Journal of Ubiquitous Systems & Pervasive Networks
Volume 14, No. 1 (2021) pp. 19-26

CLASSIFICATION OF LOGICAL VULNERABILITY BASED ON GROUP ATTACK METHOD

Faisal Nabi*[a], Jianming Yong[b] , Xiaohui Tao[c]

[a,,b] School of Management and Enterprise, University of Southern Queensland1West St, Darling Heights QLD 4350, Australia
[c] School of computer Sciences, University of Southern Queensland, QLD 4350, Australia

Abstract

New advancements in the field of e-commerce software technology have brought many benefits; at the same time, however, developing processes always leads to a number of different problems, from the design phase to the implementation phase. Software faults and defects increase the problems with reliability and security and for these reasons; a solution for these issues is needed. This paper addresses the problems associated with a lack of clear component-based web application related classification of logical vulnerabilities. The primary method of addressing the issues is through identifying Group Attacking Method by categorizing two different types of vulnerabilities in component-based web applications.

Keywords: Security Dimensions, CBS Application, Group Attack Method, Application logic, Attack Classification

1. Introduction

The growing complexity of modern e-commerce software based on component architecture is creating many benefits for the e-commerce industry. However, at the same time, critical processes of different available commercial off the shelf components may cause software application logic faults. These defects may occur during the plug and play phase of an application's new functionality development that increases the issues of reliability and security [3]. Therefore, an approach is required to classify the issues on the base of a component-based software faults and flaws categorization scheme, which can then classify each attack into a group attack ID through the attack method. The characterization of the attack method is based on vulnerability that may be caused by fault logic in an application design. The design faults or flaws are system design phase issues that cannot be mitigated through modification of a few lines of component code or interface connection code [10].The security breaches caused by such problems are discussed through the security dimension, which reflects the system aspects and attributes. This may be affected by risk of loss in the event of cyber-attack through group attacking method. The security dimensions are divided into categories of problems where the attacking method may cause logical vulnerability to enter into a system. The division into categories may help the developers understand the design issues of security related system attributes. The security dimension is based on further attributes of the security system, such as security group knowledge, attack group knowledge, vulnerability category and attack boundary, and group attack method in the system. All of these attributes perform major roles in identifying and classifying the logical vulnerabilities based on a Group Attack Method. A Group Attack Method explains the type of vulnerability and its attacking parameters that trigger an infected component in the case of particular event within the system. This process exploits the system security dimension. Therefore, such a scheme is needed to be developed that could characterize the two different vulnerabilities: logical and technical, into groups and classifications.

* Corresponding author. Tel.: 61+0405265929
Fax: +00000; E-mail: Faisal.nabi@yahoo.com

DOI: 10.5383/JUSPN.14.01.004

2. Objectives

The research focuses on the progress towards highlighting different security dimensions of categorized vulnerability into classifications of each attack with parameters trigger an exploitable event within the system. Highlighting the security dimensions will help to understand the further attributes of these dimensions related to a system.

2.1. Method

Our research methodology focuses on a classification that separates or orders main objects (or specimens) into classes. Classifications can be generated by a priori (i.e. non-empirically from an abstract model) or a posteriori (empirically) by looking at the CVE vulnerability database for security breach cases [11].

3. Related work

According to Samaila 2017, as defined in figure 1, classified the vulnerability into three units by the intersection of each of these three units. The first unit is a system's weakness that causes a flaw, the second unit is the attacker's approach of accessing the flaw, and the third unit is being able to exploit the flaw by an attacker [1] but did not propose any classification based on these three units or categorized them into attack cause.

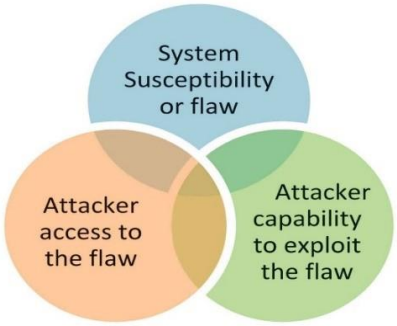

Figure 1 Vulnerability Model [1]

Krsula, 1998, defined the classification of software vulnerability related issues in figure 2, as being based on a fault, specifically those regarding faults specification, development / configuration in terms of software. For example, execution can violate clearly defined security policy [2]. This can be mitigated through the elimination of this problem in a numerous ways, such as software patches and re-configuring the devices [5]. Krsula's classification is more possibly about an environmental fault that is described below in figure 2, The Taxonomy of Software Vulnerability Causes. However, the shortcoming of his research is the limitation of the proposed scheme to the software fault and the related environmental condition.

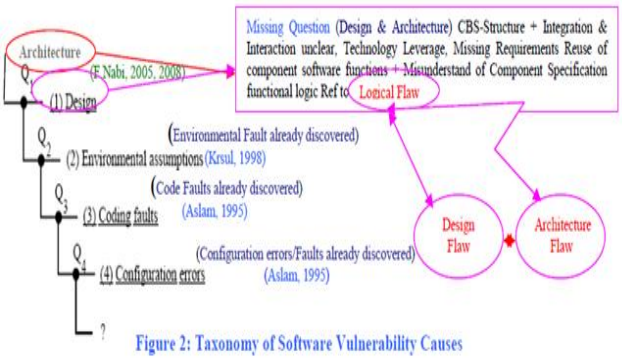

Figure 2: Taxonomy of Software Vulnerability Causes

Joshi and Singh (2014) proposed the classification five dimensional vector of vulnerability and defined the defense, method and its impact related to target attack [3]. However, their work most likely covers the network vulnerabilities and shortfalls about design flaws in the architecture of software-based applications particularly in the case of component-based development.

Software vulnerability occurs due to the existence of software bugs, faults and errors, which may cause an unchecked buffer or race condition attack [4].

To the present time, there have been many different classification schemes [12, 13, 14, 15, 16] proposing the targeting of various parameters related to the technology affected software production life cycle (SDLC) phase, the revelation process and the attack pattern [6].

Modern classification of vulnerability models mostly focuses on the vent of software vulnerability, which is a single cause, and the target domain specific application. For this reason, a single vulnerability may not be caused by a single reason [8]. A single vulnerability can occur for many different reasons in a system [9]. Therefore a single cause can be linked to different vulnerabilities in different sort of applications based on a class of domain. Therefore, it is can be argued that such presentation does not categorize the classification models in a holistic way. Moreover, the present schemes does not provide any detail about logical vulnerability based attack classification and group attack method. This paper covers the research gap between present classifications as stated in related work and the approach adopted in this paper "Classification of logical vulnerability "and group attack method".

4. Proposed Vulnerability Classification Model

The security dimensions are considered as aspects of the system and the attributes or related processes that leave their effects on a security group to know system and deliver the changes to the system as explained in figure 3. The security dimensions are based on having an

understanding of the class of vulnerability and its category. The security dimensions directly impact on security group knowledge to evaluate the issues related to the security in each network or system. The knowledge can be both logical and technical, and each aspect of both can be categorized and a classification is given before mitigating the security issues.

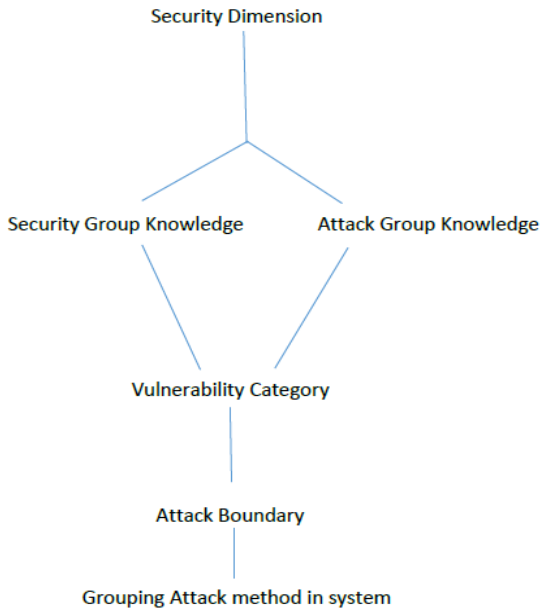

Figure 3: The proposed vulnerability classification model

The attack group knowledge also refers to an attack pattern that depends on rigorous methods of exploitation by the attacker. This dimension of security is based on a process or set of system attributes that may be exploited in an action by an attacker with the means of gaining access to the system related information.

The fourth element of security dimension is the vulnerability category. In this stage having been evaluated by the first two processes: security group and attack group knowledge gained, a vulnerability is classified and categorized into its group based on exploitation techniques and parameters. Once a vulnerability is categorized, its attack boundary profile is designed, keeping in view the level of impact on the system in case of exploitation of the security function. The attack boundary profile helps to understand the level and scale of infection and the impact on the system that became the target of attack propagation. An attack boundary is defined through a set of systems under attack that is controlled as a single administrative control. At this level, boundaries are various and vulnerabilities can become obvious, as the data object inputs the boundary race condition.

The group attacking method consist of attack ID, classification and attack group that simplifies the vulnerability and attacking technique (as defined in table 1), whereas group classifies the attack dimension fall under this category. The purpose of this model is to simplify the attack dimensions and way of attack fall under the category, where each vulnerability is subdivided into attack class and method, as defined in the model. Presentation of the model is depicted through the table Grouping Attack Method ID & Logical Vulnerability Classification.

The given below grouping attack method ID & logical vulnerability classification.

Table 1: Group attacking method ID and Logical Vulnerability Classification.

SN	Attack Classification	Attack method	Attack Group	Category
1	Application Logic attack	Logic Design Fault	Exploitation of Functionality	Web Application
2	Application Logic attack	logic diversion error	Anti-Automation	Web application
3	Application logic attack	control flow error	web function exploit	Web application
4	Application Logic attack	programme logic flaw	Subversion of Logic	Web application
5	Application Logic attack	functional flow Fault	exploit the sequences of logic order	Web application
6	Application Logic attack	Design logic flaw	web Copy Cat	Web application

The logical attacks are different types of attacks with different attack methods because logical attack has to exploit the functionality that is specific to the application and its logic. These are defined in the above-mentioned table of Grouping Attack Method ID & Logical Vulnerability Classification.

As mentioned above, the main scope of this study is “application logic based vulnerabilities”, a problem that exists because of a design flaw or fault that mismatch between design and architecture while developing component–based software application. We have classified the six vulnerabilities in the application logic and then developed the attack group and vulnerability classification to be categorized by the proposed model of classification and security dimension in the light of the vulnerability model that is the cause of design flaws in application logic and functionality.

4.1 Classification of Logical Vulnerability VS Technical Vulnerability

In the light of our research, the proposed model would turn into be a classification & characterization of two distinctive categories of vulnerability issues /problems "Technical vs Logical Vulnerabilities" as defined in figure 4. These vulnerabilities are classified based on the attack method as mentioned in the above table of vulnerability.

Therefore, keeping in view the proposed model of classification falls under the two category of vulnerabilities , which have been drawn into classification tree model dividing into sub-class of attack at the application layer of ecommerce component-based software application. This depicts the detailed classification, having characterized the each vulnerability by their unique signature of indemnity in the proposed scheme.

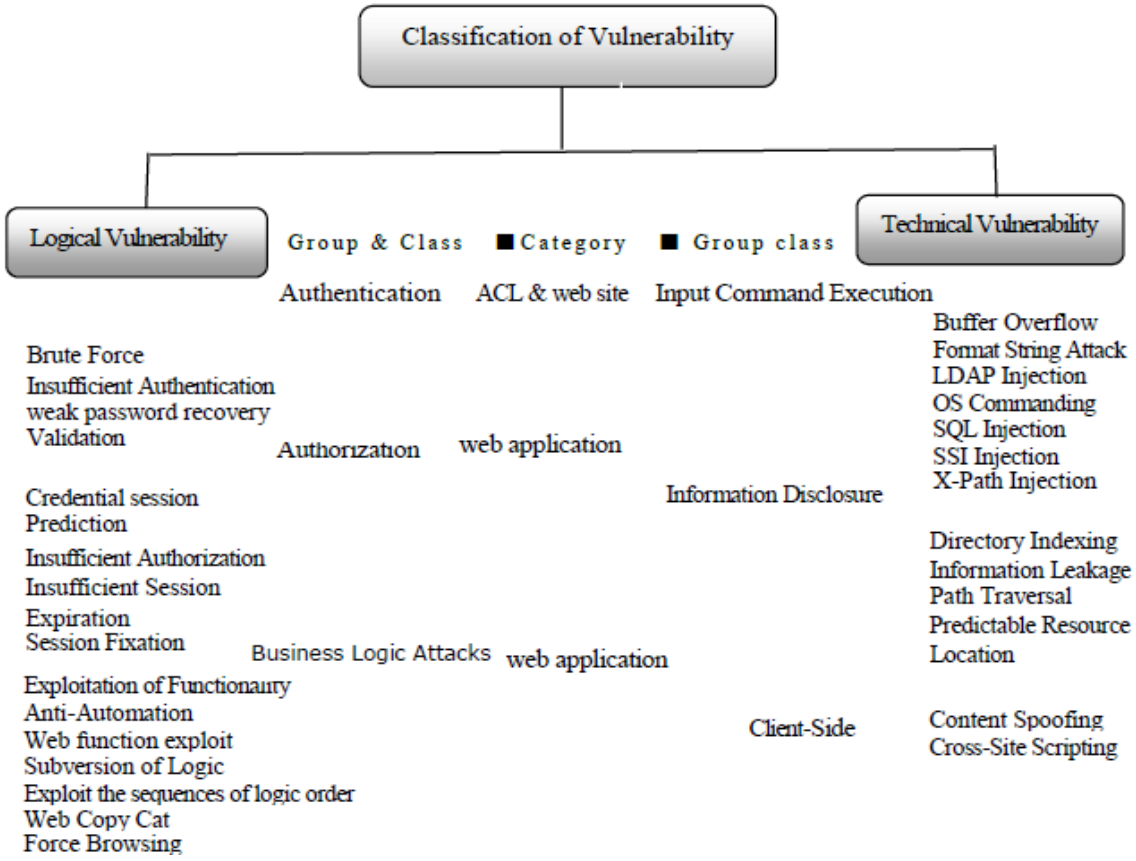

Figure 4: Classification of Vulnerability Scheme

The proposed contribution of the classification is characterized by attack pattern and target agent in each kind of attack as mentioned in the given classification scheme of application logic based attack pattern method, vulnerability class and event triggering logical element. The attack pattern technique can also be used to classify each vulnerability in the light of attack method; such classifications are characterized in groups of attacking parameters, which define the nature of vulnerability. This classification relates to the attack pattern technique. Our strategy is based on a novel paradigm of attacker / defender designed model depicted in figure 5 to represent interconnected systems that hold and work with different types of information. Consumption and provision of resources are expressed by parents and children, which require arbitrary, interdependent modelling and system Infrastructure.

4.2 Layer based Software system scenario attack modeling

Figure 5 depicts the software layer based system attack scenario to validate the above-mentioned proposed model. This figure clearly explains the role of software and service into different layers and relationships between actors of organizations and that face threats. This model help us to understand the three-dimensional layer model of software system, service, information and event; the attacker affects those and attacks must be mitigated through defender actions. This model classifies the vulnerability lifecycle in the layer based software system attack model. The method and tool for such modelling is UML and the aspect oriented modelling languages that support the event attack modelling through the attack surface, have been demonstrated in figure 5.

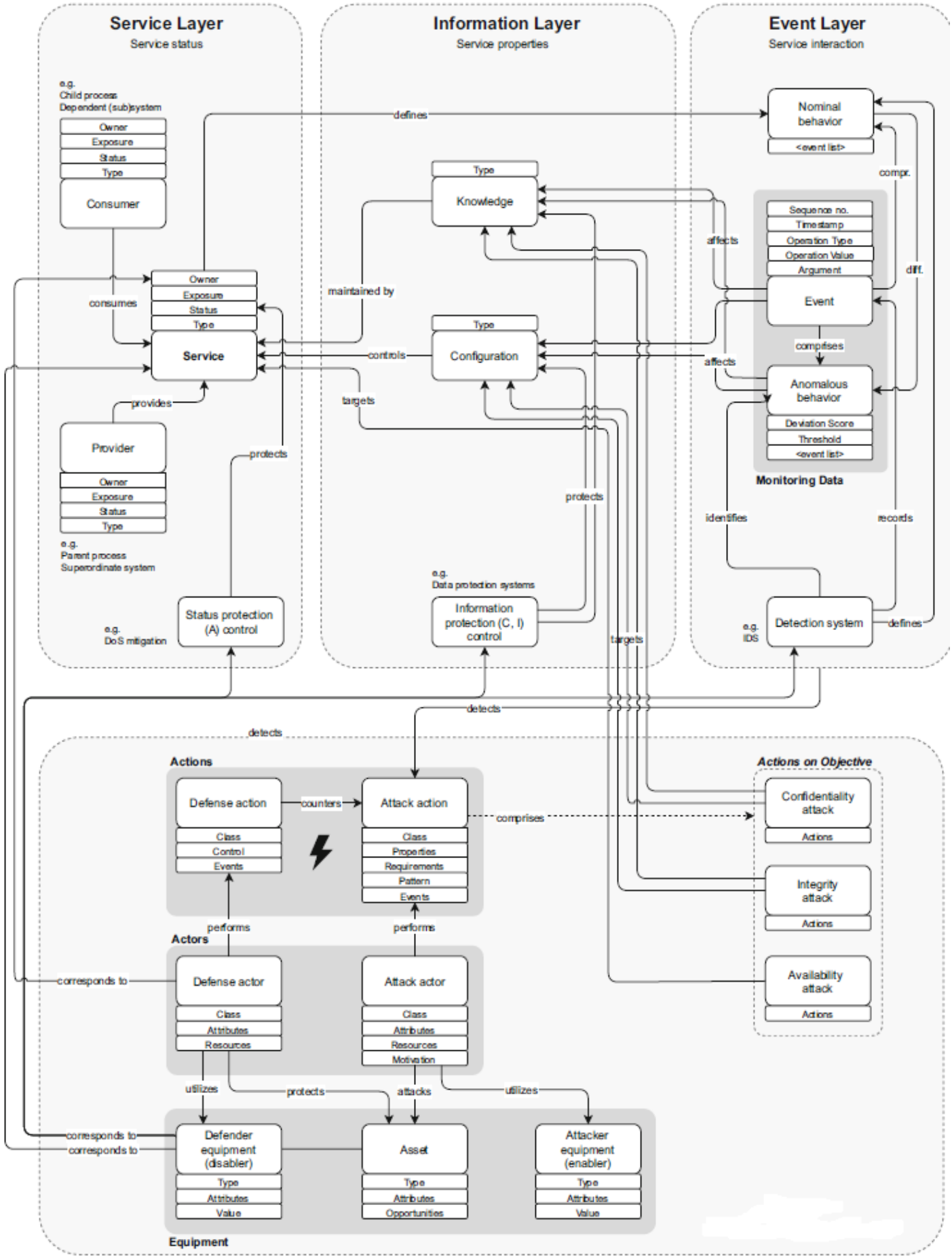

Figure 5: layer based software system attack model

4.3 Classifying and Categorizing Logical Vulnerabilities

The group attacking parameters based nature of logical vulnerability and attack technique classification are defined according to each type of attack and characterized according to attack method.

1. Attack pattern Technique & Group
Exploitation of Functionality
Class logic design fault
Category: Web server service (Target Agent)
Attack Method (encoding circumvents access controls)
Logic error (attack of Cause)
Implementation level (fault logic classification)
Vulnerability Type: Exploitation of Functionality

This vulnerability class identifies the category of this attack pattern as business logic or application logic, where the attack falls under the logic design fault in the web server side target agent and the method of avoiding it is encoding circumvents access controls.

2. Attack Pattern Technique & Group
Insufficient Anti-automation
Class logic diversion error
Category: Web software (target Agent)
Attack Method (Anti-Automation)
Divert logic (attack of Cause)
Implementation level (Process logic flaw classification)
Vulnerability Type: Insufficient Anti-automation

This class of attack falls under the classification of insufficient anti-automation attack pattern technique. The category of this vulnerability falls under the web application that is identified as application logic and the method is process logic flaw classification.

3. Attack Pattern Technique & Group
Insufficient Process Validation
Class control flow error
Category: Web application (target Agent)
Attack Method (web function exploit)
Divert application flow control (Attack Cause)
Implementation level (application logic fault classification)
Vulnerability Type: Insufficient Process Validation

This vulnerability falls under the web application category where the attack method is web function exploited with the technique of application logic fault classification and insufficient process validation technique. This comes under the business application of logic vulnerability.

4. Attack Pattern Technique & Group
Subversion of Logic
Class programme logic flaw
Category: Server application (target Agent)
Attack Method (exploit the work flow)
Subvert application logic (attack Cause)
Implementation level (application Design logic Flaw classification)
Vulnerability Type: Subversion of logic

This vulnerability class programming logic fault falls under the category of server side application target agent, where subversion of application logic diverts the control flow of the entire application logic, the method of attack is to exploit the workflow.

5. Attack Pattern Technique
Forced Browsing
Class functional flow Fault
Category: Web logic (target Agent)
Attack Method (exploit the sequences of logic order)
Divert service flow (attack cause)
Implementation level (application function error classification)
Vulnerability Type: Force-Browsing

This class of vulnerability falls under the functional flow fault classification of attack, web logic is the target agent, and where the entire function of web logic diverts service. The method of this attack is to exploit the sequences of logic order.

6. Attack Pattern Technique
Web Copycat
Class Design logic flaw
Category: Web software application (target Agent)
Attack Method (exploit the business logic)
Application logic flow diversion (attack Cause)
Implementation level (Design Flaw classification)
Vulnerability Type: web Copy Cat

This class of vulnerability is classified as web copycat attack target agent is design logic flaw at the web software application that exploit the business logic through application logic flow diversion as an attack cause.

5. Discussion

Therefore, we have detailed the classification and characterization of vulnerabilities into groups and the methods of attacking them. From this research, it may be understood that that logical vulnerabilities cannot be mitigated through traditional approaches such as web scanning tools, and vulnerabilities detection tools that are based on static analysis. Web scanners only detect the implementation bugs, programming error conditions, and faults whereas logical vulnerabilities are based on the design-phased flaw of software based applications [17].Therefore, our proposed scheme is based on

classification and categorization of each logical vulnerability based on the attack method, which is explained through the parameters of attack logic in each case presented above. .

The classification with defined detailed information about each attack and the related attack pattern will be helpful for the developers, having knowledge of the different attacks with technique to design new applications based on the idea of security by design technique.

6. Conclusion

The notion of a security development process is based on a proper classification of the vulnerability. It is useful to have knowledge about the attack and its parameters, target agent, and method. With the passage of time new technologies emerges, and more security attacks occur on the software application server side, so in this scenario the researcher has made an effort to classify the logical vulnerabilities that are never given consideration by the research community. The proposed vulnerability classification model contributed to the new classification and is related to the group attacking method ID and vulnerability classification, which has never been undertaken before. The proposed model will be useful for developers to understand the two different sorts of vulnerabilities, especially logical vulnerabilities, while designing applications or security by design based ideas intended for adoption. This model will cover the gap of logical vulnerabilities and related attack patterns, techniques, and methods. This model provides a significant improvement to taxonomies of a class of vulnerability that has not previously been given much consideration by the research community.

References

[1] Samaila, M. G., Neto, M., Fernandes, D. A. B., Freire, M. M., & Inácio, P. R. M. (2017). Security Challenges of the Internet of Things. Pp. 53–82, In J. Batalla, G. Mastorakis, C. Mavromoustakis, & E. Pallis (Eds.), Beyond the Internet of Things. Internet of Things (Technology, Communications and Computing) Cham: Springer. doi:10.1007/978-3-319- 50758-3_3 https://doi.org/10.1007/978-3-319-50758-3_3

[2] Krsul, I. V. (1998). Software vulnerability analysis (Doctoral dissertation). Retrieved from https://dl.acm.org/citation. cfm?id=927682

[3] Joshi, C., & Singh, U. K. (2014). Admit-A five dimensional approach towards standardization of network and computer attack taxonomies. International Journal of Computer Applications, 100, 5. doi:10.5120/17524-8091. https://doi.org/10.5120/17524-8091

[4] Li, X., Chang, X., Board, J. A., & Trivedi, K. S. (2017). A novel approach for software vulnerability classification. In Reliability and Maintainability Symposium (RAMS), 2017 Annual (1– 7). IEEE. doi: 10.1109/RAM.2017.7889792 https://doi.org/10.1109/RAM.2017.7889792

[5] Antoon, R. U. F. I. (2006). Network Security 1 and 2 Companion Guide. (Cisco Networking Academy).

[6] Fournaris, A. P., PoceroFraile, L., & Koufopavlou, O. (2017). Exploiting hardware vulnerabilities to attack embedded system devices: A survey of potent microarchitectural attacks. Electronics, 6(3), 52. doi:10.3390/electronics6030052 . https://doi.org/10.3390/electronics6030052

[7] Garg, S., & Baliyan, N. (2019a). A novel parallel classifier scheme for vulnerability detection in android. Computers and Electrical Engineering, (Final revision submitted) doi:10.1016/j. compeleceng.2019.04.019. https://doi.org/10.1016/j.compeleceng.2019.04.019

[8] Homaei, H., & Shahriari, H. R. (2017). Seven years of software vulnerabilities: The ebb and flow. IEEE Security & Privacy, (1), 58–65. doi:10.1109/MSP.2017.15 https://doi.org/10.1109/MSP.2017.15

[9] Sharma, C., & Jain, S. C. (2014, August). Analysis and classification of SQL injection vulnerabilities and attacks on web 18 S. GARG ET AL. applications. International Conference on Advances in Engineering and Technology Research (ICAETR), 2014 (1–6). IEEE. doi: 10.1109/ICAETR.2014.7012815 https://doi.org/10.1109/ICAETR.2014.7012815

[10] Faisal Nabi, A Process of Security Assurance Properties. Unification for Application Logic, International Journal of Electronics and Information Engineering, Vol.6, No.1, PP.40-48, Mar. 2017.

[11] Jens L. Eftang a, Martin A. Grepl b, Anthony T. Patera, A posteriori error bounds for the empirical interpolation method, C. R. Acad. Sci. Paris, Ser. I 348 (2010) 575–579 http://www.sciencedirect.com/ , https://doi.org/10.1016/j.crma.2010.03.004 .

[12] Hansman, S., Hunt R., "A taxonomy of network and computer attacks". Computer and Security, vol. 24, issue 1, Feb 2005, PP. 31-43. https://doi.org/10.1016/j.cose.2004.06.011

[13] Simmons, C., Ellis, C., Shiva, S., Dasgupta, D., & Wu, Q. "AVOIDIT: A Cyber Attack Taxonomy", University of Memphis, Technical Report CS-09-003, 2009. [Online]. Available:

[14] T. Aslam, "Use of a taxonomy of Security Faults," Technical Report 96-05, COAST Laboratory, Department of Computer Science, Purdue University, March 1996.

[15] Scott D., Angelos S," Towards a Cyber Conflict Taxonomy", 5th International Conference on Cyber Conflict K. Podins, J. Stinissen, M. Maybaum (Eds.), 2013.

[16] Lough, Daniel. "A Taxonomy of Computer Attacks with Applications to Wireless Networks," PhD

thesis, Virginia Polytechnic Institute and State University, 2001.

[17] Marco Vieira, Nuno Antunes, and Henrique Madeira, Using Web Security Scanners to Detect Vulnerabilities in Web Services, 2009 IEEE/IFIP International Conference on Dependable Systems & Networks, https://ieeexplore.ieee.org/abstract/document/5270294.,, https://doi.org/10.1109/DSN.2009.5270294