Threat and Vulnerability management life cycle in operating systems. A systematic review

Ali Raza
alirazachaddar@gmail.com
Department of Computer Science, Bahria
University Karachi Campus, Karachi, Pakistan.

Waseem Ahmed
w.ahmed555@yahoo.com
Department of Computer Science, Bahria
University Karachi Campus, Karachi, Pakistan

***Abstract*— A considerable number of security threats and vulnerabilities are reported in operating systems largely in Windows, and Linux every month. Operating system users and developers have to utilize a lot of resources to evaluate the threat level possessed by the vulnerability and to mitigate the same. Vulnerability discovery models are employed by every OS vender to elaborate the vulnerability lifecycle for efficient and timely resources allocation to complete the vulnerability lifecycle as much as possible before the exploitation by hackers. Every software product has its vulnerability lifecycle as vulnerabilities are discovered throughout it life span, by it developer, dedicated team of tester and some vulnerabilities are reported by the end user. When a vulnerability is discovered, it goes through different stages i.e. evaluation, resource allocation and release of patch as soon as possible. Recourse's allocation is performed by considering the severity of the vulnerability. Different operating systems are following different type of vulnerability life cycle to provide support to their users before vulnerability exploitation. In this paper, we have performed a systemic review to study the existing literature regarding the vulnerability life cycles followed by different operating systems**

Keywords*— *Vulnerability, Exploitation, SQL injection, Zero-day, Cross site scripting, Operating System Vulnerabilities, Windows Vulnerabilities, Vulnerability Lifecycle

I. Introduction

In this technological era, information security has gained a significant importance and has become one of the most critical matters for end users, developers and venders of application/ system software and IT infrastructures. Statistics show that businesses are striving to show their presence online and grab as much as possible share of its consumers as the online of world has grown to 4,208,571,287. The highest number (49%) of internet users are from Asia, Europe has the second largest number (16.8%), followed by Africa with 11% and at last North America has 8.2% share of total internet users. The total number of websites on the internet is 1.95 billion [1]. In a recent survey, sale of retail e-commerce has the highest number about 4.28 trillion US dollars is the history and same is project to grow to 5.4 trillion US dollars in 2022 [2]. With the increase of internet users and higher stakes of businesses, integrity of their data and least downtime to make sure their availability to their costumer has become evident. Operating systems, being the base component to every platform, should guarantee solutions to the above-mentioned concerns of the users and businesses. The major threats to the modern Operating Systems are Memory overflow, Distributed Denial of service (DDoS), Memory Corruption, XSS (Cross site scripting), Code Execution, Directory Traversal, IDS or IPS Bypass, HTTP response splitting, gaining privileges/ information and escalating the already gained privileges [3].

Vulnerability is a loophole or weakness in any software due to poor design and implementation. These weaknesses, when exposed to hackers can compromise computer systems, networks, applications, and databases [4]. Hackers gain limited access to systems by exploiting vulnerability available in operating system and escalate their privileges to super user or administrator so that they can deeply damage the under-attack system or get full access to organizational databases and other network resources. Any vulnerability has its life cycle from discovery of vulnerability to its patch release. Different organizations deploy different vulnerability life cycles for the rectification of the loophole in their systems to minimize the risk to user system and data. In this paper, we are going to take a comprehensive look at vulnerability life cycles followed by two major operating systems developing companies Microsoft windows and Linux operating systems.

A. OS Vulnerability Life Cycle

Vulnerability discovery life cycle of different operating systems can be defined as the set of different stages through which a vulnerability goes after its disclosure. This section of paper contains review some of the definitions regarding software vulnerability lifecycle. During the studies of available literature, [1] is one of the pioneer studies that defined the vulnerability discovery life cycle of Operating Systems as a set of events. These set of events are defined from the birth, the discovery (merged with the birth of vulnerability as discovered by the internal testing/QA team), vulnerability disclosure (defined as the information of vulnerability is disclosed to only security insiders), the release of the patch, the full disclosure of the vulnerability (more global than in the case of the disclosure event), the exploit availability (the vulnerability can exploit by the hackers), and the death of the vulnerability. In [7] the vulnerability is defined as full discloser whereas patches release and testing are not the part of vulnerability lifecycle. However, it is assumed that vulnerability can only be exploited after its disclosure and same is available for exploit before the patch release. In [8], authors defined an additional term "the publicity event" which is described as the moment when the already disclosed vulnerability is known by a large population. Whereas this event is comparable to the full disclosure of vulnerability already defined in [1]. Some additional vulnerability lifecycle events are suggested in [6], the vulnerability rediscovery and large-scale propagation of its information. Furthermore, exploit event cover concept discovery, successful attack event and automation of exploit to enlarge its effect to large number of victims whereas the "patch management is covers patch release and its complete release to the end users.

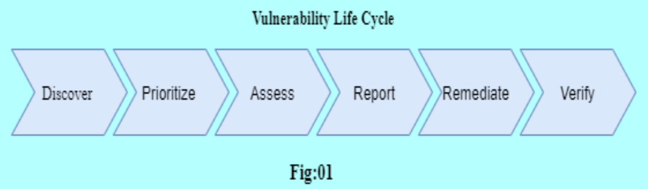

Fig:01

The vulnerability life cycle can be seen in fig 1.

II. Literature Review

During review of present literature, vulnerability exposure lifecycle events discussed in previous section are classified, 308 vulnerabilities are analyzed in [6] and focused on vulnerability exposure on attack process. Whereas an economical model was proposed after analyzing the statistics of 308 vulnerabilities to evaluate the number of attacks per host and per day. As set 240 vulnerabilities are studies in [7], the vulnerability lifecycle of these 240 vulnerabilities was completed. Firstly, each vulnerability lifecycle was classified into Zero-day attack or potential risk category, these vulnerabilities exposed and available for exploitation and the patch was not released yet. Secondly, a metric was assessed for each of these vulnerabilities, considering eight attributes such as the age of the vulnerability, the potential risk involved and the exploit existence. The results shown by this economical matrix highlighted that there a lack of patches for Microsoft vulnerabilities.

14326 number of vulnerabilities from different databases was investigated in [7], the work focused on the characterization of the probability distribution of time interval between availability of vulnerability exploit with release of patch with relevance to vulnerability disclosure. Furthermore, no specific characteristics and attributes of these vulnerabilities were discussed in this study, but the author focused on defining zero-day patch metric in this work. This zero-day patch matric also measures the proportion of vulnerabilities for which the patch and the vulnerability are disclosed at the same time [8].

During the review of available literature, It is noted that a number of researchers focused on the security and vulnerability life cycle impacts on systems. The one of the latest studies is [9], the authors used vulnerability and security incidents recorded from real computing systems deployed in different organizations. Furthermore, they grouped these incidents into nine categories and linked it to vulnerability classes. However, the analyses of these vulnerabilities are aimed to measure effective detection of tools used in experiments and are not connected to the vulnerability management life cycle. Some studies report the experimental analysis between vulnerability life cycle event and their connection with characteristics of the vulnerability. Same type of the work is done in [10], in this paper vulnerability characteristics are correlated and probable association with the reoccurrence of vulnerability life cycle events. Then they devised a model allowing the quantitative assessment of security measures considering a global set of vulnerabilities without distinguishing their specific attributes like type of operating system, severity of vulnerability. Moreover, the results presented in this paper evaluated the specific attributes in the analysis of time interval between the vulnerability life cycle events.

III. Methodology

This systematic review was conducted by following the reporting checklist of the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA; [11]). A systematic review of research papers from IEEE explorer, Springer link and ACM Digital Library on "Threat and Vulnerability management life cycle in different operating systems" was performed to identify how scholars and security professionals in the field of Cyber Security proposed the techniques and method to safeguard different operating systems from the latest threats and attack from hackers.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

A. Research Questions and Search String

The purpose of this systematic review is to identify studies and research covering the ways to discover vulnerabilities and how these vulnerabilities are managed in different operating systems. In this systematic review, we included only two operating systems (Windows by Microsoft and Linux by Linus Torvalds). We only selected three literature databases (IEEE explorer, Springer link and ACM Digital Library) and the studies from 2010 to present are included in this review. We formulated the bellow search string to extract the related studies in our review and devised research questions, which satisfies our research topic. To conduct this research and study, we formulated the following research string,

("Full Text & Metadata": vulnerability lifecycle in different operating systems OR vulnerabilities in different operating systems OR vulnerabilities in windows operating system OR vulnerabilities in Linux operating system.)

To select the relevant studies from the available literature, search with the help of the above-mentioned search string, the following research questions are devised.

RQ1: What are the latest studies on vulnerability analysis in different operating systems?
RQ2: How operating system's vulnerabilities examined and discovered?
RQ3: What are the latest techniques and technologies to remediate these vulnerabilities?
RQ4: What are different stages in vulnerability lifecycle followed by different operating systems?
The period of publications was set from 2000 and 2022. The number of research articles in a specific literature database is tabulated in Table # 01 and in Fig:01.

Table:01

S. No	Database	Number of Studies
1.	IEEE Explore	203
2.	Springer Link	128
3.	ACM Digital Library	73
	Total	404

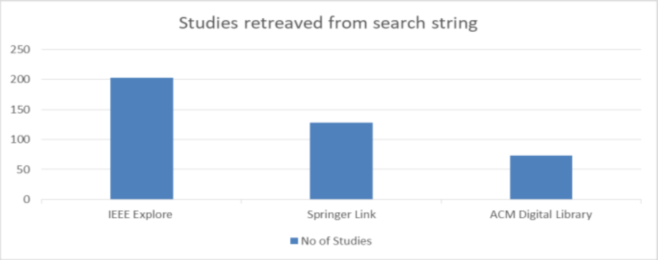

Fig:01

Following inclusion and exclusion criteria was formulated for the selection of studies according to our research topic to carry a systematic review.

Table 2. The Inclusion and Exclusion Criteria.

Code	Inclusion Criteria
IC1	Papers discussing operating systems vulnerabilities.
IC2	Papers suggesting vulnerability detection techniques.
IC3	Paper discussing vulnerabilities lifecycle and patch management.
IC4	Paper should be between 2010-2021.
IC5	Paper should be written in English Language.
Code	**Exclusion Criteria**
EC1	Papers not addressing vulnerabilities in operating systems.
EC2	Paper not addressing techniques to vulnerabilities life cycle management.
EC3	Paper addressing windows and Linux vulnerabilities.
EC4	Gray literature and book chapters.

B. *Selection Criteria*

The search string was applied to three literature databases; IEEE Explore, Springer Link and ACM Digital Library and total 404 studies enlisted and exported to Excel worksheet for further analysis. This search was conducted in Jan 2022. The identified records of 404 studies were exported to an MS Excel in three different sheets for three databases for further analysis and applied inclusion and exclusion criteria on "Title", "Abstract" and "keywords" section of papers. A new column name "Inclusion and Exclusion (IN/EX Criteria)" markings, digit 0 appended to exclude the paper from review and 1 appended to include the paper in review. Following number of papers passed/ failed the inclusion and exclusion criteria.

IV. Threat and Vulnerability Management

From the selected studies the finding regarding the vulnerabilities disclosure in today's operating systems along with its increasing tendency toward and severity reveal the serious security challenges and risks that effects the OS developers and users adversely. Security experts should not only consider the critical factors affecting system security as total number of vulnerabilities disclosed in a particular operating system, but the time it takes by the vender to release patches fixing disclosed vulnerabilities. The average days-of-risk (before the patch of disclosed vulnerability) for the studied operating systems differs from 83 days for Oracle Solaris up to 135 days for Red Hat [12].

Furthermore, it is matter of great concern that the venders of multiple operating systems does not issue patches according to the severity of vulnerability. The average days-of-risk for the most critical operating systems vulnerabilities remains higher (117 vs 100 days) then the less critical vulnerability. Patches covering the most critical vulnerabilities should be issues as early as possible and it should be make sure that all users using the respective operating systems version has installed these patches before the vulnerability exploitations. The fact exposes deficiencies in the policies for developing security patches adopted by the operating systems vendors, as well as, in the vulnerability lifecycle management process they run [12].

Vulnerability statistics of over 26 years in different operating systems (Linux) products and versions were analyzed in [13]. The authors discussed two security enhancement (hardening and LSM) briefly and they studied various Linux vulnerabilities. They suggested that in securing the Linux operating systems we still have a long way to go. We need to implement advanced technologies or products to obtain the greater security furthermore we always keep probability of some new security issues. Therefore, it is essential to understand the existing vulnerabilities and the attack patterns to protect existing operating systems and provide more secure advanced services.

Many operating systems vulnerability discovery models are tested in [14] using AIC (Akaike Information Criteria), and chi-square tests. The assessment of these tests resulted that the AML (Alhazmi-Malaiya Logistic Model) model is the best for the longer term for the systems such as Windows 95, Red Hat Linux 6.2, and Red Hat Fedora. RQ (Rescorla Quadratic Model) is proved to be good for Windows XP. Furthermore, the author suggested that the developers and users need to be used vulnerability discovery models regularly. Developers can estimate the upcoming vulnerabilities by assessing product readiness with respect to the vulnerabilities. They should allocate security maintenance resources to discover the vulnerabilities, preferably before the hackers do, and should release security patches as soon as possible before the exploitation of the vulnerability. Moreover, the users of operating systems should also evaluate the risk due to vulnerabilities ahead of patches release and application. The security patch release by concern vender should be tested for its stability before release to the end users.

A detailed study of operating systems vulnerabilities reported in Open-Source Vulnerabilities Databases was carried out in [15]. They studied the vulnerabilities of Windows, UNIX, and Mobile OS, and found that the mobile operating system's vulnerabilities have a short life cycle on average it takes 14 days from discovery of vulnerability and its patch release. The Windows vulnerability life cycle show different characteristics when compared to UNIX vulnerability life cycle. It is noted that disclosure of the vulnerabilities is intentionally delayed near to the patch release date. That is the reason that many Windows users' remains unaware of security risks even after the discovery of vulnerabilities, consequently Windows operating system has become the most favorite operating system for hackers to perform their exploitations. But this gap between discloser and patch release is decreasing in recent Windows operating systems which shows that recent Windows version vulnerabilities are being patched faster. Furthermore, it is noted form the results that remote access vulnerabilities are exploited faster than the local vulnerabilities.

V. Conclusion

In this paper, we studied the exiting literature regarding vulnerability lifecycle management followed by different operating systems. As we know that a software vulnerability always goes from different stages from discovery to release of patches. Operating system being the base system for today's information era, need to prioritize highly according to its importance. Furthermore, in exiting vulnerability lifecycles followed by different venders of different operating systems does not prioritize their resources according to the severity of vulnerability by considering their impact of confidentiality, integrity of user's data and privacy and availability of system/ services to the legitimate users. It has been observed from literature that remote vulnerabilities are exploited early then the local vulnerabilities, so remote vulnerabilities should always be allotted highest priority.

Furthermore, it would be interesting to distinguish vulnerabilities targeting the kernel of the OS and the vulnerabilities targeting the applications. Moreover, vulnerabilities targeting software's developed by commercial or non-commercial developers can also be categories to get the interesting results.

VI. References

[1] W. F. a. J. M. W. Arbaugh, "Windows of vulnerability: a case study analysis.," 2000.

[2] wpforms.com, "https://wpforms.com/the-ultimate-list-of-online-business-statistics/," [Online]. [Accessed 23 Nov 2021].

[3] S. Chevalier, "https://www.statista.com," 7 july 2021. [Online]. [Accessed 21 November 2021].

[4] "www.cvedetails.com," www.cvedetails.com, [Online]. Available: https://www.cvedetails.com/vendor/26/Microsoft.htm. [Accessed 19 December 2021].

[5] A. K. V. S. Gaurav Sharma, "Windows Operating System Vulnerabilities," *international journal of computing and corporate research,* 2011.

[6] R. K. R. T. a. Y. Y. A. Arora, "Impact of vulnerability disclosure and patch availability - an empirical analysis.," in *in In Third Workshop on the Economics of Information Security, 2004.*, 2004.

[7] M. M. U. F. a. B. P. S. Frei, "Large-scale vulnerability analysis," in *Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense*, 2006.

[8] B. T. a. B. P. S. Frei, "0-day patch - exposing vendors (in) security performance," *https://www.blackhat.com/,* 2008.

[9] Z. K. J. B. a. R. I. A. Sharma, "Analysis of security data from a large computing organization," 2010.

[10] G. Vache, " Vulnerability analysis for a quantitative security evaluation," *Proceedings of the International Symposium on Empirical Software Engineering and Measurement IEEE Computer Society,* 2009.

[11] A. L. J. T. a. D. G. A. David Moher, "Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement," *thebmj* , p. 1, 2019.

[12] A. R. O. T. a. O. B. Anatoliy Gorbenko, "Experience Report: Study of Vulnerabilities of Enterprise Operating Systems," in *IEEE 28th International Symposium on Software Reliability Engineering,* 2017.

[13] J. M. Z. Z. a. Z. L. Shuangxia Niu, "Overview of Linux Vulnerabilities," *International Conference on on Soft Computing in Information Communication Technology (SCICT),* 2014.

[14] O. H. A. a. Y. K. Malaiya, "Application of Vulnerability Discovery Models to Major Operating Systems," *IEEE TRANSACTIONS ON RELIABILITY,* 2008.

[15] V. N. M. K. Geraldine Vache Marconato, "Security-related vulnerability life cycle analysis," *HAL Open Science,* 2013.