A Study on Vulnerability Management

Piyush Somani[1], Poornima Kulkarni[2]

[1]Student, Department of Information Science and Engineering, RV College of Engineering, Bangalore, India
[2] Assistant Professor, Department of Information Science Engineering, RV College of Engineering, Bangalore, India

---***---

Abstract - Vulnerability Management is a pervasive problem in the development of any codebase. In the basic terms conceivable, a cyber vulnerability is any inaccuracy, shortcoming, or defect in an information system, internal control systems, or system processes of an organisation. It might be referred to as an imperfection or deficiency in the architecture of the code base which produces application malfunctions. Accordingly, it is important to implement a robust vulnerability management measure to avoid widespread assaults or even to mitigate the damage inflicted by a cyberattack. In this work, a Vulnerabilities Management System (VMS) solution is proposed.

***Key Words*: Software vulnerabilities, vulnerability management, vulnerability database, and vulnerability management system**

1. INTRODUCTION

Presently, firmware which governs an electronic device's operation are incorporated through every electrical appliance. Those application programs, developed by various engineers with such a smaller percentage of code, might well be enormously complicated and yet are later shown in a comprehensive technology program or project. Almost majority of cases, application software errors are caused from lingering problems or vulnerabilities inside the code that can generate unforeseen consequences. The computer program is vulnerable to the impact of this flaw inside the programming language. Software vulnerabilities therefore are discovered in application software or software platforms that also have unresolved issues, flaws, or vulnerabilities. Eventually, such program vulnerability provides a good example of the a point of entry into a software platform, which can cause significant harm to the system, that both computer hosting the software as well as the device associated towards the infected system

Notwithstanding in all security protocols, as even more individuals are browsing the network, vulnerabilities are indeed being identified at an accelerating rate. Source code abnormalities, inconsistencies, and flaws can develop in just about any device that seems to have programming functionalities; consequently, detection mechanisms should emerge both for attempting to resolve and mitigating software vulnerabilities. that both the computer hosting the software as well as the device associated with the infected system.

Notwithstanding in all security protocols, as even more individuals are browsing the network, vulnerabilities are indeed being identified at an accelerating rate. Source code abnormalities, inconsistencies, and flaws can develop in just about any device that seems to have programming functionalities; consequently, detection mechanisms should emerge both for attempting to resolve and mitigating software vulnerabilities.

A hacker continually aims to break into the a system and acquire credentials that would give them access to sensitive data or resources. A cyberattack potentially lead to significant financial destruction, damage to someone's character, including loss of irreplaceable information. If this programming structure is subjected to penetration testers, then susceptibility areas must be detected.

Figure 1 depicts a brief overview of vulnerability management has to be implemented.

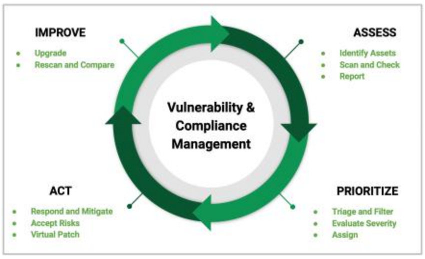

Fig 1. A Brief Overview of Vulnerability Management

1.1 Sub Heading 1

2. RELATED WORK

This section focuses on research which has already been conducted. The existing work will serve as a foundation for something like the construction of a new vulnerable management system featuring advanced benefits.

The author in [1] provides a novel vulnerability management solution, known as the Software Vulnerability Integrated Management System (SV-IMS), has indeed been introduced by that of the authors in this study. That program can operate security screening to identify program weaknesses, as well as the outcomes of these kinds of tests could be seen on a different platform. Furthermore, it specifies the Popularly Known Scoring Scheme , a worldwide ratings system that evaluates the gravity of bugs.

The author in [2] describes Spring Boot which is a java-based framework for building web and enterprise applications and how it provides the flexibility for service-oriented architecture (SOA). As a result, this paper suggests that the SOA based REST API using Spring Boot Framework has definite advantages over other spring-based frameworks.

The author in [3] highlights the necessity of accurate and comprehensive information being provided in bug reports intended to assist inside the quicker response of errors. Nevertheless, following multiple cycles of interaction among correspondents and programmers, relevant information frequently trickles to developers. Inefficient debugging methods are partially to blame for the prolonged information sharing. By recommending four main routes for improvements, it tackles the issues with bug tracking systems. It also exhibits a proof-of-concept interactive bug tracking system that asks users for pertinent information and identifies files that need to be updated in order to remedy the fault.

In [4] this research study, the author discusses Common Vulnerability Exposure (CVE) and Common Security Vulnerabilities Languages, two global, community-based efforts involving business, the public sector, and research. Although OVAL seeks to provide enough methods for comprehensive vulnerability analysis as well as result in standardized reporting information security standards for networks, CVE determines the best method for generating vulnerability notifications increasingly appropriate to different organisations.

The authors in [5] describe the tools for project management and issues/bugs tracking that are becoming useful for governing the development process of Open Source software. Such tools simplify the communications process among developers and ensure the scalability of a project. The more information developers are able to exchange, the clearer are the goals, and the higher is the number of developers keen on joining and actively collaborating on a project.

The author [6] established a method that made use of a learning algorithm and depth of knowledge. Through using this, the cyber risk management identifies, assesses, and negates the problem automatically. Challenges, security risks, reputational harm, and Economic loss all are minimized by the proposed approach.

The author in [7] provides information in order to identify the vulnerabilities for injection attacks, the author of this research has created an automated vulnerability scanner. The webpage is continuously analysed by this system for Cross - site request and Sql injection attacks. The Nationwide Vulnerable Database, or Dynamic model, is also another component of the proposed system.

The author in [8] provides data by building a vulnerability database, the author of this research study has developed a new technique for creating and managing vulnerabilities. A new National Vulnerability Dataset (NDV) platform may be used by different companies in this proposed study.A recently found bug could also be registered inside the Needs to request database while still being referred to.

In [9] the SCADA system's vulnerability is evaluated to use a paradigm that this researcher has developed in risk assessments. Three levels are engaged in this: Technology, Circumstances, and gateways. The systems with such a gateway and credential patterns serves as a foundation for this architecture. Overall impact of the operation is also analysed using the proposed framework, and countermeasures for strengthening computer security are developed.

The author in [10] implies implementing a risk database management system. The use of security devices could be extended upon and using this online system. The architecture of the weaknesses information and indeed the technique for generating the problems information are presented in this research.

3. METHODOLOGY

The objective of the vulnerabilities control system is to identify and evaluate any application software problems. The morphological inspection and static analysis are used to conduct this assessment. Every weakness discovered will be reviewed, maintained in the VMS database, and tested if any software application is tested utilising the planned VMS technology. If indeed the identified exposure is a product, it can be registered in the VMS repository after already being evaluated either by various agencies. Comparable to the Appeared As early Points System, the Risk Monitoring System would not only discover the weakness but it will also assess or rank its intensity.

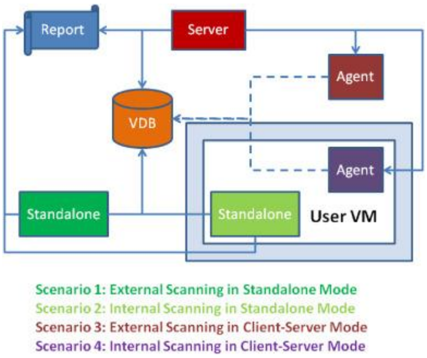
Report
Server
Agent
VDB
Agent
Standalone
Standalone
User VM
Scenario 1: External Scanning in Standalone Mode
Scenario 2: Internal Scanning in Standalone Mode
Scenario 3: External Scanning in Client-Server Mode
Scenario 4: Internal Scanning in Client-Server Mode

Fig 2. A Brief Overview of Scanning of Vulnerability Management System

A bug scanner, a process control platform, and a data storage comprise three main parts of the vulnerability management system. A bug scanner will have an unique interface it is only for and has a variety of functional characteristics. Any member of a product's security team can use this interface to advise the scanner here about how to continue with the next step. The computational system will analyse the computer user's code and binaries. Automatically selected procedures would be used to conduct this evaluation. Any weaknesses identified during the technology manufacturer's testing phase would be recorded in the database. Figure 2 depicts the use of the scanner. All previously unknown weaknesses would be recorded in the repository.

Data processing is categorized into 4 main phases, comprising finding vulnerabilities, evaluating vulnerabilities, resolving vulnerabilities, and disclosing vulnerabilities, throughout order to increase the effectiveness of scanning. Figure 3 depicts the four phases of the data processing.

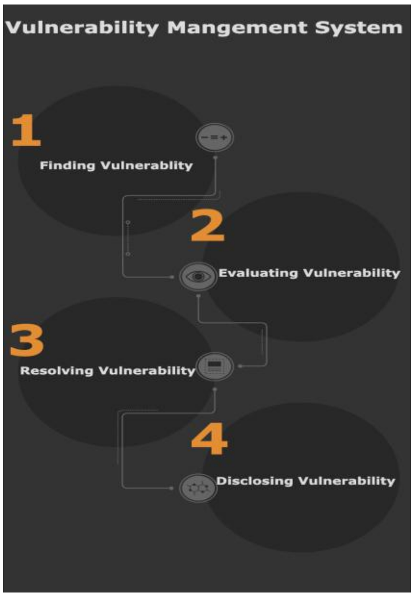

Fig 3. The four phases of the data processing.

A. Finding Vulnerabilities

The detection of risk is the most pivotal point in a Vulnerability Management system. The flaws in the evaluated software package would be disclosed as a result of this process. Any open ports and functions that really are existent in the software programme would be scanned and recognized during this process. That information would be utilized to just provide summaries, statistics, as well as other attributes.

B. Evaluating Vulnerabilities

The computer algorithm will have to evaluate all the risks once they've been identified and properly handled. Furthermore, overall intensity of the identified weakness will be evaluated in this phase, as well as the high transformation would be rated or assessed. The organization needs to determine how and where to highlight the identified issues that use these exposure ratings.

C. *Resolving Vulnerabilities*

It's essential to address risks after grading issues and prioritising many who were identified. The computer programmer of a tested application would be available seeing the weaknesses identified during this procedure.

After that, the programmer has three alternatives regarding addressing the issue: rectification, abatement, and adoption.

D. Disclosing Vulnerability

The speed and agility of detecting and preventing computer products will increase with using risk management systems.

Throughout this operation, a statement known as a software agency's scan test will indeed be created. A visual representation using several variables, including such risk scoring, etc., would be accessible with the VMS system. Every user also may raise a ticket at this point to accelerate the sharing of the comprehensive study or data

4.IMPROVING VMS SYSTEM

The complete details in the first vulnerability report, or as soon as possible, aid programmers in resolving the issue fast. This work aims at enhancing vulnerability scanning systems with the intent of creating vulnerability reports increasingly comprehensive. Researchers are particularly consisting of four ways to enhance vulnerability scanning systems.

A. Equipment Based

Vulnerability scanning processor architectures' characteristics are augmented with equipment based enhancements. Those that can help alleviate the hardship of knowledge storage and distribution. Vulnerability scanning processes, for instance, can be designed to automatically pinpoint the pertinent automatically send as well as add this to an user query. Besides that, having provided stages to produce offspring can indeed be done by machines through using acquisition techniques or meta; actual behaviour could be easily proved by standardising screen grab; and possible solutions can indeed be automatically generated. Every one of the case studies above are intended to assist inside the collection of information required by dev's to identify vulnerabilities.

B. Knowledge Based

The material becoming supplied by the reporters is the immediate priority of these improvements. Technologies like Cedilla , that provide real-time assessment on the value of the evidence provided and what can be added to maximize value, can be implemented into vulnerability systems. Reporters may be encouraged above and beyond and collect more data if indeed the system gives helpful recommendations. Technologies could be further modified to perform out assessment methods, including such verifying that supplied modifications are acceptable as well as the reported stack tracing is accurate and consistent

C. Technique Based

Technique-centric enhancements to vulnerability management systems concentrate on coordinating adware efforts. Selecting whichever worker will fix a bug, for example, could be mechanized to accelerate the process. Additional instances involve giving users an advance approximation of that when their issue would be resolved or create and encourage of something like the advancement achieved on event logs (so that reports are knowledgeable of the ways of responding to their efforts).

D. User Based

Reporters and programmers are both included in this. Reporters can understand what to do to get material or what to provide. Engineers also can benefit from similar requirements on what characteristics to expect in reports and use this information to resolve bugs.

5. ADVANTAGES

Some of the proposed benefits of the Vulnerability Management systems are envisioned as follows:

A. Increased Security

Software that has security weaknesses give hackers a way into the computer network. Finding these vulnerabilities is essential for protecting all assets and corporate data. Following that, these problems are rated for severity and given a priority. The evaluated report aids in fixing IT asset weaknesses and shielding them from cyber-attacks that might expose the network to security risks. Furthermore, IT security professionals are able to locate vulnerabilities remotely without being present physically in the computer environment. They can manage high-risk problems with the least amount of IT resources thanks to it.Figure 4 depicts the increase in security of data.

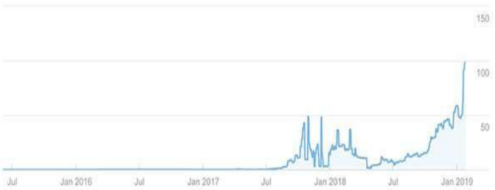

Fig 4. An example of increased security

B. Operational Efficiency

IT security professionals may remedy the most serious IT security concerns immediately and deal with the lesser ones afterwards with the review of the vulnerability list. The uncertainty during the clean-up process is eliminated if the effect of the potential dangers to the business is identified. Additionally, organisations may remotely automate and manage vulnerabilities. As a consequence, it saves time and improves operational efficiency by lessening the burden of owning and maintaining hardware and software upgrades. A thorough vulnerability management solution significantly reduces the work required by the security team. To reduce the likelihood of cyberattacks and improve the security posture, limited IT resources will be needed.

C. Far Less Expensive

No matter what kind of digital marketing strategy your business is involved in, overcrowding will always be an obstacle to success. Competing against a large marketing budget no longer has to play a factor in the ability to achieve success. It is a very good marketing platform related to marketing and anyone can do it! Paid ads quickly shut down small businesses that companies can afford.

D. Visibility and Reporting

The visibility of the security teams would suffer if the vulnerability reports are manually compiled across hundreds of assets. It is challenging to display the vulnerability data from one scan to another using conventional methodologies. Consequently, it is imperative to have a complete vulnerability management system with an operational dashboard. It provides a security flaw's severity rating, charts, solutions recommendations, and creates personalised reports, all of which assist in establishing a strong case for new security activities.

An IT team may then secure their computer environment with better-informed security decisions. Additionally, they may begin the remedy process immediately following each report. It helps teams operate more efficiently, lessens team fatigue, and takes the guesswork out of things.

6. CONCLUSIONS

The research introduced the Vulnerability Management System (VMS), which will identify and report a software product's vulnerabilities. We will research the current threat detection model and evaluate its effectiveness and speed of detection. Afterwards, we'll try to create a model that models outcomes greater rapidly, correctly, and efficiently. The improvised model also will determine the severity of an impact a weakness will have on the system, helping in prioritising vulnerabilities.

Existing vulnerability management systems do not adequately extract all of the necessary information by designers. Users believe vulnerability scanning technologies must be upgraded to efficiently obtain data because without it, developers can easily fix errors. The study postulates five main areas for improvement. It may be preferable to make a series of changes from some of these categories, problem tracking systems may also want to specialise, giving a wide variety of choices. In comparison to existing scenarios, where they all offer the same functionality, this would be a welcome improvement. Researchers also defined an online system of collecting data from reports and utilising that tool to determine the error's source as an example of the kind of enhancements we support. Researchers ran a preliminary analysis wherein the researchers mimicked a dynamic bug monitoring system in order to demonstrate the usefulness of that kind of proposal. In order to obtain relevant data about the defect immediately on and recommend prospective documents that need to be fixed, the system prompts the user situationally questions. This one will definitely speed the vulnerability process. Inside, we will develop from the dynamic system's current prototypes to a complete system which can manage a range of information gathering, as is commonly seen in the real world..

REFERENCES

[1] Madalina Aldea, Daniel Georgica, Victor Croitoru, “Software Vulnerabilities Integrated Management System”, 2020 13th International Conference on Communications (COMM), IEEE, 2020: pp. 97 - 102, doi: 10.1109/COMM48946.2020.9141970

[2] K. Guntupally, R. Devarakonda and K. Kehoe, "Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Centre Example," 2018 IEEE International Conference on Big Data (Big Data), 2018

[3] GeonLyang Kim, JinTae Oh, DongI Seo, JeongNyeo Kim, “The Design of Vulnerability Management System”, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 08 Issue: 11 | Nov 2021 www.irjet.net p-ISSN: 2395-0072 © 2021, IRJET | Impact Factor value: 7.529 | ISO 9001:2008 Certified Journal | Page 28 IJCSNS International Journal of Computer Science and Network Security, VOL.13 No.4, April 2013: pp. 19 – 24

[4] Manoj Kumar, Arun Sharma, “An integrated framework for software vulnerability detection, analysis and mitigation: an autonomic system”, Indian Academy of Sciences Sadhana Vol. 42, No. 9, September 2017, pp. 1481–1493, doi: 10.1007/s12046-017-0696-7

[5] Chee-Wooi Ten, Chen-Ching Liu, Govindarasu Manimaran, “Vulnerability Assessment of Cybersecurity for SCADA Systems”, IEEE Transactions on Power Systems, Vol. 23, no. 4, November 2008, pp. 1836-1846, doi: 10.1109/TPWRS.2008.2002298.ff

[6] Y. Jin, Z. Lin and H. Lin, "The Research of Search Engine Based on Semantic Web," 2020 International Symposium on Intelligent Information Technology Application Workshops, 2020, pp. 360-363, doi: 10.1109/IITA.Workshops.2020.193.

[7] Armold; Hyla, Rowe, “Automatically Building an Information-Security Vulnerability Database”, 2006 IEEE Information Assurance Workshop”, 21-23 June 2006, pp. 376-377, doi: 10.1109/IAW.2006.1652119

[8] Andrey Fedorchenko, Igor Kotenko, Andrey Chechulin, “Design of Integrated Vulnerabilities Database for Computer Networks Security Analysis”, 2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, 4-6 March 2015, pp. 559-566, doi: 10.1109/PDP.2015.38

[9] M. Rajaram and S. L. S. Vadivu, "Web caching in Semantic Web based multiple search engines," 2010 IEEE International Conference on Computational Intelligence and Computing Research, 2019, pp. 1-7, doi: 10.1109/ICCIC.2019.5705850.

[10] Ching-Huang Lin, Chih-Hao Chen, Chi-Sung Laih, “A Study and Implementation of Vulnerability Assessment and Misconfiguration Detection”, 2008 IEEE Asia-Pacific Services Computing Conference, 9-12 Dec. 2008, pp. 1252-1257, doi: 10.1109/APSCC.2008.212

[11] Jan-Min Chen, Chia-Lun Wu, “An automated vulnerability scanner for injection attack based on injection point”, 2010 International Computer Symposium (ICS 2010), 16-18 Dec. 2010, pp. 113 – 118, doi: 10.1109/COMPSYM.2010.5685537 Computational Intelligence and Computing Research, 2020, pp. 1-7, doi: 10.1109/ICCIC.2020.5705850.

[12] W. Xiaoyin, Z. Lu, X. Tao, J. Anvik and J. Sun, "An approach to detecting duplicate bug reports using natural language and execution information", *Proceedings of International Conference on Software Engineering*,2019

[13] R. Devarakonda and K. Kehoe, "Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example," 2018 IEEE International Conference on Big Data (Big Data), 2018

[14] Qing L, Boyu Z, Jinhua W, Qin Qian L. Research “on key technology of network security situation awareness of private cloud in enterprises”, IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA),2019

[15] Kumar R, Kumar P. “Special issue on recent trends in artificial intelligence techniques for fault-tolerance, reliability and availability in mission-critical networks. Recent Adv Comput Sci Commun”. 20205.