Vulnerability Management System

Pravin P. 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, 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 on the system or to minimize the damages produced by a cyberattack. To avoid such damages manual source code inspections or security audits are performed, which requires highly trained cyber security engineers, and it requires more time, which is prone to errors. For this reason, there is a need to automate such processes to discover vulnerabilities. This results in the implementation of the Vulnerability Management System, which will automate security testing for the identification of vulnerabilities caused in the software products.*

Key Words: Vulnerability, Vulnerability Management System, Algorithm, Software testing, Web based application.

1. INTRODUCTION

Many organizations have embraced the technologies such as software applications, web applications, software products, and many more to explore their new business opportunities and few organizations are being forced to adopt e-commerce due to advancements in software technologies, customers, or competitors. Software applications and web applications have been gaining popularity day by day, and these applications come up with different components which are highly complex and written by different software developers in different smaller chunks. Most of these applications fail to give proper output due to untreated cases or flaws. Therefore, the software application or Operating system which contains untreated cases, flaws, or weaknesses are known as software vulnerabilities. Later, the flaws in the source code of the application can be exemplified as an entry point for the hacker and can be treated as a software vulnerability.

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, and flaws. Thus, the existence of detection techniques is mandatory for software vulnerability remediation as well as prevention.

To avoid such situations manual testing, security audits, or code inspections are to be performed by highly skilled cyber security engineers or experts. But as it is labour intensive and expensive and prone to errors; automating the above steps to discover respective vulnerabilities for the software applications is required.

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 many 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.

2. LITERATURE REVIEW

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 Proposed Methodology

A hybrid algorithm is developed which automates the process of scanning software applications. The major goal of the proposed algorithm is to automate and increase the accuracy of vulnerability detection. Although the accuracy is not achieved at 100% but an effort is made to put up the proposed system above the existing systems. The proposed hybrid algorithm is similar to the existing algorithms. The OWASP results are considered with the output for better reasoning and understanding. OWASP results are updated on the regular basis to avoid any inconvenience.

The proposed system's hybrid algorithm is mainly based on the concept of combining different features which are of different components. This will result in the new algorithm which will give more impactful results on the respective scans. Therefore, the combination of such features from different components has been done based on optimization and sophistication among other components with the goal of increasing the accuracy or efficiency of the hybrid algorithm.

The Hybrid algorithm mainly consists of five phases i.e., Inspection, Scanning, Attack Detection, Analysis, and Reporting. The inspection which can also be called crawling, mainly focuses on fetching information about the application. The more information gathered in this phase the more successful the entire executed scan will be. After phase 1, phase 2 consists of scanning. Scanning is the process in which the algorithm will identify the weakness of the system on which the scan is been initiated. Once the scanning process is completed, the next step will be to identify the attacks or vulnerabilities and perform an analysis to identify the vulnerability definition and remediation methods. Later Reporting phase is initiated to generate a well informative report for the scan which was performed.

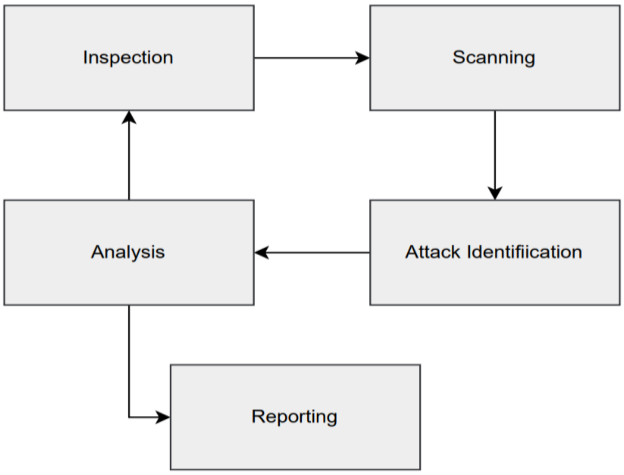


Fig -1: Component of VMS

A detailed description of the flow for the developed Hybrid algorithm:

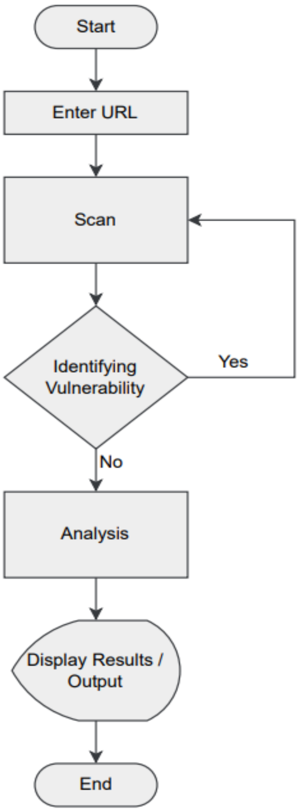


Fig -2: Flowchart for VMS

The initial stage according to the phase diagram is inspection which can be called as requirement or information gathering stage. After phase 1, the next process involves mainly crawling and parsing, and identifying new vulnerabilities. Phase 2 is repeated until all the vulnerabilities of the applications are not discovered. A further step includes analysis of the vulnerabilities found to identify proper definitions according to the OWASP and getting proper remediation for the same. Further, this analysis is summarized, and the final report is generated as an end result.

Input: Input is mainly provided by the user who is going to initiate the scanning. This input can be an IP address or the URL for the application which needs to be scanned by the VMS.

Processing: This step mainly involves fuzzing, crawling the pages, and identifying the weakness, and later vulnerabilities are identified based on the weaknesses identified.

Output: Output will be generated after the proper analysis process is done.

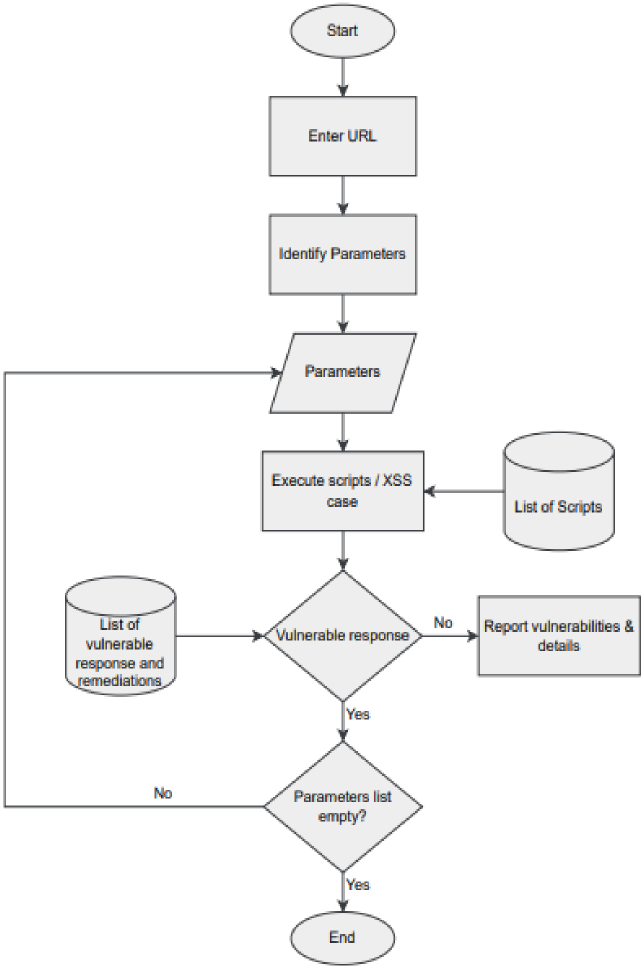


Fig -3: Flowchart for XSS

Many different scanning methods are used in VMS algorithm, considering one following method is used for Cross Site Scripting:

```
1. For each URL in the list of the visited URLs
    a) Identify all parameters
    b) Push the parameters in the list.
    c) For each of the parameter in the list
        i) give input as a XSS test case or script to the parameter and pass the request.
        ii) verify the respective response
2. Report the Vulnerability
```

3.3 Tools

These experiments or practical were performed by running different methods/tool with its respective scripts. These methods were installed and executed on Virtual machine and have the similar configurations and resources.

1. Nmap:

For probing computer networks, Nmap offers several functions, including host discovery, service detection, and operating system detection. Scripts that offer more sophisticated service discovery, vulnerability detection, and other features can extend these features. During a scan, Nmap can adjust to changing network conditions, such as latency and congestion.

2. Dirbuster:

DirBuster is an application with a GUI interface developed in Java. It is used to find concealed files and directories by brute-forcing files & directories with the aim of gaining some significant information that could help in cyber-attacks. A wordlist could influence how effective such a tool is; the more effective the wordlist, the more effective the instrument.

3. Xsser:

Cross-Site "Scripter" (also known as XSSer) is an automatic framework for finding, using and reporting XSS flaws in web-based applications. There are numerous ways to attempt to get around particular filters, as well as numerous unique code injection strategies.

4. Dnswalk:

A DNS (Domain Name System) debugger is called Dnswalk. Dnswalk carries out zone transfers for specified domains and executes precise database integrity checks in a variety of ways.

5. whois:

A query and response protocol i.e., WHOIS, which is pronounced "who is," is frequently used for accessing databases that list the registered users or assignees of Internet resources like domain names, blocks of IP addresses, and autonomous systems. On most UNIX systems, the command-line utility used to do WHOIS protocol searches is called whois. Additionally, Referral Whois is a sibling protocol of WHOIS (RWhois).

6. Nikto:

Nikto is a free command-line vulnerability scanner that looks for unsafe files/CGIs, out-of-date server software, and other issues on web servers. Checks are run on both generic and server-specific levels. Any cookies that are received are also recorded and printed. The data files used by Nikto to run the program are not free software, but the Nikto code itself is. Nikto can identify more than 6700 potentially harmful files and CGIs, as well as version-specific issues on more than 270 servers and obsolete versions on more than 1250 servers. Nikto can also identify installed web servers and software and checks for server configuration elements.

7. Dnsmap:

Dnsmap uses an internal or external wordlist to search a domain for common subdomains (if specified using -w option). There are about 1000 words in both English and Spanish on the internal wordlist, including ns1, firewall services, and smtp. Therefore, an automatic search for smtp.example.com within example.com will be available. For additional processing, results can be saved in CSV and human-readable formats. Dnsmap should not be executed with root privileges for security reasons because it does not need them to function.

9. Uniscan:

An open-source program called Uniscan can check web applications for serious flaws including cross-site scripting, blind SQL injection, remote file inclusion, web shell vulnerabilities, and hidden backdoors, among others. In addition to assessing vulnerabilities, Uniscan has the ability to search Google and Bing for domains using shared IP addresses.

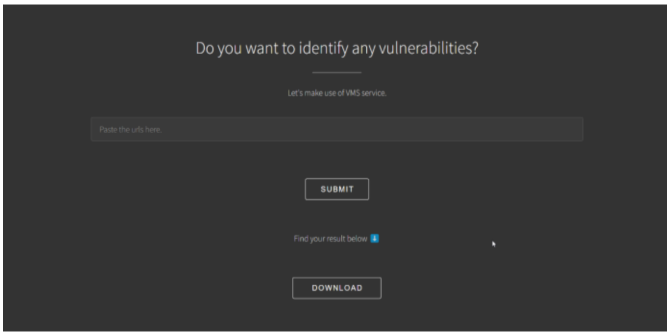
3.4 Resources required for the VMS tool

Operating System: Kali Linux or Ubuntu OS or System configured with Virtual Machine with same OS. The system and virtual machines specifications are as - processor, 2.6 GHZ Core i5, 2 GB RAM, 100 GB HDD and OS as above mentioned.

4. RESULTS AND DISCUSSION

Hi, I'm VMS : The Vulnerability Management System,
a Fine Secuirty Tool by VJTI Student.

Fig -4: Screenshot for web app of VMS tool



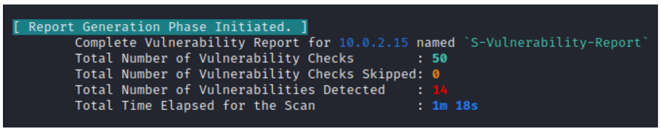
Do you want to identify any vulnerabilities?

Let's make use of VMS service.

Paste the urls here.

Find your result below ⬇

Fig -5: Screenshot of VMS URL input



```
[ Report Generation Phase Initiated. ]
        Complete Vulnerability Report for 10.0.2.15 named `S-Vulnerability-Report`
        Total Number of Vulnerability Checks         : 50
        Total Number of Vulnerability Checks Skipped: 0
        Total Number of Vulnerabilities Detected     : 14
        Total Time Elapsed for the Scan              : 1m 18s
```

Fig -6: Screenshot of VMS tool output

Vulnerability Management System generated better and faster results overall. It can automate the VA and PT process till particular instance. Also, it is able to identify the present vulnerabilities and specific remediation based on the vulnerability found and generate report with severity levels accordingly. The sole disadvantage is that, according to reports, it takes more time to scan than the majority of the web scanners utilized in this study. Although its performance is not perfect, compared to other tools, it has a larger capacity to detect more flaws.

5. CONCLUSION & RECOMMENDATIONS

5.1 Conclusion

The proposed Vulnerability Management System based on the hybrid algorithm extensively work to identify vulnerabilities based on software-based applications. Testing of such applications is done for safeguarding it. The suggested hybrid method presents additional vulnerabilities and does so in a professional manner when reporting those that have been found. However, because not all of the current vulnerabilities were completely scanned by the suggested hybrid approach. To make sure that "deep" crawling was carried out, the algorithm's crawling component needed to be increased. The results also indicate that the proposed method needs to be improved in order to complete the scanning quickly. To create an algorithm with the ability to identify more vulnerabilities, more study and research is required.

5.2 Recommendations

i) Improve crawling capabilities:

The proposed hybrid algorithm requires more methods and functions for crawling mechanisms so that VMS will be able to scan all the contents of any URL or a web application, without skipping any content of the webpage.

ii) Improve Analysis and Reporting:

VMS algorithms need to improve the accuracy so that the identified vulnerability can be stated with the severity level. To obtain high-end accuracy more sophisticated methods must be used during the scanning process which will require more research and practical implementation of the algorithm. VMS system can be upgraded for analysis and reporting in such a way that the vulnerabilities can be visualized and can be shown in a representable manner with the severity levels.

iii) Reducing scanning time:

There need to be proper results which is generated in short time frame with better accuracy and reporting. It needs to improve overall scanning method using more scripts and identification of more vulnerabilities.

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

[10] Yu, Y., Yang, Y., Gu, J., & Shen, L. (2011). Analysis and suggestions for the security of web applications. In Computer Science and Network Technology (ICCSNT), 2011 International Conference on, Vol. 1, pp. 236-240

[11] Pravin Kharat, Pramila Chawan, "Vulnerability Management System", 2021 International Research Journal of Engineering and Technology (IRJET), 25-28 Nov 2021

BIOGRAPHIES

Pravin P. Kharat
M Tech. Dept. of Computer Engineering – NIMS, VJTI,
Mumbai

Prof. Pramila M. Chawan, is working as an Associate Professor in the Computer Engineering Department of VJTI, Mumbai. She has done her B.E. (Computer Engineering) and M.E. (Computer Engineering) from VJTI College of Engineering, Mumbai University. She has 28 years of teaching experience and has guided 85+ M. Tech. projects and 130+ B. Tech. projects. She has published 143 papers in the International Journals, 20 papers in the National/International Conferences/ Symposiums. She has worked as an Organizing Committee member for 25 International Conferences and 5 ICTE/MHRD sponsored Workshops/STTPs/FDPs. She has participated in 16 National/International Conferences. Worked as Consulting Editor on – JEECER, JETR, JETMS, Technology Today, JAM&AER Engg. Today, The Tech. World Editor – Journals of ADR Reviewer -IJEF, Inters cience She has worked as NBA Coordinator of the Computer Engineering Department of VJTI for 5 years. She had written a proposal under TEQIP-I in June 2004 for 'Creating Central Computing Facility at VJTI'. Rs. Eight Crore were sanctioned by the World Bank under TEQIP-I on this proposal. Central Computing Facility was set up at VJTI through this fund which has played a key role in improving the teaching learning process at VJTI. warded by SIESRP with Innovative & Dedicated Educationalist Award Specialization: Computer Engineering & I.T. in 2020 AD Scientific Index Ranking (World Scientist and University Ranking 2022) – 2nd Rank- Best Scientist, VJTI Computer Science domain 1138th Rank- Best Scientist, Computer Science, India.