Vulnerability Management System

Pravin Kharat[1], Prof. Pramila M. Chawan[2]

[1]M. Tech Student, Dept of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India
[2]Associate Professor, Dept of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India

---***---

Abstract - *In simple terms, a vulnerability in cyber security refers to any fault or flaw or weakness in an information system, or internal controls or system processes of an organization. It can also be defined as a flaw or a fault in the source code design which determines the application malfunctions. Therefore, a good Vulnerability Management plan should be implemented to avoid attacks over the system or to minimize the damages produced by a cyberattack. This paper proposes a solution for vulnerability management by implementing a Vulnerability Management System (VMS) that will perform security tests to detect the vulnerability of the software product, the result of the vulnerability tested can be viewed by using a dedicated dashboard of the new platform created. This paper also defines the method of constructing and managing the vulnerability database. The data from the system that manages and calculates the relative severity of software vulnerabilities will be stored in the database.*

***Key Words*: Vulnerability management, vulnerability database, software vulnerabilities, Vulnerability management system.**

1. INTRODUCTION

Nowadays, every electronic device contains software that controls the function of the electronic device. These software components can be extremely complex, written by different developers, with a smaller part of code, which is later contained into a final software project or product. In most cases, the software application fails because the source code of the software product contains some untreated cases or flaws in the program which gives abnormal results. This error in the source code of the software application makes the application vulnerable. Therefore, the software application or Operating system which contains untreated cases, flaws, or weaknesses are known as software vulnerabilities. Later this software vulnerability exemplifies an entry point into a software system, which can lead to major damage to the application, the system where the application has been hosted and the system which is connected to the compromised system.

Despite all the security measures the number of vulnerabilities discovered continues to grow as the number of users using the internet has increased. Any device which contains software functions can tend to have source code errors, logical errors, flaws, the existence of detection techniques is mandatory for software vulnerability remediation as well as prevention.

A hacker always tries to penetrate a system and to gain certain privileges which can bring him access to valuable information or asset. A cyberattack can cause a huge loss of money, reputational damage, and valuable loss of data. If any security test is performed upon such a software system and vulnerability points are detected, a vulnerability plan is a must. This stage is one of the important roles, the way vulnerabilities are managed can drastically reduce the possible damages. A well-structured plan is mandatory in case of a vulnerable system.

1.1 Software Vulnerabilities

An error or a flaw or a weakness of the application's source code that an attacker or a hacker can take advantage of is known as software vulnerability. These errors tend to make the system function abnormally and undesirable actions. These flaws or errors in code may arise due to the lack of knowledge of the developer or programmer who is developing the software application. These flaws may lead to system crashes, loss of data, reputational damage, major damage to the targeted system, loss of customers, personal data being exposed, etc.

1.2 Types of vulnerabilities

The common security goals i.e., confidentiality, availability, integrity, non-repudiation, and usability, can be affected by the software vulnerabilities.

Following listed below are cyberattacks associated with software vulnerabilities:

Phishing: Phishing is a cyberattack that attempts to steal sensitive information. This sensitive information can be login credentials and credit card details. This attack can also be a form of social engineering where an attacker tries to mislead the user into clicking a malicious link created by the attacker, downloading some malicious attachments, or revealing sensitive data.

DDoS Attacks: Distributed denial of service attack is an attempt to spoil an online service or a website or a server or network by making it unavailable by sending a large number of access requests that it cannot manage.

Computer Viruses: Computer code or a program that modifies the way a computer behaves is known as Viruses. They are meant to spread through contaminated data, files, and insecure networks. And once it enters the system, it can replicate and spread from one program to another and infect other computer systems also.

Attack Vectors: Attack vector is a malicious term used to discover system vulnerability points, launch cyberattacks or install malicious software. Following are the four important attack vectors: Drive-by, Zero-day attack, MITM (man in the middle), SQL Injection.

Vulnerability Management System is not only intended to identify and evaluate vulnerability, but it will also generate a detail report which will report of the vulnerability point found in the software application which will be tested. VMS system will also maintain a proper database which will be known as VMS database. This database will store all the vulnerabilities which were found throughout the scanning.

2. LITERATURE SURVEY

In this section summarization of the existing research work is done. A new vulnerable management system will be created based on the existing work with additional functionality.

Mădălina Aldea.[1] The author in this paper has introduced a new vulnerability management system i.e., SV – IMS – Software Vulnerability Integrated Management System. This system can perform security tests to detect software vulnerabilities and the result of this test can be viewed upon a dedicated platform. It also gives defines the CVSS i.e., Common Vulnerability Scoring System, which is an international scoring system that describes how severe a vulnerability is.

Robert A. Martin.[2] The author in this research paper describes Common Vulnerability Exposure (CVE) and Open Vulnerability Assessment Language (OVAL) which are a pair of international, community-based efforts amongst industry, government, and academia. Where CVE is aimed to create a means for making vulnerability alerts more applicable to individual enterprises and OVAL is aimed to provide the means for standardized vulnerability assessment and result in uniform and standardized information assurance parameters for systems.

GeonLyang Kim.[3] The author of this research paper has introduced a new method for constructing and managing Vulnerabilities by creating a vulnerability database. In this research work, a new National Vulnerability Database (NDV) system is created which can be used by various enterprises. While referring a new vulnerability found can also be registered in the NVD system.

Manoj Kumar.[4] In this author proposed a framework that uses a knowledge base and inference engine. Using this the vulnerability management automatically takes required actions, classifies, prioritizes, and mitigates the vulnerability. The proposed system reduces the threats, security risks, and reputational and Monterey loss.

Chee-Wooi Ten.[5] This Author has proposed a Vulnerability assessment framework that evaluates the vulnerability of the SCADA system. This is done at three levels – System, Scenarios, and access points. This framework is based on the system which has firewall and password models. This proposed framework also evaluates the impact of the attack launched and countermeasures are identified for improvement of cyber security.

Jan-Min Chen.[6] In this paper, the author has implemented an automated vulnerability scanner that identifies the injection attack vulnerabilities. This system automatically examines the website to find the XSS and SQL injection vulnerabilities. The proposed system also uses NVD i.e., National Vulnerability Database.

Andrey Fedorchenko.[7] In this research paper, the author has proposed the process of integrating a vulnerability database system. This integrated database can be used for the further application of security systems. In this paper, the structure of the vulnerabilities database is suggested, and the process of vulnerabilities database generation is suggested.

3. PROPOSED SYSTEM

3.1 Problem Statement

To develop a Vulnerability Management System (VMS) which will detect vulnerability using source code and Binary code analysis of the software product and also analyze the intensity of the vulnerability found.

3.2 Problem Elaboration

The vulnerability Management system is aimed to identify and evaluate the vulnerabilities which are existing in the software application. This evaluation is performed using the source code and binary analysis. If any software application is tested using the proposed VMS system, the vulnerability discovered will be evaluated and will be stored and maintained in the VMS database. If the vulnerability found is totally new, it can be registered in the VMS database after it is classified by the competent entities. Not only the Vulnerability Management System will detect the vulnerability but also will analyze or will identify the intensity of the vulnerability found similar to the Common Vulnerability Scoring System (CVSS).

3.3 Proposed Methodology

The Vulnerability management system contains three main components: a vulnerability scanner, a data processing platform, and a vulnerability database. The vulnerability scanner will have its own dedicated dashboard with various attributes for functional purposes. This dashboard can be used by the product security team member to give input to the Scanner to process the further steps. The data processing platform will perform source code and binary code analysis of the software application or product. This analysis will be done using different defined steps. The vulnerability Database will store all the vulnerabilities found during the testing phase of the software product. New vulnerability found, will be registered in the database.

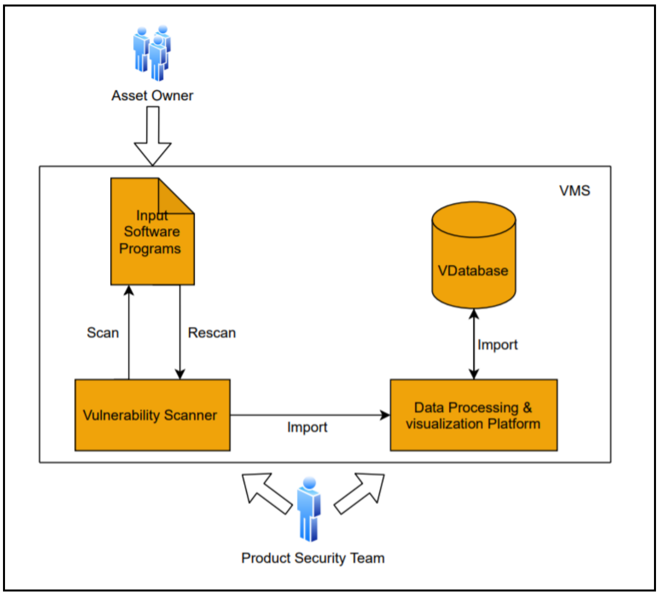

Fig -1: VMS Architecture

To improve the performance of the scanning, the data processing is divided in 4 major steps i.e., Identifying vulnerability, evaluating vulnerability, treating vulnerability, reporting vulnerability.

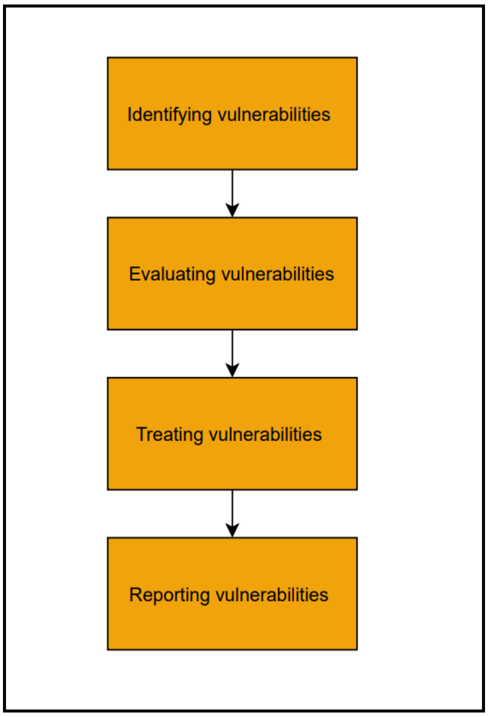
Identifying vulnerabilities
Evaluating vulnerabilities
Treating vulnerabilities
Reporting vulnerabilities

Fig -2: VMS Process Model

Identifying vulnerability:

The most important step in the process of a Vulnerability Management system is the identification of vulnerability. This process will bring to light all the vulnerabilities that exists across the tested software program. This process will scan and identify any open ports and services that are present in the software program. This insight will be used to create reports, metrics, other attributes.

Evaluating Vulnerabilities:

Once all the vulnerabilities are identified across the software program, it will need to evaluate them appropriately to deal with these findings. This step will also identify the severity of the vulnerability found and will rate or score the intensity factor. These vulnerability scores can help the organization to identify how to prioritize the discovered vulnerabilities.

Treating Vulnerabilities:

After scoring the vulnerabilities and prioritizing the vulnerabilities found, it is important to treat the vulnerabilities. In this process, the software developer of the tested program will be able to view the vulnerabilities found. After which, the developer can process the vulnerability treatment in three paths i.e., Remediation, Mitigation, and Acceptance.

Reporting vulnerabilities:

Vulnerability Management System will improve the speed and accuracy to detect vulnerability in the software product. In this process, the report will be generated based on the scan test run upon the software program. VMS system will be able to show visual representation using different parameters such as vulnerability scoring, etc. At this stage, the user can also raise the ticket to accelerate the process of sharing the detailed report or data.

4. CONCLUSIONS

In this paper, we have introduced a Vulnerability Management System (VMS), which will identify and report the vulnerabilities of the software product. We will study the existing vulnerability detection model, wherein we will check the accuracy and the speed of detection. After that, we will try to design a model which will give more accuracy, efficiency, and accelerated results. The improvised model will also analyze the level of the impact a vulnerability will impose on the system which will help in vulnerability prioritization.

REFERENCES

[1] Mădălina Aldea, Daniel Gheorghică, 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] Robert A. Martin, “Integrating Your Information Security Vulnerability Management Capabilities Through Industry Standards (CVE & OVAL)”, 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance, pp. 1528 – 1533), doi: 10.1109/ICSMC.2003.1244628

[3] GeonLyang Kim, JinTae Oh, DongI Seo, JeongNyeo Kim, “The Design of Vulnerability Management System”,

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

[6] Jan-Min Chen, Chia-Lun Wu, "An automated vulnerability scanner for injection attack based on injection point", 2010 International Computer Symposium (ICS2010), 16-18 Dec. 2010, pp. 113 – 118, doi: 10.1109/COMPSYM.2010.5685537

[7] 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

[8] 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

[9] 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