ASTES

Advances in Science, Technology and Engineering Systems Journal Vol. 7, No. 6, 230-235 (2022)

www.astesj.com

ASTESJ

ISSN: 2415-6698

Operating Systems Vulnerability - An Examination of Windows 10, macOS, and Ubuntu from 2015 to 2021

Jasmin Softić[*], Zanin Vejzović

Faculty of Computer Science, Sarajevo School of Science and Technology, 71000 Sarajevo, Bosnia and Herzegovina

ARTICLE INFO

Article history:
Received: 06 September, 2022
Accepted: 30 November, 2022
Online: 20 December, 2022

Keywords:
Operating systems
Vulnerability database
Vulnerability
Vulnerability score
Vulnerability life cycle
days-of-risk

ABSTRACT

This study investigated the vulnerabilities of three operating systems: Windows 10, macOS, and Ubuntu. The analysis of secondary data obtained from the CVE and NVD databases for the study period demonstrates varying OS vulnerability. Quantitative assessment of the vulnerability (using the vulnerability score) for the investigated operating systems found consistent results in the security vulnerability of these OS. The correlation of the disclosed vulnerabilities data and the average weighted vulnerability yielded coefficients of -0.3674, -0.4081, and 0.3473 for macOS, Windows 10, and Ubuntu Linux. These results demonstrate windows 10 as having the highest security vulnerability, followed by macOS. Ubuntu Linux had the lowest vulnerability scores. These results were validated by the CVSS distribution of the vulnerability score. The results point to the impact of the popularity of OS on the number of attacks in a given period. OS used by many people tend to attract significant attacks testing their integrity, security, and safety.

1. Introduction & Background

With the advent of big data and analytics, the internet and information systems have become increasingly important for organizations. The growth in the significance of information and computer systems has witnessed increased attacks characterized as malware, virus, or ransomware. Since 2017, cybercriminals have increasingly deployed ransomware to information systems, gained access to files, encrypted them, and demanded millions of dollars from victims for a decryption key. According to [1], ransom demands have increased significantly since 2020. Others also observed cyber-attacks have evolved and are difficult to detect. The success of these attacks points to a continuing vulnerability in information technology systems that attackers can exploit to their advantage. According to the Common Vulnerabilities and Exposures (CVE) (n.d.), a vulnerability refers to “a weakness in the computational logic (e.g., code) found in software and hardware components that, when exploited, results in a negative impact on confidentiality, integrity, or availability.” Therefore, operating system (OS) vulnerability can be described as exposures or weaknesses within an OS that allows a cyber-attacker/intruder to undermine the integrity of the OS, or any system installed on it as per [2], [3] and others.

OS vulnerability can either be due to errors in the development process or unpatched or outdated OS that increases the opportunity for security breaches as described in [2] and [4] When coupled with negative user behaviors such as those examined by [5] and others, these vulnerabilities provide attackers with easy access to a system. Outdated software is also a growing cause of vulnerability as it does not take into consideration new updates released as a result of new research or studies indicating their areas of weaknesses [6], though some studies indicate outdated software is difficult to compromise than up to date software [7]. Particularly, attackers have exploited these vulnerabilities to execute Ransom denial of service (RDoS) attacks that have cost individuals and organizations millions of dollars across the world [5], [8]. Besides a denial of service, other vulnerabilities reported by vulnerability databases and vendors include code execution, overflow, exploits, memory corruption, SQL injection, gaining of privileges, HTTP response splitting, file inclusion, XSS, and directory traversal. These vulnerabilities are reported alongside the vulnerability life cycle. Various studies have been conducted on the vulnerability life cycle of software applications and operating systems. They include [9]-[12] studies. While the vulnerability cycle of operating systems is still under exploration, researchers tend to agree that the life cycle is divided into five crucial stages as illustrated in Figure 1. These stages include (a) vulnerability birth or creation (the time when OS weakness is created), (b) vulnerability discovery (the time when OS vulnerability is identified by vendor) (c)

[*] Corresponding Author: Jasmin Softić, jasmin.softic@stu.ssst.edu.ba

https://dx.doi.org/10.25046/aj070625

vulnerability disclosure (vendor makes the vulnerability known to the public), (d) patch availability (vendor provides a quick fix to the weakness), and (e) patch installation (the public users of the affected OS install the quick fix solution to address system weakness) [9], [11]. In [3] the author suggested an extra stage described as the "exploit stage" to be inserted between the first and fifth stages indicating that the vulnerability of the system could be exploited before the availability of a patch. They provided a clear demonstration of the vulnerability life cycle as demonstrated in Figure 1.

The time intervals between these stages carry different risks for users for a given system vulnerability. These risks have been given different names. The period between vulnerability discovery or its disclosure to when the patch is installed to fix it is known as the days of risk [3]. The terms black risk, grey risk, and white risk are utilized to describe the awareness of the public regarding vulnerability. Black risk describes the lack of public awareness about the existence of vulnerability in the software or hardware they use.

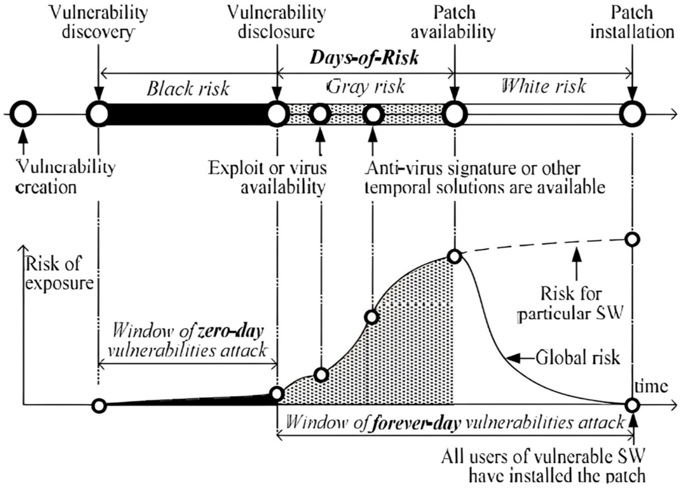
Vulnerability discovery
Vulnerability disclosure
Patch availability
Patch installation
Days-of-Risk
Black risk
Gray risk
White risk
Vulnerability creation
Exploit or virus availability
Anti-virus signature or other temporal solutions are available
Risk of exposure
Window of zero-day vulnerabilities attack
Risk for particular SW
Global risk
time
Window of forever-day vulnerabilities attack
All users of vulnerable SW have installed the patch

Figure 1: Vulnerability life cycle for software products

While developers and researchers continue to investigate how to address these vulnerabilities, attacks continue to be reported for new product releases by various vendors, indicating the continued existence of errors or weaknesses in the new OS development or evolution of attacks with the ability to overcome OS updates and upgrades. Besides, OS vendors also release patches to their old operating systems to fix vulnerabilities that could be exploited by attackers. Some OS vulnerabilities have been reported, while others have not. For reported vulnerabilities, patches have been released soon before they are exploited. However, for other OS weaknesses, vendors have taken longer to release patches, further exposing users of their products to cyber risks [13]. Enterprise operating systems by major technology companies, including Windows, macOS, Ubuntu, and Google Chrome OS, have been attacked due to their vulnerabilities as discussed by [14], [15] and [16] in their studies. These vulnerabilities have been defined in the literature. However, this study focuses on selected vulnerabilities reviewed in Table 1.

Table 1: OS vulnerabilities

Vulnerability	Definition and Impact of Attack
DoS	An attack meant to shut down OSs, servers, or computing systems making them inaccessible to intended users. DoS attacks can lead to bandwidth depletion or resource depletion [17]
Code Execution	In this type of vulnerability an attacker is able to run an arbitrary code of their choosing with system level privileges on an OS or server that possesses the appropriate weakness. A remote code execution can lead to a denial-of-service attack and sensitive data exposure [18]
Overflow	An overflow is vulnerability in coding errors that attackers can exploit to access systems without authorization. An attacker can manipulates coding errors, altering an application path, and overwrite system memory with this vulnerability [19] ,[20]
Memory Corruption	Entails the altering of OS memory without explicit assignment due to programming errors [21]
Sql Injection	A vulnerability type that allows an attacker to interfere with the queries of applications to their databases, allowing them to access sensitive data [22]
XSS (Cross-site scripting)	This vulnerability makes it possible for attackers to interfere with user interaction with the OS or an application by allowing them to circumvent same origin policy [23],[24]
Directory Traversal	A system weakness that provides an attacker with a means for reading arbitrary files on the server running certain applications. It allows attackers access to sensitive information and data [25]
Http Response Splitting	A web application vulnerability caused by the failure of an application to properly sanitize various input values [26]. Its impacts include web-cache poisoning, XSS attacks, and cross user defacement.
Bypass something	Entails the exploitation of weak OS authentication mechanisms allowing the attacker to access system data [27]
Gain Information	Vulnerabilities that allow attackers to gain information from an operating system
Gain Privileges	OS weaknesses that provide a means for hackers to gain system privileges without proper authentication

A review of the literature by the researcher found limited investigations into the vulnerabilities of popular enterprise operating systems. In [28] conducted a vulnerability assessment of Windows 10 and employed CVE data to test the security of the system. In [29] author examined Linux OS security and how updates can help overcome some of its vulnerabilities. Additionally, in [3] the author examined the vulnerabilities of six operating systems, including Ubuntu, Red Hat, Windows, macOS, Oracle, and Linux. The study analyzed data from 2012 to 2016. In [3] study is among those that extensively utilized already existing vulnerability data to assess different OSs. However, there are no recent studies examining the vulnerability of the operating systems in the context of increased cyber-attacks and cyber wars between countries before and after COVID-19. The researcher attempts to fill this gap by examining the vulnerability of three popular OSs including Ubuntu, Windows, and macOS. The study focuses on the period 2015 to 2021 to investigate how OS vulnerability has changed over the years amid these cyber wars. The year 2015 is considered because the researcher wants to observe the changes in the subsequent years to understand how vulnerability statistics have changed over the years.

2. Study Objectives

This paper aims to identify changes in operating systems vulnerability over the study period even as cyber wars increase globally to extend literature findings and contribute to the body of knowledge on OS security. To address the research gaps, this study is guided by the following objectives:

- Analyze vulnerabilities that have been disclosed and fixed by respective vendors for the operating systems under study.
- Perform a quantitative comparison of the vulnerability (using the vulnerability score) for different operating systems for the period under study.
- Identify the major vulnerabilities common to the operating systems under examination reported by vendors.
- Identify how OS vulnerabilities have changed over the years (increased or decreased) at the time when significant global attacks (such as ransomware) have been reported.

3. Research Methods

This study employs data from two major databases that aggregate OS statistics to investigate the security vulnerabilities of the selected operating systems. These databases are the Common Vulnerabilities and Exposures System (CVE) and the National Vulnerabilities Database (NVD). CVE (https://cve.mitre.org/) and NVD (https://nvd.nist.gov/) make vulnerability data readily available through their websites. While there are other institutions (such as VNDB and Security Tracker) providing data on the vulnerability of operating systems and other software, the researcher considers CVE and NVD data to be sufficient for this study.

CVE is provided by MITRE Inc., a not-for-profit organization that generates a list of known vulnerabilities and assigns them a CVE-ID. The ID is used for synchronizing with CVE, enabling data exchange. NVD, on the other hand, is provided by the U.S. National Institute of Standards and technology. NVD provides a classification of the severity of the vulnerability and type, which are crucial for this study. The severity of the vulnerability is classified using a CVSS (Common Vulnerability Scoring System) score. Other scoring systems include the Common Platform Enumeration Dictionary (CPE) and the Common Weakness Enumeration Specification (CWE). CWEs are used to classify OS vulnerabilities and provide "a common language of discourse for discussing, finding, and dealing with causes of software security vulnerability as they are found in code, design, or system architecture",[30].A single vulnerability is represented by a unique CWE. A hierarchical structure is utilized to hold CWEs, enabling multiple levels of abstraction. For example, CWE-311 (missing encryption for sensitive data) is split into CWE-312 (clear-text storage of sensitive information) and CWE-319 (clear-text transmission of sensitive information). The study identifies the most common vulnerabilities facing these OSs using their CWEs. The study examines data from 2015 to 2021, a six-year period. The operating systems to be examined are Windows 10 (by Microsoft Corporation), Mac OS (from Apple Inc.), and Ubuntu (Canonical Ltd.). The vendors for the selected operating systems often issue bulletins about the vulnerabilities in their operating systems. These vendors and their operating systems were chosen because of their popularity among users, suggesting their attractiveness to malicious attackers. It is expected that the higher the number of users for an operating system, the elevated the rate of vulnerability identification and reporting. This ensures the researcher accesses sufficient data for analysis. The study period is chosen to ensure the collected data has an element of reliability by combining vulnerability data for old operating systems and recent operating systems. In [3] the author argued that NVD and CVE vulnerability reports are statistically insufficient for the most recent versions of operating systems. For Microsoft, the operating system under investigation is Windows 10, released on 29 July 2015. For Canonical, vulnerabilities for Ubuntu Linux operating systems are examined. The Apple operating system investigated in this study has undergone several changes in name over the years. Mac OS X data is utilized in this study. Data extracted from these websites are analyzed using Microsoft Excel software to attain the study objectives.

4. Results and Discussions

4.1. Disclosed Vulnerabilities

Table 2 lists these operating systems and the number of vulnerabilities reported from 2015 to 2021. Between 2015 and 2021, the Ubuntu OS [31] reported the highest number of vulnerabilities followed by Windows 10 [32], and macOS [33] came in the third position as illustrated in Table 2.

Table 2: Disclosed cumulative vulnerabilities for the selected Oss

Year	Windows 10	Ubuntu Linux	macOS
2015	57	321	407
2016	172	319	218
2017	262	228	308
2018	258	860	110
2019	448	484	308
2020	807	423	306
2021	485	25	315
Total	**2489**	**2660**	**1972**

Table 3 shows the vulnerabilities reported for the studied OSs. The top five vulnerabilities for the three operating systems are code execution (22.97%), DoS attacks (22.83%), overflow (18.48%), memory corruption (11.25%), and gaining information (9.65%). The code execution vulnerability was highest for macOS followed by Windows 10. DoS vulnerability was highest for Ubuntu while for Windows 10, code execution was the dominant vulnerability.

Table 3: Vulnerability types reported between 2015 and 2021

	Windows 10	Ubuntu	macOS	Total	% Of All
DoS	164	823	639	1626	22.83
Code Execution	514	290	832	1636	22.97
Overflow	161	520	635	1316	18.48
Memory Corruption	49	143	609	801	11.25
Sql Injection	0	2	2	4	0.06
XSS	4	22	8	34	0.48
Directory Traversal	4	30	8	42	0.59
Http Response Splitting	0	3	0	3	0.04
Bypass something	152	122	134	408	5.73
Gain Information	336	155	199	690	9.69
Gain Privileges	205	48	121	374	5.25

Figure 2 shows the disclosed vulnerabilities for the three OSs from 2015 to 2021. The cumulative vulnerabilities for windows 10 have been increasing since 2015, while that for Ubuntu have fluctuated over the years, dropping in 2017 and shooting to the highest in 2018, before dropping further to the lowest for the three OSs.

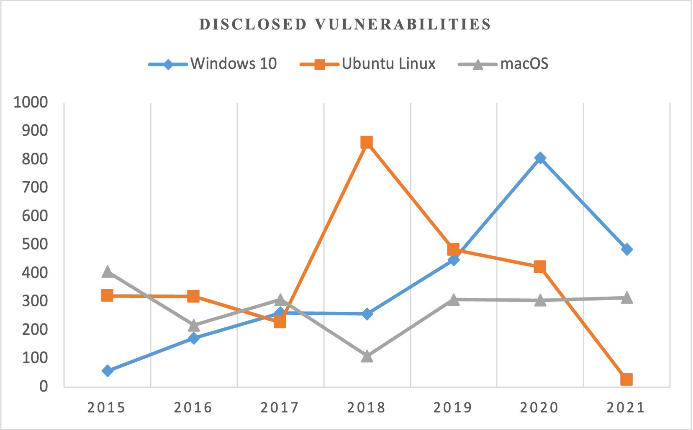

Figure 2: Disclosed vulnerabilities for windows 10, Ubuntu, and macOS (2015 to 2021)

4.2. Vulnerability Scores for Operating Systems – CVSS

Table 4 provides the weighted average vulnerability level for the three operating systems for the study from the CVE databases for the period under study. The weighted average score for each OS was retrieved from the CVE website by searching each year from January to December. Each CVSS score (0-1, 1-2, 2-3…9-10) is assigned the reported number of vulnerabilities and a percentage which are then utilized to compute the weighted average CVSS score. For example, the data for computing the 2015 weighted average for windows 10 can be found in [34]. The process is repeated for the entire period for all the OSs. The average vulnerability score for the study period is computed utilizing a similar approach.

Table 4 and Figure 3 shows the average vulnerability severity level by aggregating common vulnerability scoring system (CVSS) from the CVE database. The figures illustrate macOS to have higher vulnerability scores while Ubuntu has the lowest average for the study period. Vulnerabilities of the Ubuntu Linux are least critical with an average of 6.0 followed by Windows 10 and lastly macOS.

Table 4: Weighted average vulnerability severity for OSs from CVE (CVSS Score)

Year	Windows 10	Ubuntu Linux	macOS
2015	7.6	6.6	7.0
2016	7.5	6.0	7.6
2017	5.4	6.2	7.3
2018	5.8	6.0	7.1
2019	6.9	6.0	7.0
2020	6.2	5.4	6.9
2021	5.7	4.9	6.7
Weighted Average CVSS score	6.2	6.0	7.0

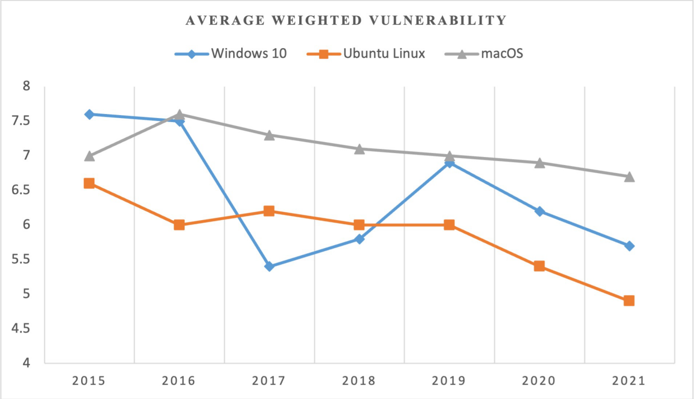

Figure 3: Average weighted vulnerability for the OS

4.3. Disclosed Vulnerability vs Average Weighted Vulnerability

Correlation between disclosed vulnerabilities and average weighted vulnerabilities for the study period yielded coefficient of -0.4081, 0.3473, and -0.3674 for windows, ubuntu, and macOS respectively. The coefficients suggest Ubuntu Linux OS as having the lowest severities while windows has the highest security severities resulting from their vulnerabilities during the study period.

4.4. Vulnerability Severity

The impact of a vulnerability on the integrity, confidentiality, and security of a system is described as vulnerability severity. Vulnerability severity is quantified using the CVSS systems that assigns a score from zero (least severe) to ten (most severe) [35]. A CVSS score is computed from a combination of various metrics including the easiness of exploitation of a vulnerability and its impact (https://www.first.org/cvss/specification-document).

The calculation of the vulnerability severity score utilized in this study is achieved utilizing three metric group – base, temporal, and environmental metrics. The base metric generates a vulnerability score from 0 to 10 and is intrinsic to a vulnerability and does not change. The base score is then modified by scoring environmental and temporal metric. Temporal metrics are those that change over the lifetime of a vulnerability while environmental metric consider the specific environment where the vulnerability exists (www.balbix.com).

NVD uses two ratings of severity scores namely CVSS v2.0 and the CVSS v3.0. Under CVSS v2.0, a range of 0.0 to 3.9 is classified as low, 4.0-6.9 as medium, and 7.0-10.0 (High). In CVSS v3.0, 0.0 denotes no vulnerability, 0.1-3.9 (low), 4.0-6.9 (medium), 7.0-8.9 (High), and 9.0-10.0 (Critical) 35.

Table 4 presents the average severity for the three operating systems for each year and the average for the study period.

Table 5 provides data for the different severity levels for the three operating systems for the study period.

Table 5: Number of vulnerabilities by severity score

Year	OS	Number of vulnerabilities by severity score									
		0-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10
2015	Windows 10	0	1	6	0	5	0	7	21	0	17
	Ubuntu	0	3	12	13	60	72	48	88	2	23
	macOS	0	5	22	6	48	52	107	118	3	46
	Total	0	9	40	19	113	124	162	227	5	86
2016	Windows 10	0	3	18	4	18	6	12	54	0	57
	Ubuntu	0	9	22	11	103	61	47	41	0	25
	macOS	0	1	11	2	37	17	30	39	3	78
	Total	0	13	51	17	158	84	89	134	3	160
2017	Windows 10	0	52	36	4	56	7	36	45	2	24
	Ubuntu	0	0	8	2	84	42	39	51	0	2
	macOS	0	0	22	0	62	21	65	44	0	94
	Total	0	52	66	6	202	70	140	140	2	120
2018	Windows 10	0	26	46	4	63	7	29	54	1	28
	Ubuntu	0	9	52	19	295	151	159	164	2	9
	macOS	0	0	9	0	26	9	17	20	0	29
	Total	0	35	107	23	384	167	205	238	3	66
2019	Windows 10	0	3	64	4	97	28	21	111	3	117
	Ubuntu	0	4	25	25	147	100	81	92	1	9
	macOS	0	1	22	2	56	51	71	30	1	74
	Total	0	8	111	31	300	179	173	233	5	200
2020	Windows 10	0	1	104	11	310	16	79	209	1	76
	Ubuntu	0	7	42	25	179	78	55	29	0	8
	macOS	0	1	20	4	73	32	72	34	0	70
	Total	0	9	166	40	562	126	206	272	1	154
2021	Windows 10	0	0	72	9	192	37	97	64	0	14
	Ubuntu	0	1	10	2	4	1	0	7	0	0
	macOS	0	1	18	0	91	25	103	22	2	53
	Total	0	2	100	11	287	63	200	93	2	67
Sum-Total	Windows 10	0	86	346	36	741	101	281	558	7	333
	Ubuntu	0	33	171	97	872	505	429	472	5	76
	macOS	0	9	124	14	393	207	465	307	9	444
	Total	0	128	641	147	2006	813	1175	1337	21	853

Figure 4 shows OS distribution by severity levels. All the OS have the highest quantity of vulnerability at a severity level between 4 and 5. Ubuntu has the lowest critical level severity while macOS has the highest numbers. No operating system has a score between 0 and 1.

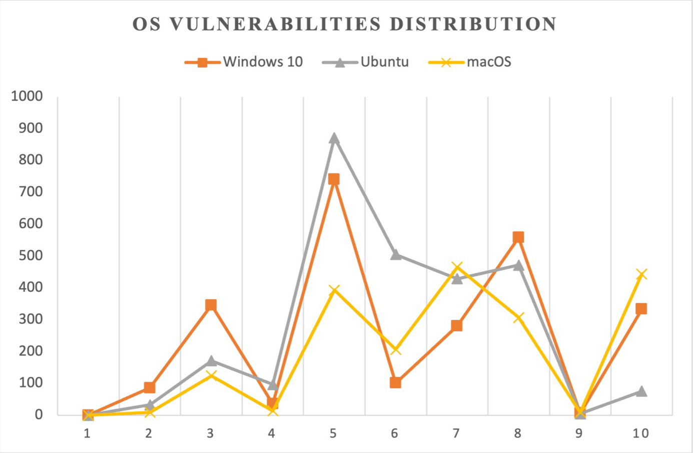

Figure 4: OS vulnerabilities distribution by CVSS severity level

4.5. The Most Common Vulnerabilities

OSs vulnerabilities are classified using the CWE scheme by NVD as proposed by MITRE. The study identified common vulnerabilities as CWE-119, CWE-19, and CWE-20 for macOS and Ubuntu and CWE-119, CWE-19, CWE-20, CWE-281 for Windows 10.

- MITRE Corporation describes CWE-119 as the "improper restriction of operations within the bounds of a memory buffer". The "software performs operations on a memory buffer, but it can read from or write to a memory location that is outside of the intended boundary of the buffer".[36]
- CWE-19 – Weaknesses in the processing of data
- CWE-20 – described as "improper input validation which may result in altered control flow, arbitrary code execution or illegal access to and control of resources"
- CWE-281 – Weakness in the proper presentation of permissions as the software fails to preserve permission or incorrectly preserves it during copying, restoring, or sharing of objects.

5. Discussion and Conclusion

The rise in cyber-attacks in recent years is demonstrated by the findings of this study. Using the operating system vulnerability data from CVE and NVD, this study has shown that operating systems of different versions continue to be attacked and exploited due to their vulnerabilities. However, Windows 10 was attacked more than other OS, demonstrating the impact that a high number of users utilizing a particular OS could have on their long-term security and integrity.

Cybercriminals and attackers exploit these weaknesses to exploit these operating systems to the detriment of the users. Between 2015 and 2021, Ubuntu reported the highest vulnerability (2660), followed by Windows 10 (2489), and lastly macOS (1972). MacOS seems to be most secure of the three operating, but not exempted from further exploitation. The low vulnerability may also suggest low disclosure rates by Apple compared to the other developers. Among the three OSs, code execution was the most common type of vulnerability followed by DoS. Nonetheless, Ubuntu had the lowest vulnerability score while macOS had the highest suggesting that the few vulnerabilities reported for macOS were very serious compared to the many reported for Ubuntu or Windows. The implication of this study is that OS developers and companies need to enhance the security of their products extending findings by [3]. A change in software development processes and practices for disclosing vulnerabilities needs to be effected to enhance the security, confidentiality, and integrity of these systems and, therefore, the user's data. While this study focused on three OS, future work can study more than one OS of different versions using data from multiple sources to develop relationships between vulnerabilities, actual executed exploits, and system safety rating for a given period.

References

[1] D. Palmer, Ransomware demands are growing, but life is getting tougher for malware gangs, ZDNET, 2022.

[2] Kelley Karin, Vulnerability in Security: A Complete Overview, Simplilearn, 2022.

[3] A. Gorbenko, A. Romanovsky, O. Tarasyuk, O. Biloborodov, "From Analyzing Operating System Vulnerabilities to Designing Multiversion Intrusion-Tolerant Architectures," IEEE Transactions on Reliability, **69**(1), 22–39, 2020, doi:10.1109/TR.2019.2897248.

[4] A. Al-Boghdady, K. Wassif, M. El-Ramly, "The Presence, Trends, and Causes of Security Vulnerabilities in Operating Systems of IoT's Low-End Devices," Sensors, **21**(7), 2329, 2021, doi:10.3390/s21072329.

[5] L. Li, W. He, L. Xu, I. Ash, M. Anwar, X. Yuan, "Investigating the impact of cybersecurity policy awareness on employees' cybersecurity behavior," International Journal of Information Management, **45**, 13–24, 2019, doi:10.1016/j.ijinfomgt.2018.10.017.

[6] I. Astaburuaga, A. Lombardi, B. la Torre, C. Hughes, S. Sengupta, “Vulnerability Analysis of AR.Drone 2.0, an Embedded Linux System,” in 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), IEEE: 0666–0672, 2019, doi:10.1109/CCWC.2019.8666464.

[7] M. Vasek, J. Wadleigh, T. Moore, “Hacking Is Not Random: A Case-Control Study of Webserver-Compromise Risk,” IEEE Transactions on Dependable and Secure Computing, **13**(2), 206–219, 2016, doi:10.1109/TDSC.2015.2427847.

[8] L. Tung, FBI warning: This ransomware uses DDoS to threaten victims. Here’s what to watch out for, ZDNET, 2022.

[9] L. Bilge, T. Dumitras, “Before we knew it,” in Proceedings of the 2012 ACM conference on Computer and communications security - CCS ’12, ACM Press, New York, New York, USA: 833, 2012, doi:10.1145/2382196.2382284.

[10] S. Frei, M. May, U. Fiedler, B. Plattner, “Large-scale vulnerability analysis,” in Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense - LSAD ’06, ACM Press, New York, New York, USA: 131–138, 2006, doi:10.1145/1162666.1162671.

[11] J. Ruohonen, S. Hyrynsalmi, V. Leppänen, “Software Vulnerability Life Cycles and the Age of Software Products: An Empirical Assertion with Operating System Products,” in Conference: Advanced Information Systems Engineering Workshops, 207–218, 2016, doi:10.1007/978-3-319-39564-7_20.

[12] A. Raza, W. Ahmed, “Threat and Vulnerability management life cycle in operating systems. A systematic review,” Journal of Multidisciplinary Engineering Science and Technology (JMEST), **9**(1), 2022.

[13] S. Farhang, J. Weidman, M.M. Kamani, J. Grossklags, P. Liu, “Take It or Leave It,” in Proceedings of the 34th Annual Computer Security Applications Conference, ACM, New York, NY, USA: 490–504, 2018, doi:10.1145/3274694.3274733.

[14] F. Alharbi, Y. Zhou, F. Qian, Z. Qian, N. Abu-Ghazaleh, “DNS Poisoning of Operating System Caches: Attacks and Mitigations,” IEEE Transactions on Dependable and Secure Computing, **19**(4), 2851–2863, 2022, doi:10.1109/TDSC.2022.3142331.

[15] M. Araba Vander-Pallen, P. Addai, S. Isteefanos, T. Khan Mohd, “Survey on Types of Cyber Attacks on Operating System Vulnerabilities since 2018 onwards,” in 2022 IEEE World AI IoT Congress (AIIoT), IEEE: 01–07, 2022, doi:10.1109/AIIoT54504.2022.9817246.

[16] F. Alharbi, J. Chang, Y. Zhou, F. Qian, Z. Qian, N. Abu-Ghazaleh, “Collaborative Client-Side DNS Cache Poisoning Attack,” in IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, IEEE: 1153–1161, 2019, doi:10.1109/INFOCOM.2019.8737514.

[17] R. v. Deshmukh, K.K. Devadkar, “Understanding DDoS Attack &amp; its Effect in Cloud Environment,” Procedia Computer Science, **49**, 202–210, 2015, doi:10.1016/j.procs.2015.04.245.

[18] S.-P. Oriyano, R. Shimonski, Client-Side Attacks Defined, Elsevier: 1–24, 2012, doi:10.1016/B978-1-59-749590-5.00001-8.

[19] C. Cowan, F. Wagle, Calton Pu, S. Beattie, J. Walpole, “Buffer overflows: attacks and defenses for the vulnerability of the decade,” in Proceedings DARPA Information Survivability Conference and Exposition. DISCEX’00, IEEE Comput. Soc: 119–129, doi:10.1109/DISCEX.2000.821514.

[20] K.-S. Lhee, S.J. Chapin, “Buffer overflow and format string overflow vulnerabilities,” Software: Practice and Experience, **33**(5), 423–460, 2003, doi:10.1002/spe.515.

[21] J. Xu, P. Ning, C. Kil, Y. Zhai, C. Bookholt, “Automatic diagnosis and response to memory corruption vulnerabilities,” in Proceedings of the 12th ACM conference on Computer and communications security - CCS ’05, ACM Press, New York, USA: 223, 2005, doi:10.1145/1102120.1102151.

[22] M. Junjin, “An Approach for SQL Injection Vulnerability Detection,” in 2009 Sixth International Conference on Information Technology: New Generations, IEEE: 1411–1414, 2009, doi:10.1109/ITNG.2009.34.

[23] A. Shrivastava, S. Choudhary, A. Kumar, “XSS vulnerability assessment and prevention in web application,” in 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), IEEE: 850–853, 2016, doi:10.1109/NGCT.2016.7877529.

[24] M. Liu, B. Zhang, W. Chen, X. Zhang, “A Survey of Exploitation and Detection Methods of XSS Vulnerabilities,” IEEE Access, **7**, 182004–182016, 2019, doi:10.1109/ACCESS.2019.2960449.

[25] M. Flanders, “A Simple and Intuitive Algorithm for Preventing Directory Traversal Attacks,” ArXiv, 2019.

[26] D. Kshirsagar, S. Kumar, L. Purohit, “Exploring usage of ontology for HTTP response splitting attack,” in 2015 1st International Conference on Next Generation Computing Technologies (NGCT), IEEE: 437–440, 2015, doi:10.1109/NGCT.2015.7375156.

[27] A. Atamli-Reineh, R. Borgaonkar, R.A. Balisane, G. Petracca, A. Martin, “Analysis of Trusted Execution Environment usage in Samsung KNOX,” in Proceedings of the 1st Workshop on System Software for Trusted Execution, ACM, New York, NY, USA: 1–6, 2016, doi:10.1145/3007788.3007795.

[28] J. Softic, Z. Vejzovic, “Windows 10 Operating System: Vulnerability Assessment and Exploitation,” in 2022 21st International Symposium INFOTEH-JAHORINA (INFOTEH), IEEE: 1–5, 2022, doi:10.1109/INFOTEH53737.2022.9751274.

[29] M.R. Yaswinski, M.M. Chowdhury, M. Jochen, “Linux Security: A Survey,” in 2019 IEEE International Conference on Electro Information Technology (EIT), IEEE: 357–362, 2019, doi:10.1109/EIT.2019.8834112.

[30] n.d., NVD Vulnerabilities, NVD, 2022.

[31] n.d. - CVE, Canonical Ubuntu Linux: CVE security vulnerabilities, versions and detailed reports, CVE, 2022.

[32] n.d. - CVE, Microsoft Windows 10: CVE security vulnerabilities, versions and detailed reports, CVE, 2022.

[33] n.d.-CVE, Apple Mac Os X : CVE security vulnerabilities, versions and detailed reports, CVE, 2022.

[34] n.d.-CVE, Computing Weighted average for Windows 10 , CVE, 2022.

[35] n.d.-NVD, NVD Vulnerability Metrics, CVE, 2022.

[36] n.d., Improper Restriction of Operations within the Bounds of a Memory Buffer (4.8), CWE, 2022.