Security Auditing Methodology Using Crowdsourced Testing

Kalapala Naga Sasank   
Dept of Computer science and engineering (Honors)  
KLEFVijayawada, India  
nagasasank64@gmail.com

Atchyuth Kumar Panidepu  
Dept of Computer science and engineering (Honors)  
KLEFVijayawada, India  
atchyuththkumar225@gmail.comSunkara Revanth  
Dept of Computer science and engineering (Honors)  
KLEFVijayawada, India  
klucse2000030970@gmail.com

K.V.D Kiran  
Professor, Dept of Computer Science and Engineering (Honors)  
KLEF  
Vijayawada, India  
kiran\_cse@kluniversity.in

Jetreya Yadavalli  
Dept of Computer science and engineering (Honors)  
KLEFVijayawada, India  
2000030389@kluniversity.in

Venkata Vara Prasad Padyala  
Assoc. Professor, Dept of Computer Science and Engineering (Honors)  
KLEFVijayawada, India  
varaprasad\_cse@kluniversity.in

*Abstract*—The term cyber is derived from French, meaning "the art of governing". Cyber security as we know is the process of securing software and electronic networks from malicious attacks. Cyber security is a collective phrase and can be divided into different common categories and domains like information security. Most importantly cyber security has become essential for maintaining confidentiality and integrity. Vulnerabilities briefly defined are the backdoors, a weakness in the developed software or IT application that can be exploited to harm our software's integrity and disclose our confidentiality. Cybersecurity deals with these vulnerabilities as they seriously threaten critical information. As for how the security software resolves such issues is that they maintain a database, and a record of vulnerabilities and threats which were previously addressed and resolved, using them as a reference we scrutinize the software thoroughly to know if there are any known vulnerabilities or any other malware of similar origin.

Keywords—Vulnerabilities, Vulnerability Assessment, Cybersecurity, Information security.

# Introduction

Vulnerabilities are defined as a weakness, exploit, or defects that can be easily misused by attackers to steal information and collapse system integrity. Vulnerabilities can either be hardware defects or a defect caused while building the software either way it provides an effortless path for the attacker to exploit a network or confidential information. Vulnerabilities can be of many types, One of the most prevalent types of vulnerabilities that may be readily exploited by hackers is unchecked and weak passwords. It would be a serious error to download or install dubious files and software since this would expose your system to malicious attacks. Another common place where vulnerabilities are abundant is the internet where different types of spyware and malware are present which infect your system and cause various malfunctions. There is a specific procedure called vulnerability management that may be used to control and address such issues. Vulnerability management entails three key phases that are, vulnerability detection, assessment, and addressing the detected vulnerabilities. Vulnerability assessment can be defined as the process by which we evaluate the detected vulnerabilities and consider the possible resolutions to resolve the vulnerability. Vulnerability assessment in a sense can be very helpful in stopping serious threats such as SQL injections and code injections.

Information security is a part of cyber security that deals with securing confidential information and protecting it from external attacks which could alter, delete or destroy the software. Information security validates the privacy and integrity of the software's details as well as the protection of the software and the data it contains. There are several information security measures to be followed in order to protect our information and data. Such as having a data backup, is one of the most common and prominent ways of protecting our data in case of an accidental loss of data or data lost in an intentional attack. Firewalls are the first line of defense in securing our system and its information, Data encryption offers our information a special level of protection since only we would be able to access the data or information.

Cyber Security deals with the protection of digital systems, physical systems, and information stored within them from digital attacks and physical attacks. In general protection from physical attacks is mostly provided by using a firewall. When coming to the severity of the attacks both can be catastrophic but the main difference is that the physical attacks need human actions which means they cannot be successful without human intervention. Some of the physical attacks may include password capture by using physical devices such as. Digital attacks include Phishing Attacks, Malware Attacks, Man-in-the-Middle attacks, and Denial of Service. The most effective way to detect any malicious software or applications in any system is through the usage of the Intrusion Detection System(IDS) which is inbuilt into almost all of the present generation's antivirus software.

Despite physical attacks that require human presence at the attack region digital attacks do not require the physical presence of the attacker. When it comes to Digital attacks the attacker tries to know the vulnerabilities in the network that we are connected to and exploits them. He tries to compromise any one system that is connected to the network and then uses the compromised system to launch other attacks. Digital attacks enable the attacker to gain remote access to the victim's system which means the attacker can access and do any kind of modifications in the system without his physical presence. This enables the attacker to be able to monitor the victim's system remotely using his own system from a different location. These digital attacks are further classified into Active Attacks and Passive Attacks. Active attacks tend to cause damage to the system whereas passive attacks are only for monitoring purposes and do not tend to harm the victim system in any manner. Passive attacks are Easy to stop but are hard to detect whereas Active attacks are easy to detect but hard to stop.

Patches are security upgrades that fix vulnerabilities detected in a system or application of software already in use. These vulnerabilities provide a simple route for unauthorized access and data leakage, which result in malicious attacks. Maintaining the system's integrity and security necessitates the application of patches. Patches are an important part of system security and maintenance, there are different sorts of patches each adhering to its respective tasks in system security. Security patches are those that deal with vulnerabilities in the system, these are the updates that mainly focus on the problem causing the vulnerability including solutions to resolve it. Patches that fix bugs are used to repair or modify the functionality of the software which causes the system to malfunction. Patches are a crucial component of cyber security since they are responsible for updating virus databases and the system's security.

# Related Works

Most of the vulnerabilities are exploited in the web applications that we use daily. For Assessing such vulnerabilities, we can make use of different exploitation techniques for identifying them, these techniques exploit almost all the vulnerabilities in the application if that's not the case then we add and test the vulnerabilities found in the config files. These vulnerabilities can be exploited by making use of Automation tools like SQL Map, and OWASP framework. If there is a vulnerability found, then we have to examine the vulnerable code snippet and suggest secure coding practices that are to be implemented to patch those vulnerabilities. Websites should be continuously patched to get rid of vulnerabilities that may result in making the closed ports open in the network by which any malicious attacker can exploit the total network and the devices that are connected to it. Patching in some cases results in new vulnerability to be found in the application and hence every application that is being patched by the companies must make sure that the patch should result in the reduction of any vulnerability and must not result in the raise of any new kind of vulnerability.

# Existing System

According to Xiong Zhang, Haoran Xie, Hao Yang, Hongkai Shao, and Minghao Zhu, proposed a framework that characterizes the information system vulnerabilities using text input from crowdsourced testing platforms. This framework is intended to identify and classify vulnerabilities and, where appropriate, offer solutions that lower the risk of data loss and security problems. This framework is split into 6 steps which include recognition, analysis, classification, and providing solutions for the vulnerabilities. Those 6 stages might include data cleansing, elimination of missing values, and also replacement of letters appropriately.

First and foremost, the obtained raw data is preprocessed and made available for data cleansing, the replacement of uppercase characters, the removal of missing values, and data cleansing. The vulnerabilities are categorized and standardized in the second stage due to the fact that vulnerabilities, however, how similar they may seem, can sometimes have significant variances. Standardizing the categorized vulnerabilities is therefore advised. The third stage of the framework process involves breaking down the characteristics, of the data, into three levels. The word level describes the frequency of words in the dataset, followed by the phrase level, and finally, the record level contains the contents and statistics of the records. All the above data goes through a thorough data checking and cleansing process before proceeding to the fourth step in which the models are trained to test and verify the datasets of vulnerabilities and vulnerability classification. Risk-level prediction is done for the overall condition of the project later in module 5. Step 6 uses topic analysis to automatically provide the hints and recommendations needed to fix the persistent vulnerabilities.

To classify, process, and analyze the vast quantities of data sets needed for this procedure to draw a certain conclusion A variety of classification and regression approaches, including the K-Nearest Neighbor classification and regression technique, were utilized in the initial phase of the data collection process since KNN is an instance-based learning approach that is sensitive to the structure of data samples. Another classification approach utilized in this data filtering is SVM, also known as the support vector machine algorithm. SVM works by mapping and classifying the data to a higher feature in the data space so that the supplied data sets may be categorized more effectively.

As a result, after categorizing and filtering the data, we have a record of data sets that precisely depict the data that was affected in webpages by either SQL injection or other malicious cyberattacks. Next, vulnerability classification is started using the data that was filtered by the previously stated classification techniques. The table-formatted classification data serves as the foundation for the AUC-ROC curve, which graphically illustrates how well each classification method worked and what proportion of vulnerabilities were classified during the process. This displays the many categories of vulnerabilities in the process, such as SQL injection, file operation faults, and design or logic defects.

This improves our understanding of current vulnerabilities, and by using an artificial intelligence approach to identify and classify vulnerabilities, we can be more accurate in detecting those said vulnerabilities. Unlike earlier stages, where vulnerabilities were only identified, this process ensures that they are identified and classified according to their attack patterns or potential risk factors.



# Proposed System

The framework we're going to suggest is vulnerability resolution after finding and categorizing the vulnerabilities and classifying them in accordance with their potential risks gives us an idea of how to resolve such potential vulnerabilities. For this framework, we're developing a methodology based on crowdsource testing that depends on the data found to detect and assess the vulnerabilities to resolve them using existing patches in the various virus databases or quarantining them until a patch is found.

## **Crowdsourcing**

Crowdsource testing is a form of software testing method in which a significant number of people collaborate to evaluate a certain software application or software system. Typically, the group consists of volunteers or paid software testers who are evaluating the program for any vulnerabilities and spyware.

To make sure that the program is functioning effectively under diverse circumstances and settings, crowdsource testing is frequently utilized as a benefit for the software tests that have previously been completed. Most of the crowdsourced testing occurs during the beta testing phase when the product is practically complete but may still have bugs and vulnerabilities.

## **Categorizing of databases**

A cybersecurity training and research company that focuses on offensive security maintains the Exploit database, which is a repository of vulnerabilities. Cybersecurity experts utilize this database of numerous vulnerabilities and exploits, as a resource for creating optimized solutions for cybersecurity issues in software applications and systems. As new exploits and vulnerabilities are updated often, this database is quite helpful for cybersecurity and research purposes.

In order to analyze the vulnerabilities discovered and determine whether a patch is available, we are using the exploit DB database for this framework operation. This allows us to determine whether the vulnerabilities discovered are critical and must be quarantined, or if they already exist and can be patched.

## **Resource table**

#### We need datasets that have been carefully filtered and categorized in order for them to be appropriate for all of the operations we want to run on them in order to analyze vulnerabilities. It is feasible to create datasets by gathering information from numerous websites and software programs that have been attacked, filtering the information to meet our needs, and then organizing the information into the appropriate datasets.

As we input all our datasets from various sources, creating a resource table is beneficial for referencing and research. It can also be efficient to use resource tables as we have categorized all the raw data available in our datasets. The following table provides an example of how a resource table functions.

Resource Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Resource Name | Resource Version | Provider | Criticality level | Latest version | Latest patch |
| Apache web server | 2.04.54 | Apache | High | 2.04.54 |  |
| MySQL server | 8 | MySQL | High | 8 |  |
| Java MySQL Connector | 8.00.31 | MySQL | High | 8.00.31 |  |
| JBOSS server | 7.01.01 | RedHat | High | 7.01.01 |  |
| AWS EC2 | Ubuntu Linux | RedHat | Medium | Ubuntu Linux |  |
| Ubuntu | 22.04 | RedHat | High | 22.04 |  |

1. Table depicts the Typical resource table of an organization.

##### Security Auditing

The auditing method begins by examining the datasets in the resource table and then compares them to the existing exploits and vulnerabilities to determine whether or not the found flaws are already patched in existing versions. If the vulnerabilities have security patches to remedy the issues, prompt a security update to resolve the vulnerability. If the detected exploits are harmful and could harm your system or software application, quarantine the function or vulnerability that is causing the problem before continuing to use the product. However, no matter how cautious we are, attackers find several ways to exploit system vulnerabilities, therefore if the fault is too serious or catastrophic in nature, immediately close the software or system application to prevent additional data loss.

##### Pseudocode

1. Create a resource chart
2. Setup a vulnerability database
3. Check for vulnerabilities matching with resource table
4. If found
5. Determine if the resource is critical or not
6. If critical

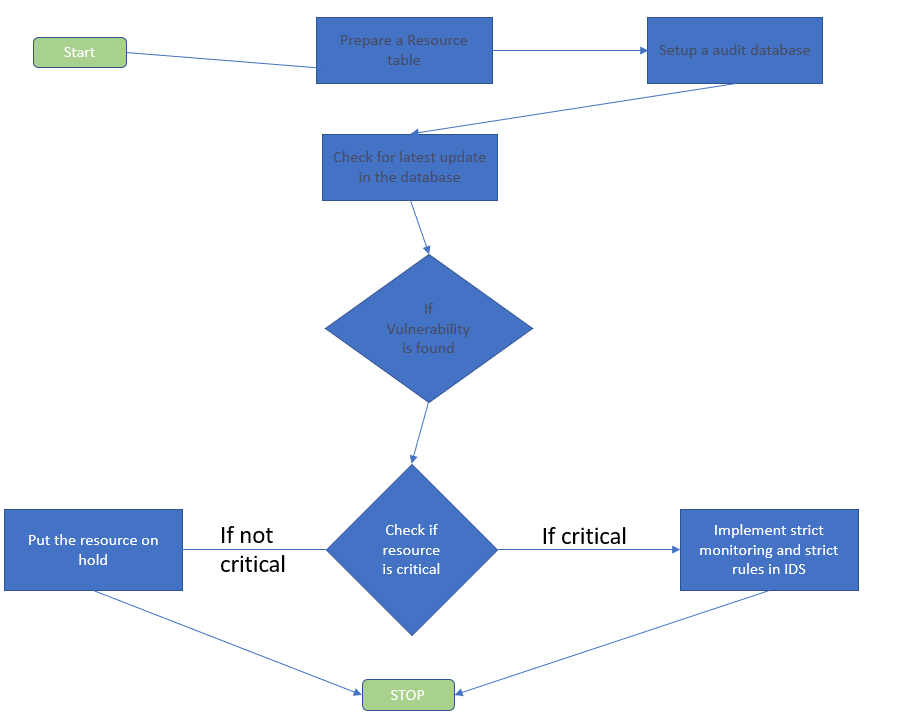
* Implement strict monitoring and strict IDS rules for the resource till patched.

1. If not critical

* Shut down the resource till patch

1. If not patched for a long time, then develop a patch (or) downgrade to a secure version of the same resource.

##### flow chart



##### Conclusion

Finally, the suggested system may fix the discovered vulnerabilities by comparing them to existing exploits and recommending existing version patches or quarantining the vulnerabilities if they are proven to be critical and could potentially leak system information. As a result, the research is complete in terms of identifying, categorizing, and fixing these vulnerabilities.

##### Acknowledgment

In this research, we proposed a methodology that organizations may use to conduct a continuous audit of their security. The mechanism for producing the dataset was proposed by Xiong Zhang, Haoran Xie, Hao Yang, Hongkai Shao, and Minghao Zhu,[1], making crowd-sourced testing viable.

##### References

[1] M. Rouse, “Vulnerability (Information Technology),” *TechTarget*, 2019. [Online]. Available: https://whatis.techtarget.com/definition/vulnerability. [Accessed: 25-Mar-2020].

[2] Positive Technologies, “Vulnerabilities in Online Banking Applications,” London, 2019.

[3] A. V. Revnivykh and A. M. Fedotov, “Main reasons of information systems vulnerability,” *Glob. J. Pure Appl. Math.*, vol. 12, no. 3, pp. 2133–2142, 2016.

[4] A. Hovsepyan, R. Scandariato, W. Joosen, and J. Walden, “Software Vulnerability Prediction Using Text Analysis Techniques,” in *Proceedings of the 4th International Workshop on Security Measurements and Metrics*, 2012, pp. 7–10.

[5] F. G. Bulut, H. Altunel, and A. Tosun, “Predicting Software Vulnerabilities Using Topic Modeling with Issues,” in *Proceedings of 4th International Conference on Computer Science and Engineering (UBMK2019)*, 2019, pp. 739–744.

[6] K. Tuma, G. Calikli, and R. Scandariato, “Threat Analysis of Software Systems: A Systematic Literature Review,” *J. Syst. Softw.*, vol. 144, pp. 275–294, 2018.

[7] T. H. Chen, S. W. Thomas, and A. E. Hassan, “A survey on the use of topic models when mining software repositories,” *Empir. Softw. Eng.*, vol. 21, no. 5, pp. 1843–1919, 2016.

[8] F. Farahmand, S. B. Navathe, P. H. Enslow, and G. P. Sharp, “Managing Vulnerabilities of Information Systems to Security Incidents,” in *Proceedings of the 5th International Conference on Electronic Commerce*, 2003, pp. 348–354.

[9] J. Radianti and J. J. Gonzalez, “Understanding Hidden Information

Security Threats: The Vulnerability Black Market,” in *Proceedings of the 40th Hawaii International Conference on System Sciences*, 2007, pp. 1–10.

[10] G. Li, J. Wu, J. Li, Z. Guan, and L. Guo, “Fog Computing-Enabled Secure Demand Response for Internet of Energy Against Collusion Attacks Using Consensus and ACE,” *IEEE Access*, vol. 6, pp. 11278–11288, 2018.

[11] Z. Guan *et al.*, “Cross-lingual multi-keyword rank search with semantic extension over encrypted data,” *Inf. Sci. (Ny).*, vol. 514, pp. 523–540, 2020.

[12] X. Zhang, A. Tsang, W. T. Yue, and M. Chau, “The classification of hackers by knowledge exchange behaviors,” *Inf. Syst. Front.*, vol. 17, no. 6, pp. 1239–1251, 2015.

[13] Y. Cong and J. Romero, “On information systems complexity and vulnerability,” *J. Inf. Syst.*, vol. 27, no. 2, pp. 51–64, 2013.

[14] A. K. Ghoujdi, *Exploit Prediction and Vulnerability Clustering Using Text Mining*. LAP LAMBERT Academic Publishing, 2013.

[15] Caiming Liu, Minhua Guo, Lingxi Peng, Jing Guo, Shu Yang, and Jinquan Zeng, “Artificial immunity-based model for information system security risk evaluation,” in *2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT2010)*, 2010, vol. 1, pp. 39–42.

[16] S. Samtani, R. Chinn, and H. Chen, “Exploring hacker assets in underground forums,” in *2015 IEEE International Conference on Intelligence and Security Informatics (ISI2015)*, 2015, pp. 31–36.