to fix, to prevent and to monitor the vulnerability constantly [30]. The features and descriptions in the Snyk database security report are given in Table 6.

Table 6. Snyk Database Features

Feature	Description
CVE-ID	Unique ID assigned by CVE
SNYK ID	ID defined by Snyk company
Title	Affected software name and vulnerability type
Overview	Vulnerability summary
Details	Detailed information about the vulnerability
References	External links with information about the vulnerability
Remediation	Solution suggestions for the deficit
Severity	CVSS V3.1 scores and vulnerability vector
CWE	Category number of the vulnerability
Dates	Disclosure and publication dates
Credit	Notifying authority or author information

G. SARD – Software Assurance Reference Dataset Project

The collection started in 2005 by the National Institute of Standards and Technology (NIST). When it was first announced, it was initially named as the Standard Reference Dataset (SRD). This name was changed to the Software Assurance Reference Data Set (SARD) in 2014. It aims to help users, researchers, and software developers to improve security tools by providing a range of common vulnerabilities. In addition, it provides an archive that includes all stages of the software life cycle by providing data such as test scenario designs, source codes, and binary files. This enables end users to test and evaluate the tools and tool development methods they have developed. The dataset includes "real" (production), "artificial" (written to test) and "academic" (from students) test cases. This database also contains a real software application with known bugs and vulnerabilities. The dataset covers a wide variety of potential vulnerabilities, languages, platforms and compilers. The dataset grows by collecting test cases from many participants [31]. The features and description in the SARD database security report are given in Table 7.

Table 7. SARD Database Features

Feature	Description
Test Case ID(up)	Unique ID assigned by SARD
Description	Detailed information about the vulnerability
Language	Supported Programming Language
Type of Artifact	Method of Generating Test Code
Status	Status information
Weakness CWE	Category number of the vulnerability
Submission Date	Release date

5. Comparison of the Databases

Vulnerability databases often have similar characteristics. When these characteristics are taken into consideration, their common interests are numbering, description, author information, references, severity score, solution methods, exploit codes, vulnerability category and date information. However, as can be seen in the details in Table 8, each database provider possesses different features that are unique to them besides these features. In addition, it is seen that the values included in the features show significant differences.

The databases whose definitions, advantages and disadvantages were presented above and included in the study were compared with a systematic comparison method. In the comparison of the databases examined within the scope of this study, the evaluation was made according to the following criteria; whether there is a vulnerability scoring, whether it includes a solution method for the vulnerability, whether there are exploit codes, whether the vulnerabilities have been tested, and who reported them, the presence of reference information, whether the author information is provided, whether it contains information about the category of the vulnerability, whether supports data feed technologies, business model and data size. In Table 9, the selected databases and their comparisons according to the determined evaluation criteria are given. In this section, we will try to find an answer to AS2.

Table 8. Matching Security Report Information of Vulnerability Databases

Database	**Numbering**	**Description**	**Referans**	**Author**	**Dates**	**CVSS Version**	**Solution**	**Exploit Code**	**Affected Software**	**CWE**	**Different Fields**
CVE	CVE-ID	Description	References	Assigning CNA	Date Entry Created						Phase, Votes, Comments, Proposed
NVD	CVE-ID	Current Desc., Analysis Desc.	References to Advisories		Change History	3.x,2.0	Solutions	References to Advisories	Known Affected Software Configurations	Weakness Enumeration CWE-ID CWE Name Source	Tools
Exploit-DB	CVE-ID	Vuln. Details		Exploit Author, Vendor Homepage, Author Contact	Date			PoC Code	Exploit Title, Version		Tested On

Rapid7	CVE-ID	Description	References		Published, Created, Added, Modified	2.0	Solution		Title		
Snyk	CVE-ID, SNYK ID	Overview, Details	References	Credit	Dısclosed, Publıshed	3.1	Remediation	Details	Title	CWE	
Security Focus	CVE-ID, Bugtraq ID	Info, Discussion	References	Credit	Dates		Solution	Exploit	Vulnerable	Class	Remote, Local, Not Vulnerable
SARD	Test Case ID	Description			Sub. Date					Weakness CWE	Type of Artifact, Language, Status

When a security report is issued, there happens an ID assignment. The CVE-ID value given by CVE is usually used. However, some databases provide their own ID values along with the CVE-ID. SARD database prefers to use only its own TEST CASE ID value. This is because the SARD database has a completely different list of content from other providers. In addition, SecurityFocus and Snyk databases use their own numbering systems together with the CVE-ID value. These values are Bugtrag-ID in SecurityFocus database and Snyk-ID in Snyk.

Another common feature of the databases is that they all include description and date in their reports. The only exception is that some of Exploit-DB's reports do not include disclosure information. In addition, although the content of this information is specific to each database, it can be expressed with different titles. In addition, all databases except SARD databases provide reference information in their reports. However, there are no references in some of the information provided by the Exploit-DB database.

Security scores are used to express the impact value of vulnerabilities. When Table 9 is examined, it is seen that the score information can be obtained from the databases of NVD, Rapid7 and Snyk. However, NVD is the only database in which CVSS 2.0 and CVSS 3.x values from the official scoring versions are given together. Only the values of version 2.0 are given in the Rapid7 database. In the Snyk database, only 3.1 version has score values.

Table 9. Comparison of Vulnerability Databases

Databse	**Numbering**	**Description**	**Referans**	**Author**	**CVSS Version**	**Solution**	**Exploit Code**	**Test**	**CWE**	**Json / XML/RSS**	**Reports**	**Business Model**	**Data Size***
CVE	CVE-ID	✓	✓	(✓)	×	×	×	×	×	✓	Everyone	Public	139.407
NVD	CVE-ID	✓	✓	(✓)	2.0-3.X	✓	(✓)	×	✓	✓	Members	Public	147.510
Exploit-DB	CVE-ID	✓	×	✓	×	×	✓	✓	×	×	Everyone	Public	42.962
SecurityFocus	CVE/BUGTRAG ID	✓	✓	✓	×	✓	(✓)	✓	×	×	Everyone	Public	102.330
Rapid7	CVE-ID	✓	✓	×	2.0	✓	×	×	×	×	Emp.	Com.	171.816
Snyk	CVE/SNYK ID	✓	✓	✓	3.1	✓	×	×	✓	×	Emp.	Com.	6.012
SARD	TEST CASE ID	✓	×	×	×	×	×	✓	✓	✓	Everyone	Public	177.184

✓: There is - ×: None - (✓): For some data there is for some not [9].

* data sizes are the values as of 31.08.2021 and it continues to increase [9].

There are some risks of making vulnerabilities available to public. It is important to offer the method together with the vulnerability in order to avoid from exploitation of this public use. Information on how to resolve security vulnerabilities can be found on SecurityFocus, NVD, Rapid7 and Snyk databases. Exploit-DB is the only database that regularly contains PoC codes, which is a proof of the exploitability of the vulnerability. Despite not being included for each data, PoC codes can be found on NVD, Snyk and SecurityFocus databases.

When a vulnerability is detected, it is evaluated by experts. At this point, it is important to decide whether to conduct verification tests or not. All databases except Rapid7 and SARD databases provide the information of the notifying author. However, only Exploit-DB, SecurityFocus and SARD databases perform tests on data.

NVD, which accepts only the reports of its members, differs from other databases in terms of reporting security vulnerabilities in the system. In addition, commercial databases Rapid7 and Snyk have their employees do the reporting. Other databases are open to everyone in reporting a security vulnerability.

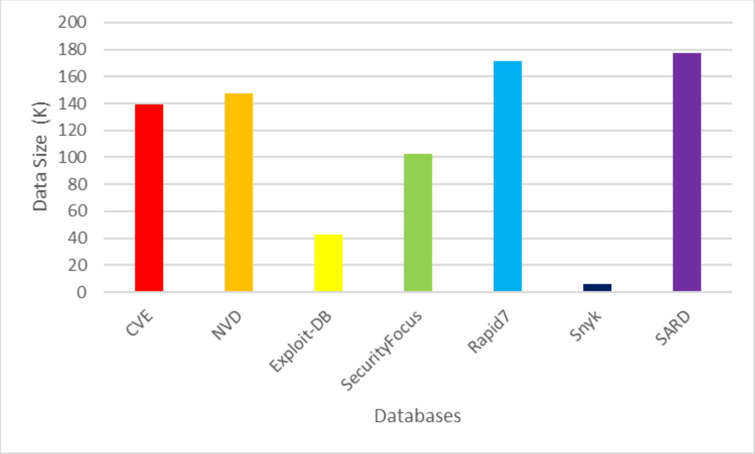

Fig. 3. Data Dimensions

Considering their business model criterion, it is seen that all databases are divided into two categories as being public and commercial. While the public business model refers to working for the public benefit without making any profit, the purpose of the commercial business model is to make profit with the tools it offers, but also to make a certain amount of data available to researchers free of charge. Rapid7 and Snyk databases have adopted the Commercial business model.

Considering the data size, Rapid7 provides a high volume of data free of charge, while Snyk provides the least amount of data. In Figure 3, the data dimensions of the databases can be found. It can be noticed that the databases basically based on CVE lists provide data sets of different sizes arising from their interpretation and evaluation methodologies.

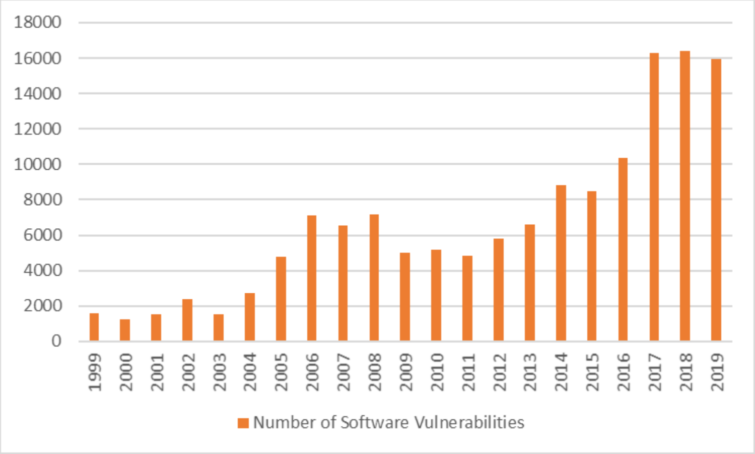

Fig. 4. Distribution of NVD database data by years

With respect to downloading lists of databases, except CVE and SARD, the other databases do not have such a service, as the former provides JSON data feeds and the latte has a batch download feature. In order to download all of the data from these databases, it is necessary to scan websites using regular expressions.

The processing of growing data in vulnerability lists is a manual process carried out by experts. [32]. The number of security vulnerabilities is increasing rapidly, as can be seen clearly in Figures 4 and 5. When NVD database data is examined, especially as seen in Figure 4, the rate of increase in vulnerability reporting increases by approximately 60% annually after 2016. In addition, when the graph in Figure 5 is examined, it is understood that the number of total deficits will continue to increase.

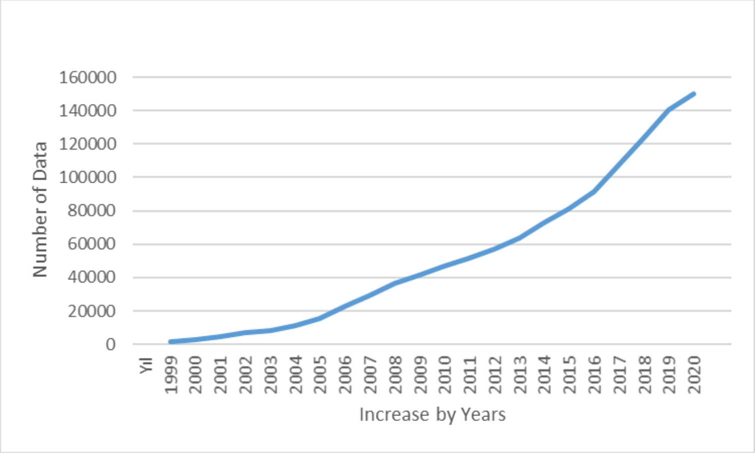

Fig. 5. Annual growth graph of NVD database data

Whether a published vulnerability seriously violates the security policy of the relevant product should be determined quickly. In addition, it needs to be removed immediately before being used for abuse. Recent studies confirm today's concerns about exploitation of vulnerabilities [5]. Financial damage of cybercrimes is estimated to cost $ 6 trillion by 2021 [4]. When a security vulnerability is found, it is published in public databases. However, these published reports are prepared in natural language and cannot be interpreted automatically by machines [33].

Table 10. Search terms.

Id	Terms
1	"common vulnerabilities exposures" or "CVE"
2	"national vulnerability database" or "NVD"
3	“exploit-DB” or “exploitdb” or “exploit db”
4	“SecurityFocus” or “Security Focus”
5	“Rapid7”
6	“Synk”
7	"Standard Reference Dataset" or "SRD" or "Software Assurance Reference Dataset" or "SARD"

Due to the unavoidable increase in the number of published vulnerabilities, it is observed that the size of the archive material has begun to lose its statistical significance for applied researches. This seems to continue to increase more in the coming years [6]. This growth, which has been accumulated for years and turned into a large unstructured dataset, can only be solved with high calculations and machine learning algorithms that have been successfully applied in many problems. Due to the shortcomings in this area, comprehensive analysis of the data cannot be made and different algorithms cannot be tested. This means that there are mostly undiscovered spots [7].

Table 11. Summary of search results per publishers.

Publisher	CVE	NVD	ExploitDB	SecurityFocus	Rapid7	Synk	SARD
ACM	232	315	192	25	32	2	0
Science Direct	263	176	4	48	88	27	6
IEEE	25	40	2	1	4	0	0
Scopus	2299	998	6	6	9	7	14
Springer	582	457	22	207	97	37	1
Web Of Science	1103	430	2	4	4	4	11
John Wiley & Sons	168	80	2	14	37	14	2
Others	91	1714	1268	2402	1614	2232	154
Total	4763	4210	1498	2707	1885	2323	188

Table 10 shows the list of search terms we use to investigate the use of the databases we have evaluated in academic publications. In addition, the values per publishers of academic studies conducted in the last ten years are given in Table 11.

Figure 7 shows the use of databases examined within the scope of the study in academic studies in the last decade. When closely examined, the intensity of usage continues at a certain rate until 2016. However, there has been a serious increase in studies since 2016. We think that the reason for this increase is related to the use of machine learning algorithms in the vulnerability problems. Ghaffarian et al.'s [10] studies on this subject confirm our opinion. In addition, as seen in Figure 6, studies in some databases (rapid7, synk, securityfocus, sard) tend to decrease gradually. We think that the reason for the situation above is due to the reporting structures of the databases. Theisen [7] and Ruohonen [6] also support our view on this subject in their studies. When Figures 6 and 7 are examined, another important point is that the NVD and CVE databases are clearly used more than others. However, although they form the basis for their institutional structures and vulnerability ecosystem, the use of other databases as an alternative has been on the increase recently. This is due to the fact that the features included in CVE and NVD reports are not sufficient [7], [12]. At this stage, it is obvious that the need to report and archive especially the accumulated and continuously growing data with a new structure [10,32].

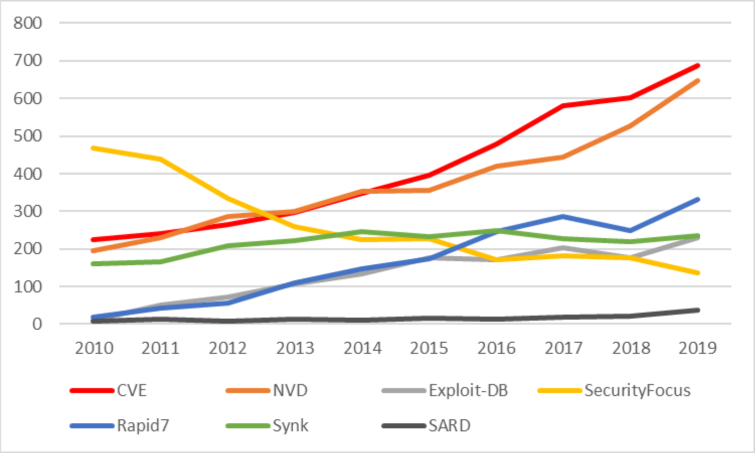

Fig. 6. Frequency of use of databases in academic studies by years

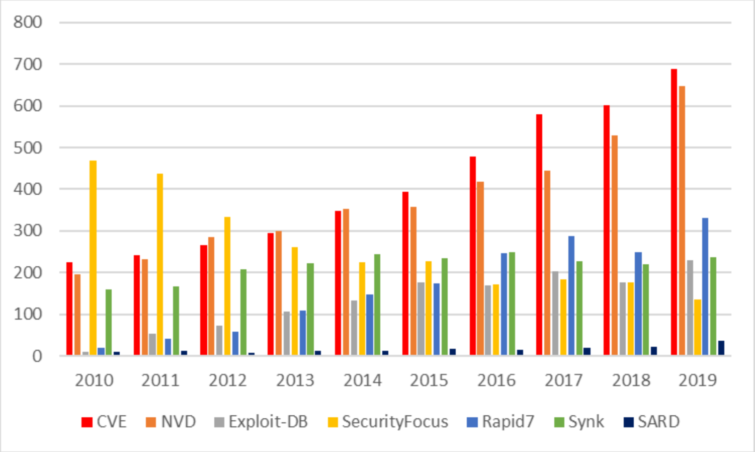

Fig. 7. Frequency of use of databases in academic studies by years

6. Discussion

Software vulnerabilities pose a major threat in modern society, where the majority of people's daily operations are carried out with the support of information systems. According to NVD database statistics, there is a 100% increase in software vulnerabilities since 2016. This huge increase in software vulnerabilities has caused this issue to become an important research area for the cyber security community. This growing problem of security vulnerability leads researches to anticipate them.

Common Vulnerabilities and Exposures (CVE) technical reports published in the NVD are a natural language set of all vulnerabilities that have been detected since 1988. With this aspect, it is predicted that people, not machines, will understand and interpret vulnerabilities. However, due to the insufficient information in the NVD dataset derived from the CVE dictionary, there are different datasets created for CVE data by different communities. However, these databases are also written in natural language, although they usually add new features to security vulnerabilities. Nevertheless, these datasets are basically based on CVE technical reports such as NVD and bring new features and explanations to the same vulnerabilities. For example, a SecurityFocus database was created by experts who thought that the descriptions in the NVD were insufficient. Likewise, the ExploitDB database, which contains the PoC codes of security vulnerabilities, is a data set with an exploit code. There are different open source databases with different features. However, as given above, all of them are databases prepared in natural language that people can only understand. It is not possible to use them directly in machine learning algorithms.

As a result, the CVE dictionary provides a common reliable and standard platform. Other databases update their data sets using CVE lists. NVD adds expert opinions and new features and explanations to the raw data in these lists. However, the adequacy of these added features and explanations is a matter of debate. Due to this problem, expert circles in the field have created new data sets by adding more understandable and useful features to these lists. SecurityFocus and ExploitDB are the foremost databases. The main differences of these databases from NVD are their structures that contain more understandable descriptions and exploitation codes. Commercial databases such as Rapid7 and Snyk also legally evaluate the commercial value of security vulnerabilities. In addition, this enables the development of commercial security frameworks and supports the industry for more secure software products. The security test scenarios provided by the SARD database are the most distinctive feature of this database. Thus, this makes important contributions to the development of software test engineering. Each database in the field stands out with a different feature.

7. Threats to Validity

We consider limiting our research only with seven large databases as the most important threat to the validity of our study. There are other databases that we excluded from the study because they do not meet our selection criteria. There may be some points that we overlook in terms of the data we examine. It should be kept in mind that all database information is extracted from official websites. Information published or updated after our review are disregarded. However, academic studies have attempted to verify these data crosswise.

8. Conclusion

The discovery and analysis of software vulnerabilities is an important issue. However, the industry and affected users should be informed about a software security vulnerability in fast and secure ways. Software security vulnerabilities are trying to respond to this need thanks to the reports they publish. Software vulnerability databases have become a large pile of unstructured data that has accumulated over a long period of time. In addition, published vulnerability reports also contain database-specific features. With this study, it has been tried to perform a comprehensive analysis of this problematic area for both sectoral and academic research.

As a result of the analyzes made, seven databases were selected that fit your criteria of being open to access and up-to-date. These databases are NVD, CVE, Exploit-DB, SecurityFocus, Rapid7, Snyk and SARD. It has been tried to access all the data that can be accessed for these databases and this study, which is an original research that fills an important gap in its field, is presented. The most important contribution of this study is that it offers a wide perspective for those working in the field of software security and can be a reference in the field. The analyzes made show that each database has its own advantages. CVE and NVD databases offer the most reliable data in the industry as a result of their institutional structure and government support. However, the technical explanations they offer are found weak by the researchers. The SecurityFocus database closes this gap and stands out with its more specific technical explanations. SARD database makes an important contribution to software test engineering with its test scenarios. In addition, databases such as Rapid7 and Snyk reveal the commercial potential of software vulnerabilities.

In the second part of our study, the use of software vulnerability databases in academic studies was analyzed. At this stage, many publishers were scanned with certain keywords. Analyzes were performed by grouping the results according to years and software vulnerability databases. It has been determined that software vulnerability databases have been used in 17,574 studies in the last ten years. When the duplicate publications are combined, the number of single publications detected is 8099. CVE has become the most used database in academic publications with 4763 studies. The second-order NVD database was used in 4210 studies. Other studies most frequently used in academic research were SecurityFocus 2707, Snyk 2323, Rapid7 1885 and ExploitDB 1498 times, respectively. The database included the least in academic studies was SARD, which was used in academic studies 188 times.

In addition to the advantages of using CVE and NVD mostly in the existing structures of the databases, it brings some disadvantages. Although there are delays in the publication of security reports due to their institutional structure and use of human resources, there are situations such as underestimating the values of severity scores than they should have been in the initial calculations. Considering that most of the abuses occurred in the first two weeks of publication of the deficit, its importance can be better understood. Calculation of severity scores by using machine learning algorithms will guide experts to make estimations. In addition, designing reports by all database providers in the system using data feeding technologies and relational databases will increase the opportunity for researchers and other database providers to work together and reduce the error rate.

9. Future Works

In our future studies, we plan to conduct studies using machine learning algorithms to calculate vulnerability vectors and severity scores. We also plan to research and develop word frequency lists. In addition, despite the availability of a large database, we aim to create a single and comprehensive processed and structured database from the large data that cannot be directly used in machine learning and deep learning algorithms. This database will be made available to researchers with the open source principle.

CRediT authorship contribution statement

Hakan KEKÜL: Conceptualization, Methodology, Validation, Formal analysis, Resources, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization. **Burhan ERGEN:** Conceptualization, Methodology, Validation, Formal analysis, Writing - Review & Editing, Supervision, Project administration. **Halil ARSLAN:** Conceptualization, Methodology, Validation, Formal analysis, Writing - Original Draft, Writing - Review & Editing, Supervision

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding Sources

This study is supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) with project number 121E298.

References

[1] H. Kekül, B. Ergen, H. Arslan, A multiclass hybrid approach to estimating software vulnerability vectors and severity score, J. Inf. Secur. Appl. 63 (2021) 103028. https://doi.org/https://doi.org/10.1016/j.jisa.2021.103028.

[2] H. Kekül, B. Ergen, H. Arslan, A New Vulnerability Reporting Framework for Software Vulnerability Databases, Int. J. Educ. Manag. Eng. 11 (2021) 11–19. https://doi.org/10.5815/ijeme.2021.03.02.

[3] S. Zhang, X. Ou, D. Caragea, Predicting Cyber Risks through National Vulnerability Database, Inf. Secur. J. A Glob. Perspect. 24 (2015) 194–206. https://doi.org/10.1080/19393555.2015.1111961.

[4] L.P. Kobek, The State of Cybersecurity in Mexico: An Overview, Wilson Centre's Mex. Institute, Jan. (2017).

[5] T.W. Moore, C.W. Probst, K. Rannenberg, M. van Eeten, Assessing ICT Security Risks in Socio-Technical Systems (Dagstuhl Seminar 16461), Dagstuhl Reports. 6 (2017) 63–89. https://doi.org/10.4230/DagRep.6.11.63.

[6] J. Ruohonen, A look at the time delays in CVSS vulnerability scoring, Appl. Comput. Informatics. 15 (2019) 129–135. https://doi.org/10.1016/j.aci.2017.12.002.

[7] C. Theisen, L. Williams, Better together: Comparing vulnerability prediction models, Inf. Softw. Technol. 119 (2020). https://doi.org/10.1016/j.infsof.2019.106204.

[8] C.W. Samuel Ndichu, Sylvester McOyowo, Henry Okoyo, A Remote Access Security Model based on Vulnerability Management, Int. J. Inf. Technol. Comput. Sci. 12 (2020) 38–51. https://doi.org/10.5815/ijitcs.2020.05.03.

[9] H. Kekül, B. Ergen, H. Arslan, Yazılım Güvenlik Açığı Veri Tabanları, Avrupa Bilim ve Teknol. Derg. (2021) 1008–1012.

[10] S.M. Ghaffarian, H.R. Shahriari, Software vulnerability analysis and discovery using machine-learning and data-mining techniques: A survey, ACM Comput. Surv. 50 (2017). https://doi.org/10.1145/3092566.

[11] X. Wu, W. Zheng, X. Chen, F. Wang, D. Mu, CVE-assisted large-scale security bug report dataset construction method, J. Syst. Softw. 160 (2020) 110456. https://doi.org/10.1016/j.jss.2019.110456.

[12] Y. Fang, Y. Liu, C. Huang, L. Liu, Fastembed: Predicting vulnerability exploitation possibility based on ensemble machine learning algorithm, PLoS One. 15 (2020) 1–28. https://doi.org/10.1371/journal.pone.0228439.

[13] H. Yang, S. Park, K. Yim, M. Lee, Better not to use vulnerability's reference for exploitability prediction, Appl. Sci. 10 (2020). https://doi.org/10.3390/app10072555.

[14] R. Raducu, G. Esteban, F.J.R. Lera, C. Fernández, Collecting vulnerable source code from open-source repositories for dataset generation, Appl. Sci. 10 (2020). https://doi.org/10.3390/app10041270.

[15] D. Miyamoto, Y. Yamamoto, M. Nakayama, Text-mining approach for estimating vulnerability score, Proc. - 2015 4th Int. Work. Build. Anal. Datasets Gather. Exp. Returns Secur. BADGERS 2015. (2017) 67–73. https://doi.org/10.1109/BADGERS.2015.12.

[16] E. Yasasin, J. Prester, G. Wagner, G. Schryen, Forecasting IT security vulnerabilities – An empirical analysis, Comput. Secur. 88 (2020) 101610. https://doi.org/10.1016/j.cose.2019.101610.

[17] I.V. Krsul, Software vulnerability analysis, Purdue University, 1998.

[18] A. Ozment, Improving vulnerability discovery models: Problems with definitions and assumptions, Proc. ACM Conf. Comput. Commun. Secur. (2007) 6–11. https://doi.org/10.1145/1314257.1314261.

[19] I.S.C. Committee, others, IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990). Los Alamitos, CA IEEE Comput. Soc. 169 (1990).

[20] G. Schryen, Security of open source and closed source software: An empirical comparison of published vulnerabilities, AMCIS 2009 Proc. (2009) 387.

[21] A. Kuehn, M. Mueller, Shifts in the cybersecurity paradigm: Zero-day exploits, discourse, and emerging institutions, in: Proc. 2014 New Secur. Paradig. Work., 2014: pp. 63–68.

[22] O. Bozoklu, C.Z. Çil, Yazılım Güvenlik Açığı Ekosistemi Ve Türkiye’deki Durum Değerlendirmesi, Uluslararası Bilgi Güvenliği Mühendisliği Derg. 3 (2017) 6–26.

[23] Mitre Corporation, (2020). https://www.mitre.org (accessed July 25, 2020).

[24] CVE, CVE, Common Vulnerabilities Expo. (2020). https://cve.mitre.org (accessed July 25, 2020).

[25] G. Schryen, Is Open Source Security a Myth?, Commun. ACM. 54 (2011) 130–140. https://doi.org/10.1145/1941487.1941516.

[26] NVD, NVD, Natl. Vulnerability Database. (2020). https://nvd.nist.gov (accessed July 25, 2020).

[27] ExploitDB, Exploit Database, (2020). https://www.exploit-db.com (accessed July 25, 2020).

[28] SecurityFocus, SecurityFocus, (2020). https://www.securityfocus.com (accessed July 25, 2020).

[29] Rapid7, Rapid7, (2020). https://www.rapid7.com/db/ (accessed July 25, 2020).

[30] Snyk, Snyk, (2020). https://snyk.io (accessed July 25, 2020).

[31] SARD, SARD-Software Assurance Reference Dataset Project, (2020). https://samate.nist.gov (accessed July 25, 2020).

[32] G. Spanos, L. Angelis, A multi-target approach to estimate software vulnerability characteristics and severity scores, J. Syst. Softw. 146 (2018) 152–166. https://doi.org/10.1016/j.jss.2018.09.039.

[33] E.R. Russo, A. Di Sorbo, C.A. Visaggio, G. Canfora, Summarizing vulnerabilities’ descriptions to support experts during vulnerability assessment activities, J. Syst. Softw. 156 (2019) 84–99. https://doi.org/10.1016/j.jss.2019.06.001.

Authors’ Profiles

Hakan Kekül is currently working as a teacher at Sivas Information Technology Technical High School. In 2006, he received his undergraduate degree from Sakarya University, Department of Electronics and Computer Education. In 2018, he received his bachelor's degree in Computer Engineering from Cumhuriyet University. In 2017, he received his Master's degree from Cumhuriyet University. Since 2018, he is a PhD candidate at Fırat University, Department of Computer Engineering.

Burhan Ergen is currently Prof. in Department of Computer Engineering at Fırat University. He received his BS degree in Electronics Engineering from Karadeniz Technical University in 1993. He received his master's degree from Karadeniz Technical University in 1996 and his doctorate degree from Fırat University in 2004. He is currently working at Fırat University.

Halil Arslan is currently a faculty member at Cumhuriyet University Computer Engineering Department. He received his undergraduate, graduate and doctorate degrees from Sakarya University Electronics and Computer Education Department in 2006, 2008 and 2016, respectively. He is currently working at Cumhuriyet University.

How to cite this paper: Hakan KEKÜL, Burhan ERGEN, Halil ARSLAN, "Comparison and Analysis of Software Vulnerability Databases", International Journal of Engineering and Manufacturing (IJEM), Vol.12, No.4, pp. 1-14, 2022. DOI:10.5815/ijem.2022.04.01