International Journal of Latest Engineering and Management Research (IJLEMR)
ISSN: 2455-4847
www.ijlemr.com || Volume 08 – Issue 01 || January 2023 || PP. 14-23

Analysis of Trends in Software Vulnerabilities

Santosh Saklani[1],*, Anshul Kalia[2] and Sumesh Sood[3]
[1]*Department of Computer Science, Himachal Pradesh University, Shimla, India*
[2]*Department of Computer Science, Himachal Pradesh University, Shimla, India*
[3]*Department of Computer Science, Himachal Pradesh University, Shimla, India*
**Corresponding author*

Abstract: Software vulnerabilities have focused attention of many researchers in recent years because of its impact on software security**.** To protect software from vulnerabilities, it is important to understand the pattern or trends of these vulnerabilities over the years. This will help software vendors and developers to prepare better solutions for the vulnerabilities before they get exploited in future. In this paper the trend of different types of software vulnerability and severity of vulnerabilities in National Vulnerabilities Database (NVD) are examined from 2016 to 2021. In addition to this, trends of most exploited software products during the period of 2016 to 2020 are analyzed. This will help the software vendors to find if trends in vulnerabilities have any significant meaning or not. In this study results show that cross site scripting (CWE-79), improper privilege management (CWE-269), incorrect authorization (CWE-863) and deserialization of untrusted data (CWE-502) vulnerabilities follow an increasing trend from 2016 to 2021. While on the other hand buffer error (CWE-119) and access control (CWE-264) vulnerabilities follow an important decreasing trend. Results show that vulnerabilities with low severity follow an increasing trend and vulnerabilities with critical severity follow a decreasing trend. It has also been observed that most of the software products does not follow any trend.Android is observed as a highly vulnerable software product in 2016, 2017 and 2020 while in 2018 and 2019 Debian Linux is observed as the most vulnerable software product.

Keywords: trend in software vulnerabilities, vulnerabilities severity, national vulnerability database

1. Introduction

Use of software is increasing day by day. It can be seen everywhere from traditional use of software in computer to web applications, apps on smartphones, embedded code in smart devices like AC, cars, washing machine etc. With the growing dependency on software, it is important to provide a secure software product to customer. Threats posed by vulnerabilities are growing rapidly. As per National Vulnerability Database (NVD),the number of vulnerabilities increased from approximately 4650 in 2010 to 6500 in 2015 and jumped to 18608 in 2020. Vulnerability number increased rapidly after 2015 with the increase of software and its users. It is very important to detect the vulnerabilities before they get exploited by attacker or unauthorized person which result in leak of user data and also damage the software reputation among user.

Vulnerabilities are the weakness in the computational logic, e.g., code, found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, or availability [1]. In other simple word vulnerability is a specific flaw in a piece of software that allows attackers to do something malicious. To protect software, it is important to understand the trend and pattern of vulnerabilities over the period as it will be helpful for software vendors or developers to prepare better solution for vulnerabilities that may occur in a software system. In this study following research questions have been addressed:
RQ1: How significantly the software vulnerabilities trend is changing over the period of time?
RQ2: What is the trend of Common Vulnerability Scoring System over the period of time?
RQ3: What is the trend of vulnerabilities within software products over the period of time?

2. Background and Related Work

National Vulnerability Database (NVD) is an important source of software vulnerabilities. It is a database which includes information about all public known software vulnerabilities [1]. Each vulnerability in NVD is assigned a unique id, known as Common Vulnerabilities and Exposures (CVE) identifier. Each CVE consists of important information which include name of affected software product, version of product, vulnerability description, vulnerability exploitation impact and vulnerability score provided by Common Vulnerability Scoring System (CVSS) as shown in below Table1.

Table 1: Example of vulnerability

CVE ID	CVE-2021-45911
Description	An issue was discovered in gif2apng 1.9. There is a heap-based buffer overflow in the main function. It allows an attacker to write 2 bytes outside the boundaries of the buffer. Publish Date: 2021-12-28 Last Update Date: 2022-03-24
CVSS Score	6.8
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Access Complexity	Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Vulnerability Type(s)	Overflow
CWE ID	787

The Common Weakness Enumeration (CWE) identifier is used to define vulnerability type. CWE is a hierarchical list of vulnerability types which is prepared by security experts. Each vulnerability belongs to a certain CWE identifier.The Common Vulnerability Scoring System (CVSS) provides a technique to measure the major characteristics of a vulnerability and produce a numerical score (0-10) depending on its severity [2] . It helps organizations/software vendors to properly assess and prioritize their vulnerability management processes.

Gopalakrishna et al. (2005) performed a study on vulnerabilities data from 1994 to 2002. In its study vulnerabilities in five software artifacts are examined [3]. Among these five software the highest number of vulnerabilities are reported for Microsoft IIS in 1999. As per their result coding errors account for almost 72% of all the vulnerabilities and Buffer Overflows constitute nearly 20% (one-fifth) of all the vulnerabilities.

Kuhn & Johnson (2010) used software vulnerabilities record from 2001 to 2010 to examine the status of vulnerabilities. They found that despite the rapid growing of Applications in number, vulnerabilities have declined by 26 percent Since 2006 [4].

Neuhaus & Zimmermann (2010) find the trend of software vulnerabilities from 2002-2009 [5]. In their study they found that buffer overflows and format strings vulnerabilities are declined. They also found that buffer overflows are harder to exploit now than they were in the past and SQL injection and cross-site scripting have decreasing trend in the last few years.

Chang et al. (2011) performed exploratory study on CVE for the period of 2007 to 2010[6]. Their result shows decreasing trend of high severity vulnerabilities through the years. They also found that over 80% of the vulnerabilities were exploitable by network access without authentication from 2007 to 2010.The number of vulnerabilities with partial CIA impacts has also decreased every year and reduced roughly 12% from 2007 to 2010.

Wijayasekara et al. (2012, June) performed a study on software vulnerability that appeared long after the bug been made public, called hidden impact vulnerabilities for the period from January 2006 to April 2011 [7]. They analysed hidden impact vulnerabilities in Linux kernel and Mysql. They observed that 32% and 62% of vulnerabilities discovered in this time period were hidden impact vulnerabilities in Linux and MySQL respectively and the percentage of hidden impact vulnerabilities in the last two years has increased by 53%for Linux and 10% for MySQL.

Murtaza et al. (2016) used software vulnerabilities data to quantity severity in vulnerabilities. They identified that proportion of high severity vulnerabilities is trending downward, declining about 15 percentage

points since 2008 to 2016[8]. About two-thirds of this fraction has shifted to medium severity vulnerabilities, which increased from about 46% to 55% of the total, while Low severity numbers increased from 3% to nearly 10% of the total.

Kuhn et al. (2017) used software vulnerabilities data from NVD in their study and found that trend of any type of vulnerability in software applications cannot be considered significantly increasing or significantly decreasing despite the rise and fall of trend of vulnerabilities in different years [9].They also found that the next vulnerability can be predicted with approximately 90% precision and 80% recall by using the previous vulnerability.

Ahvanooey et al. (2020) in their study investigate about sensitive security issues affecting on smartphones such as malware attacks, vulnerabilities and categorize them over the period of 2011 to 2017 by focusing on software attacks [10].

In this paper an exploratory study is performed on software vulnerabilities on NVD. A dataset of vulnerabilities from 2016 to 2021 is prepared by downloading data from NVD. Data mining is done to answer the research questions.

3. Research Methodology

In this study firstly dataset is downloaded from the NVD repository for the period of 2016-2021 to measure the likelihood of each vulnerability in each year. This allows us to determine the answer to RQ1. Secondly, trends of vulnerabilities severity based on CVSS is examined to answer RQ2. Lastly, trends of vulnerabilities within individual software applications/products from 2016 to 2020 is analysed.

3.1 Measuring how significantly the software vulnerabilities trend is changing over a period of time (RQ1)

To determine the significance in the trends of vulnerabilities, likelihood of each vulnerability in a year according to equation 1 is measured. In equation 1*vi* is the vulnerability of which likelihood is measured in year '*y'*. This is measured by counting the occurrence of that vulnerability in a year and dividing it by the total number of all vulnerabilities in that year. In equation 1, *V* is a set of all vulnerability types in a year and $vi \in V$.

$$\text{Likelihood (vi, y)} = \frac{\text{count } (vi, y)}{\Sigma\, v \epsilon\, V \text{ count } (v, y)} \quad (1)$$

3.2 Measuring trends of Common Vulnerability Scoring System over a period of time (RQ2)

To determinetrends of Common Vulnerability Scoring System (CVSS) over a period of time (RQ2). Severity of vulnerability for the period of 2016-2021 are analysed with the aim of examine the year wise trend in CVSS. Percentage of CVSS for different levels and rating is calculated for every year to find the trend.

3.3 Measuring trends of vulnerabilities with in software products over a period of time (RQ3)

To determine trends of vulnerabilities with in software products, percentage of software vulnerabilities within the top 25 most frequently exploited software in the period of 2016-2020 is analysed (due to unavailability of records, data for year 2021 is not included). In RQ3, significance of trends in the frequency of vulnerabilities within a particular software application is investigated. This will help in determining, whether a software product have really reduced the vulnerabilities or not.

To determine the significance in the trends of vulnerabilities in software products, percentage of vulnerabilities in software products is measured using equation 2. In equation 2*vp* is the number of vulnerabilities in a product *pi* of which percentage is measured in a given year '*y'*. This is measured by counting the number of vulnerabilities occurred in a product in a given year and dividing it by the total number of all vulnerabilities in that year. In equation 2, *V* is a set of all vulnerability types in a given year and $vp \in V$.

$$\text{Percentage } (vp, pi, y) = \frac{\text{count } (vp, pi, y)}{\Sigma\, v \epsilon\, V \text{ count } (v, y)} \quad (2)$$

4. Result

This section provides answers to all the research questions that are identified in earlier section. In this study, vulnerability identifier (CVE), vulnerability type, vulnerability severity and vulnerable software applications/products from NVD repository are used. Dataset of software vulnerability download from NVD from 2016-2021 consist of 90710 vulnerabilities. There are 13508 vulnerabilities whose vulnerability types did

not exist. those vulnerabilities are not considered in this study for analysis, finally dataset of 77202 records is selected.

RQ1: How significantly the software vulnerabilities trend is changing over a period of time?

Table 2: Vulnerability Type (Common Weakness Enumeration: CWE) and its Description

CWE id	**Vulnerability Description**	**CWE id**	**Vulnerability Description**
CWE-79	Cross Site Scripting	CWE-400	Uncontrolled Resource Consumption
CWE-787	Out-of-bounds Write	CWE-264	Permissions, Privileges, and Access Controls Category ID: 264
CWE-119	Buffer Errors	CWE-732	Incorrect Permission Assignment for Critical Resource
CWE-20	Input Validation	CWE-862	Missing Authorization
CWE-200	Information Leak	CWE-284	Improper Access Control
CWE-125	Out-of-bounds Read	CWE-502	Deserialization of Untrusted Data
CWE-89	SQL Injection	CWE-798	Use of Hard-coded Credentials
CWE-416	Use After Free	CWE-77	Command Injection
CWE-22	Path Traversal	CWE-611	Improper Restriction of XML External Entity Reference
CWE-352	Cross Site Request Forgery	CWE-94	Code Injection
CWE-78	OS Command Injection	CWE-295	Improper Certificate Validation
CWE-190	Integer Overflow or Wraparound	CWE-522	Insufficiently Protected Credentials
CWE-476	NULL Pointer Dereference	CWE-362	Race Conditions
CWE-287	Authentication Issues	CWE-306	Missing Authentication for Critical Function
CWE-269	Improper Privilege Management	CWE-918	Server-Side Request Forgery
CWE-434	Unrestricted Upload of File with Dangerous Type	CWE-601	URL Redirection to Untrusted Site
CWE-863	Incorrect Authorization	CWE-74	Injection
CWE-120	Buffer Copy without Checking Size of Input		

In NVD software vulnerabilities are described into different types by providing it a CWE-id. The CWE identifier of top 35 vulnerabilities with their description are presented in Table 2. Likelihood of these vulnerabilities from 2016-21 are calculated using equation 1, as shown in Table 3. Table also present the trend of these vulnerabilities over the years. CWE-79 (Cross Ste Scripting), CWE-269 (Improper Privilege Management), CWE-863 (Incorrect Authorization) and CWE-502 (Deserialization of Untrusted Data) have seen as increasing trend from 2016-21. CWE-119 (Buffer Error) and CWE-264 (Access Control) have seen decreasing trend over the time. Other vulnerabilities in Table 3 show fix trend during 2016-21. Vulnerabilities with important increasing and decreasing trend are highlighted in Table 3. These highlighted vulnerabilities of Table 3 are shown in Fig.1 for better visualization.

In 2018, 2019, 2020 and 2021 CWE-79 (Cross Site Scripting vulnerability) has the highest likelihood while likelihood of CWE-119 (Buffer Errors vulnerability) has observed highest in the year 2016 and 2017.

Table 3: Likelihood of vulnerabilities (in percentage) from 2016 to 2021 (CWE is the vulnerability identifier described in Table 2)

Vulnerability Type	**Year**					
	2016	**2017**	**2018**	**2019**	**2020**	**2021**
CWE-79	**9.678954**	**11.6206**	**14.92548**	**13.43989**	**14.61487**	**14.49003178**
CWE-787	1.129608	2.182789	6.577816	10.53579	9.397187	6.927188674
CWE-119	**16.37337**	**16.33166**	**6.07624**	**1.755966**	**1.111855**	**1.596359434**

CWE-20	8.43044	8.825377	7.437661	3.969683	4.534494	3.842819994
CWE-200	11.58145	9.657663	6.226713	2.071139	1.366376	2.311470673
CWE-125	2.378121	5.464824	5.846948	6.130872	4.889484	3.597226235
CWE-89	2.021403	3.83951	3.589854	3.129221	3.83791	3.698353077
CWE-416	2.009512	2.567525	3.453712	3.489419	2.632284	3.286622363
CWE-22	1.498216	2.740264	3.045285	2.971634	3.020764	2.701531349
CWE-352	2.187872	2.167085	3.245916	3.196758	2.732753	2.188673794
CWE-78	0.463734	1.389761	2.730009	3.016659	2.993972	1.697486276
CWE-190	1.533888	0.188442	4.721983	1.575867	1.306095	1.386882404
CWE-476	1.736029	2.41049	1.991975	1.906048	1.359678	1.964750072
CWE-287	1.200951	1.923681	1.991975	1.54585	1.835231	2.304247327
CWE-269	**0.225922**	**0.950063**	**0.938664**	**1.020561**	**1.6142**	**4.752961572**
CWE-434	0.344828	0.675251	1.210949	1.343239	1.728064	1.632476163
CWE-863	**0.118906**	**0.431847**	**0.695042**	**0.675371**	**1.373074**	**3.481652702**
CWE-120	0.107015	0.353329	0.336773	1.298214	2.183523	2.087546952
CWE-400	0.499405	1.146357	1.175122	0.802942	1.312793	1.639699509
CWE-264	**9.203329**	**0.078518**	**0.071654**	**0.165091**	**0.006698**	**0.007223346**
CWE-732	0.071344	0.997173	1.941817	1.245685	0.843938	0.794568044
CWE-862	0.011891	0.439698	0.523073	1.650908	1.694575	1.061831841
CWE-284	7.122473	0.125628	0.222127	2.131172	0.147354	0.390060676
CWE-502	**0.487515**	**0.683103**	**0.738034**	**0.930512**	**1.125251**	**1.27130887**
CWE-798	0.475624	0.800879	0.931499	0.923008	1.118553	0.816238081
CWE-77	0.796671	0.526068	0.29378	0.742909	1.024782	1.791389772
CWE-611	0.618312	0.722362	1.275437	0.855471	0.756865	0.693441202
CWE-94	0.451843	0.643844	1.096303	0.81795	0.629605	1.047385149
CWE-295	0.511296	1.256281	0.680711	0.705388	0.83724	0.693441202
CWE-522	0.023781	0.510364	0.809688	1.395768	0.977897	0.722334585
CWE-362	0.760999	0.902952	0.66638	0.697884	0.884126	0.650101127
CWE-306	0.059453	0.274812	0.315277	1.613387	1.440054	0.332273909
CWE-918	**0.154578**	**0.408291**	**0.580396**	**0.622843**	**1.011386**	**0.960704999**
CWE-601	0.487515	0.683103	0.566065	0.7279	0.663094	0.780121352
CWE-74	0.297265	0.879397	0.336773	0.56281	0.797053	0.881248194

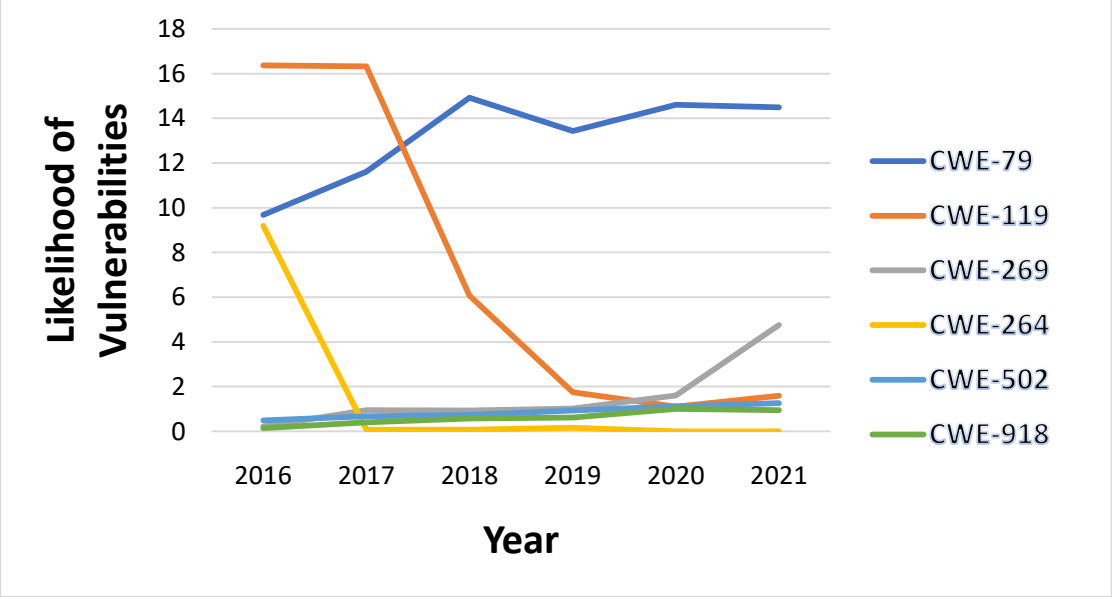

Figure 1: Selected vulnerabilities and their likelihood over a period of time

RQ2: What is the trend of Common Vulnerability Scoring System over a period of time?

In this section trend of Common Vulnerability Scoring System is taken over a period of time to examine the severity of vulnerabilities for the period of 2016-2021. Table 4 shows trend of CVSS of these vulnerabilities over the year.

Table 4: Year wise percentage of Vulnerability Score

CVSS Score	**percentage of CVSS score**					
	2016	**2017**	**2018**	**2019**	**2020**	**2021**
0 - 1	0	0.2	0.2	0.1	0	0.9
1 - 2	1	0.9	0.7	0.5	0.6	0.4
2 - 3	4.5	3.7	3.3	5.1	6.3	6.1
3 - 4	**4**	**5**	**6.7**	**6.1**	**8.2**	**9.1**
4 -5	24	27.8	25.2	28.9	26.4	27.1
5 - 6	15	17	21.6	18.2	20.3	18.2
6 - 7	13.4	16.1	16.5	16.2	14.3	18.3
7 - 8	16.3	18.4	15.8	16.3	15.1	13.5
8 - 9	0.6	0.4	0.5	0.5	0.6	0.6
9 - 10	**21.1**	**10.4**	**9.5**	**8.2**	**8.2**	**5.8**

Table 5: Severity Rating of CVSS score

Severity Rating	**CVSS Score**
None	0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

Table 5 categorized CVSS into different rating. CVSS score 0-3.9 (low severity) have seen as increasing trend with 7% increase and CVSS score 9-10 (critical severity) have seen as important decreasing trend with15.7 % decrease from 2016-21 as shown in Table 4 andTable 6. Other CVSS score and severity shown in Table 2 andTable 4 respectively have fix trend during 2016-21. Vulnerabilities with important increasing or decreasing trend are highlighted in Table 4 and Table 6. Year-wise percentage of rating for vulnerability severity is also shown in Fig. 2 for better visualization.

Table 6: Year-wise percentage of CVSS rating

Severity Rating	**Percentage of CVSS score**					
	2016	**2017**	**2018**	**2019**	**2020**	**2021**
Low	**9.5**	**9.8**	**10.9**	**11.8**	**15.1**	**16.5**
Medium	52.4	60.9	63.3	63.3	61	63.6
High	16.9	18.3	16.3	16.8	15.7	14.1
Critical	**21.1**	**10.4**	**9.5**	**8.2**	**8.2**	**5.8**

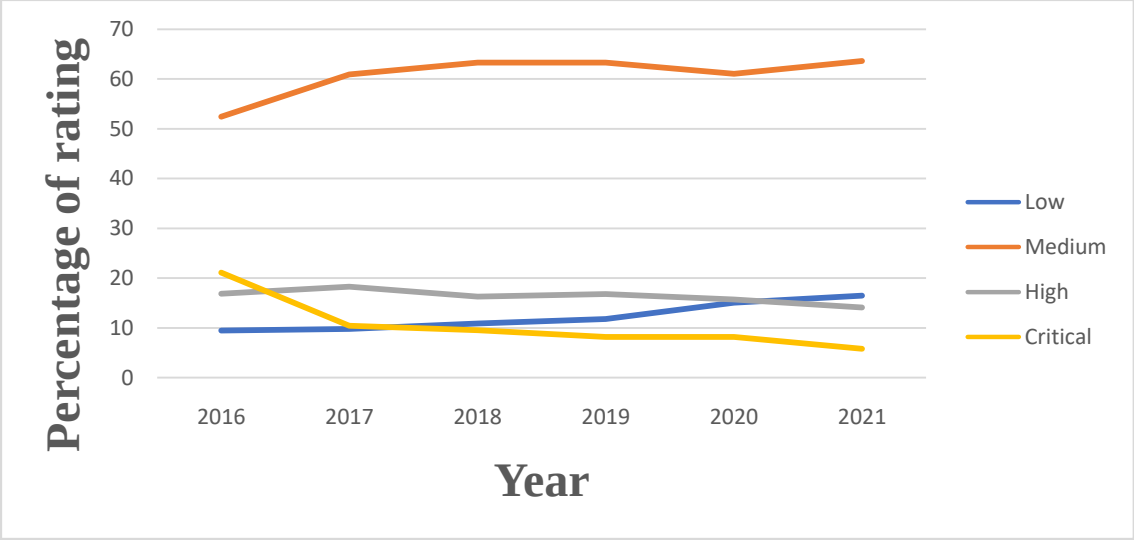

Figure2: Year-wise percentage of rating for vulnerability severity

RQ3: What is trend of vulnerabilities with in software products over a period of time?

In this section percentage of software vulnerabilities within the top 25 most frequently exploited software in the period of 2016-2020 is analysed. Data for year 2021 was not available at the time of data collection. The percentage of vulnerabilities are calculated using equation 2.

Trend on percentage of vulnerabilities in most exploited software from 2016-20 is shown in Table 7. Table 7 also present detail of vendor name to which respective product belong. results show that no product have increasing or decreasing trend from 2016-20. Debian Linux has big increase in vulnerability percentage in year 2018 but after 2018 it shows important decreasing trend. Fedora,

Windows 10, Windows Server 2016, Windows Server 2008, Windows 7, Windows Server 2012, and Leap show increasing trend after year 2017. Vulnerabilities with important increasing and decreasing trend are highlighted in Table 7 and presented in Fig. 3 for better visualization. All other products show mix trend during 2016-20(for example Android as shown in Fig. 3.

It is observed from Table 7 that Android is the highly vulnerable software product in 2016, 2017 and 2020. Similarly, Debian Linux is observed as the most vulnerable software product in 2018 and 2019.

Table 7: Percentage of vulnerabilities in selected software products from 2016 to 2020

SN	Product Name	Vendor Name	2016	2017	2018	2019	2020
1	Debian Linux	Debian	**4.429713805**	**0.051305**	**8.755413**	**5.616293**	**4.956305**
2	Android	Google	5.260942761	0.057385	3.822256	3.067408	5.362047
3	Ubuntu Linux	Canonical	3.282828283	0.015576	5.38505	2.973699	2.515605
4	Fedora	Fedora project	**1.420454545**	**0.00772**	**0.483274**	**3.111139**	**4.263421**
5	Mac Os X	Apple	2.293771044	0.021041	0.690391	1.924158	1.910112
6	Linux Kernel	Linux	2.262205387	0.030674	1.117178	1.811707	0.786517
7	Windows 10	Microsoft	**1.80976431**	**0.017899**	**1.619281**	**2.798776**	**5.037453**
8	IphoneOs	Apple	1.767676768	0.02678	0.784535	2.224027	1.966292
9	Windows Server 2016	Microsoft	**0.410353535**	**0.016874**	**1.525137**	**2.767539**	**4.956305**
10	Chrome	Google	1.788720539	0.004372	1.004205	1.949147	1.416979
11	Windows Server 2008	Microsoft	**1.399410774**	**0.016532**	**0.985376**	**1.961642**	**2.384519**
12	Windows 7	Microsoft	**1.40993266**	**0.015644**	**1.023034**	**2.005373**	**2.421973**
13	Windows Server 2012	Microsoft	**1.641414141**	**0.015986**	**1.02931**	**1.961642**	**2.765293**
14	Windows Server 2019	Microsoft	**--**	**--**	**--**	**2.705067**	**4.637953**
15	Firefox	Mozilla	1.399410774	0	1.826398	0.712188	0.88015
16	Windows 8.1	Microsoft	1.62037037	0.015234	1.035587	1.936653	2.715356
17	Windows Rt 8.1	Microsoft	1.473063973	0.012228	0.872403	1.849191	2.677903
18	Enterprise Linux Desktop	Redhat	1.767676768	0.015029	3.364087	1.399388	--
19	Enterprise Linux Server	Redhat	1.799242424	0.014893	3.539823	1.48685	--
20	Enterprise Linux Workstation	Redhat	1.778198653	0.014005	3.420574	1.430624	--
21	Leap	Opensuse	**3.061868687**	**0.006285**	**0.621352**	**2.954957**	**3.289638**
22	Tvos	Apple	1.157407407	0.015576	0.370301	1.574311	1.373283
23	Enterprise Linux Server Aus	Redhat	0.70496633	0.006148	1.807569	0.612232	--
24	Mysql	Oracle	1.073232323	0.00608	0.677838	0.874617	0.867665
25	Safari	Apple	0.589225589	0.012502	--	1.037046	0.474407

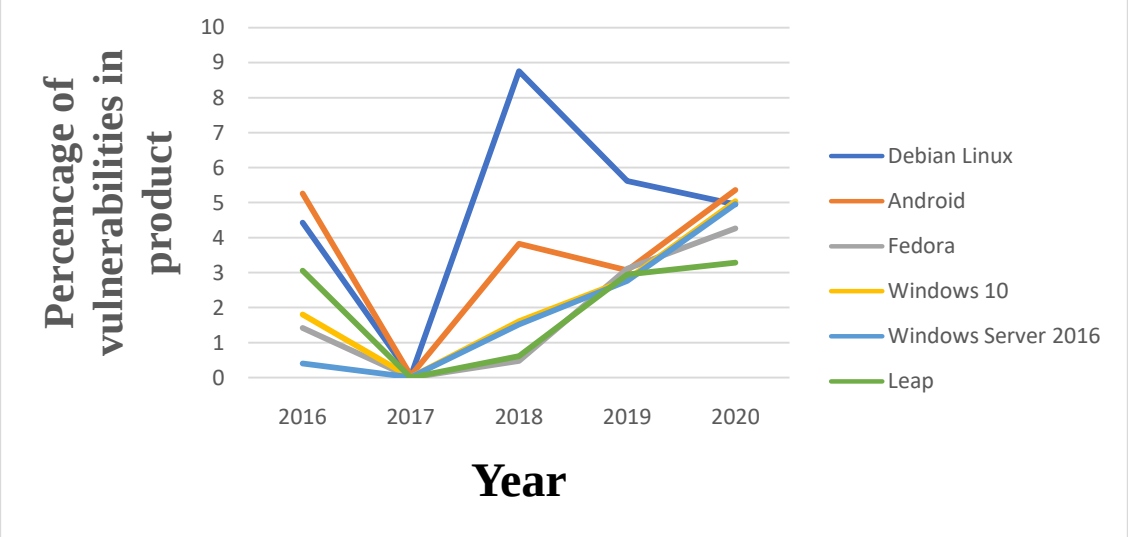

Figure 3: percentage of vulnerabilities in product

5. Threats to Validity

A threat to validity may occur due to the unavailability of information about some vulnerabilities. In the original NVD dataset (2016-2021) some vulnerabilities contain null value or does not have information about CWE identifier (vulnerability type). These vulnerabilities are ignored during pre-processing and filtering of dataset. This may have altered the results of this study. However, such vulnerabilities are very less in number and they do not affect the overall result.

6. Conclusion and Future Work

Main finding obtains from the selected studies are:

- Likelihood of vulnerability type CWE-79, CWE-269, CWE-863 and CWE-502 have shown increasing trend over the time. Likelihood of vulnerability types CWE-119 and CWE-264 have shown decreasing trend from 2016-21.
- In 2018, 2019, 2020 and 2021 cross site scripting vulnerabilities and in 2016 and 2017 buffer error vulnerabilities have shown the highest likelihood.
- CVSS score 0-3.9 (low severity) have seen as increasing trend and CVSS score 9-10 (critical severity) have seen as important decreasing trend from 2016-21. Fix trend is observed for medium and severity. Majority of the vulnerabilities from 2016 to 2021 has medium severity.
- Percentage of vulnerabilities in most of the software products does not follow any significant trend. Debian Linux has big increase in 2018 but after 2018 it shows decreasing trend. Fedora, leap, Windows 7 &10, windows server 2008, 2012 and 2016 have shown increasing trend in vulnerabilities percentage after 2017.
- Android is observed as highly vulnerable software product in 2016, 2017 and 2020. Similarly, Debian Linux is observed as the most vulnerable software product in 2018 and 2019.

Guideline for researchers and software practitioners for carrying out research work in future on software Vulnerabilities:

(1) Vulnerability dataset can be used to train machine learning model to predict the vulnerability in a software product.
(2) Security of software is also one of the important sub characteristics of software product quality. More work needed to be done in finding trend of software vulnerabilities so that software vendors can work to ensure software security during development process.

References

[1] National Vulnerability Database, Vulnerabilities. Retrieved from http://nvd.nist.gov. Accessed July 24, 2022.

[2] FIRST, Common Vulnerability Scoring System.Retrieved from https://www.first.org/cvss/user-guide. Accessed August 16, 2022.

[3] R. Gopalakrishna, E. H. Spafford, J. Vitek, A trend analysis of vulnerabilities, *West Lafayette: Purdue University*, *13*,2005.

[4] R. Kuhn, C. Johnson, Vulnerability trends: measuring progress, *IT professional, 12(4)*, 51-53, (2010).

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

[6] Y.Y. Chang, P. Zavarsky, R. Ruhl, D. Lindskog, Trend analysis of the cve for software vulnerability management. *IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing pp*. 1290-1293, IEEE, 2011.

[7] D. Wijayasekara, M. Manic, J. L. Wright, M. McQueen, Mining bug databases for unidentified software vulnerabilities, *5th International conference on human system interactions*, pp. 89-96, IEEE, 2012

[8] S. S. Murtaza, W. Khreich, A. Hamou-Lhadj, A. B. Bener, Mining trends and patterns of software vulnerabilities, *Journal of Systems and Software, 117*, 218-228, 2016.

[9] D. R. Kuhn, M. S. Raunak, R. Kacker, An Analysis of Vulnerability Trends, 2008-2016, *IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pp. 587-588, IEEE, 2017.

[10] Ahvanooey Milad Taleby, Qianmu Li, Mahdi Rabbani, Ahmed Raza Rajput, A survey on smartphones security: software vulnerabilitis,malware, and attacks, *arXiv preprint arXiv:2001.09406*, 2020.