www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 5 May 2016, Page No. 16471-16477

Study of vulnerability taxonomies to provide a Novel approach for vulnerability categorization

Bindu Dodiya Ranawat[1], Dr.Umesh Kumar Singh[2]

[1] Institute Of Computer Science Vikram University , Dewas Road Ujjain
Madhya Pradesh,India
bindu.dodiya@yahoo.co.in

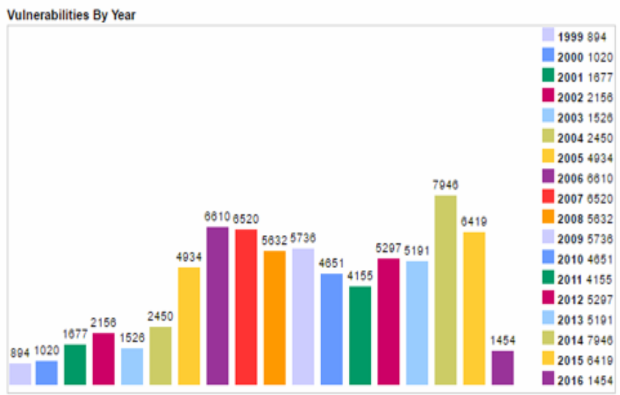
[2] Institute Of Computer Science Vikram University , Dewas Road Ujjain
Madhya Pradesh,India

ABSTRACT ***:In this age of universal electronic connectivity when world is becoming a global village ,different threats like viruses and hackers, eavesdropping and fraud, undeniably there is no time at which security does not matter. In view of large growing population of vulnerabilities, major challenge is how to prevent exploitation of these vulnerabilities by attackers. The first step in understanding vulnerabilities is to classify them into a taxonomy based on their characteristics. A good taxonomy also provides a common language for the study of the field. Properties and requirements of good taxonomy are described in this paper to lead security experts for the development of secure infrastructure. An analysis of some prominent taxonomies and their valuable aspects are highlighted that can be used to create a complete useful taxonomy.***

Keywords: CVE,CVSS,Taxonomy,Vulnerability

1. INTRODUCTION

Computer vulnerabilities are omnipresent .In recent years there have been numerous reported exploits targeting software applications [1] Because of these exploits software security has gained prominence and priority. Software applications are exploited by using vulnerabilities present in them. Vulnerability is defined as a state of the system from which it is possible to transition to an incorrect system state [2]. In other words, vulnerability is a defect which, when exercised, can produce undesired and incorrect behaviour [3].The number of vulnerabilities has increased vastly in last decade. Total number of 64545 new vulnerabilities has listed in CVE[4]from year 2005 to march 2016.Fig presents number of vulnerabilities listed by CVE from year 1999 to march 2016. The first step in understanding vulnerability is to classify them into a taxonomy based on their characteristics .A taxonomy classifies the large number of vulnerabilities into a few well defined and easily under stable categories. Such classification can serve as a guiding framework for performing a systematic security assessment of a system. In fact one of the goals of producing taxonomy of vulnerabilities is to develop automated tools for performing security assessment. In this paper we provide a pervasive survey of important work done for developing taxonomies of attacks and vulnerabilities in computer systems.This survey covers work done in security related taxonomies from 1976 to 2014. We summarize the important properties, goals, classification criteria, limitations of the taxonomies to provide a framework for organizing information about known vulnerabilities into a taxonomy that would benefit the security assessment process.



2. Motivation:

Most existing classification schemes, as is evident, begin with a theoretical and comprehensive approach to classifying security defects. Most research to date has been focusing on making the scheme deterministic and precise, striving for a one-to-one mapping between a vulnerability and the category the vulnerability belongs to[5] .Taxonomies developed for a particular system are rarely useful for different systems. This is one of the reasons there are so many taxonomies in the literature. Each of them addresses a specific kind of system .For example, a taxonomy of vulnerabilities in operating systems is of little use when conducting a security assessment of a cryptographic protocol., An analysis of some prominent taxonomies has been done in this paper and valuable aspects are highlighted that are needed to create a complete useful taxonomy.

3. Standard Properties of Taxonomy

Before examining existing taxonomies and developing new ideas and methods, it is important to define what a good taxonomy consists of. A number of requirements have been compiled from various sources in Lough (2001) [6] and are listed below:

Accepted: The taxonomy should be structured so that it can become generally approved.
Comprehensible: A comprehensible taxonomy will be able to be understood by those who are in the security field, as well as those who only have an interest in it.
Completeness)/Exhaustive: For taxonomy to be complete/exhaustive, it should account for all possible attacks and provide categories accordingly. While it is hard to prove a taxonomy that is complete or exhaustive, it can be justified through the successful categorisation of actual attacks.
Determinism: The procedure of classifying must be clearly defined.
Mutually exclusive: A mutually exclusive taxonomy will categorise each attack into, at most, one category.
Repeatable : Classifications should be repeatable.
Terminology complying with established security terminology Existing terminology should be used in the taxonomy so as to avoid confusion and to build on previous knowledge.
Terms well defined: There should be no confusion as to what a term means.
Unambiguous: Each category of the taxonomy must be clearly defined so that there is no ambiguity with respect to an attack's classification.
Useful: A useful taxonomy will be able to be used in the security industry and particularly by incident response teams.
It is not necessary for any taxonomy to satisfy all of the properties identified above because depending on the field to which they belong, they have different goals. But it is desirable that a good taxonomy must adhere all of the above properties.

4. Survey Of existing Taxonomies :

The RISOS study[7] , focused on flaws in operating systems. The RISOS (Research In Secure Operating Systems) study defines seven classes of security flaws: Incomplete parameter validation ,Inconsistent parameter validation , Implicit sharing of privileged /confidential data, Asynchronous validation/Inadequate serialization, Inadequate identification /authentication /authorization ,Violable prohibition/limit ,Exploitable logic error .Here, all vulnerabilities have a 1-tuple.

The objective of the Protection Analysis (PA) project [8] was to enable anybody to discover security errors in the system by using a pattern-directed approach. The idea was to use formalized patterns to search corresponding errors.

Landwehr et al. [9] focused on nature of flaws and classified security flaws according to three criteria: genesis (how did flaw entered in system), time of introduction (when in development cycle flaw entered) and location (where in the system flaw exists). Motive was to consider possible sources of flaws from different perspectives. Within each of these categories, sub categorization provided. Defects by genesis were broken down into intentional and inadvertent, where the intentional class was further broken down into malicious and no malicious. Defects by time of introduction were broken down into development, maintenance and operation, where the development class was further broken down into design, source code and object code. Defects by location were broken down into software and hardware, where the software class was further broken down into operating system, support, and application.
Aslam developed taxonomy to organize information being stored in a vulnerability database by using causes of flaws as criteria for classification [10]. He focused on UNIX operating system flaws only and presented three main categories: Operational fault, Environmental fault, Coding fault. Operational and coding fault categories are further subcategorized. Same fault can be classified in more than one category. Viewpoint is very narrow as flaws can be generated due to many other reasons also.
Krusal [11] adopted assumptions made by programmers as classification criteria. Krusal extends Aslam's work [14] and developed a detailed taxonomy. Main categories proposed in this taxonomy were: Design, Environmental assumptions, Coding faults, Configuration errors. Ambiguity in distinguishing between objects and attributes because of interpretation scope permitted by taxonomy. It also fails to elaborate on how assumptions lead to vulnerabilities.
Howard [12] presents a taxonomy of computer and network attacks. The approach taken is broad and process-based, taking into account factors such as attacker motivation and objectives. The taxonomy consists of five stages :attackers, tools, access, results and objectives. The attackers consist of a range of types of people who may launch an attack. These range from hackers to terrorists. Tools are the means that the attackers use to gain access. Access is gained through either an implementation, design or configuration vulnerability. Once access is gained, the results may be achieved such as corruption or disclosure of information. From this process the attacker achieves their objectives which may vary from inflicting damage, to gaining status. Howard attempts to focus attention on a process driven taxonomy, rather than a classification scheme. This means the whole attack process is considered, which is certainly valuable.
Howard's approach is useful in gaining insight into the process of attacks. However, for information bodies such as CERT, such a taxonomy may not be of much practical value. Information bodies are more concerned with the attack itself, than with the motivations and objectives behind it.

Bishop presents taxonomy of UNIX vulnerabilities [13] by classifying vulnerabilities along six axes (categories): Nature of vulnerability, Time of introduction, Exploitation domain, Effect domain, Minimum number of components necessary to exploit the vulnerability, Source of the identification of vulnerability Bishop's approach is different as it uses axes instead of flat or tree like taxonomy. Proposed axes unable to divide software domain according to software functionality. Time of introduction can be non mutual exclusive for some vulnerabilities.

Du and Mathur [14] proposed a three dimensional taxonomy with the goal to develop a practical and usable categorization of software errors. As proposed single error can be assigned to multiple categories to cover all the features of an error, in contrast to mutual exclusiveness desired in any standard categorization scheme. Three proposed dimensions based on operational viewpoint are: By cause (Seven subclasses), By direct impact (Four subclasses), By fix (Four subclasses) First dimension by cause is similar to Landwehr's genesis category excluding intentional part. This taxonomy is flexible and can be adopted in other systems for cause and impact relationship analysis as done in [15]

In 2001, Lough[6] proposed a taxonomy called VERDICT (Validation Exposure Randomness Deallocation Improper Conditions Taxonomy) and is based upon the characteristics of attacks. Instead of a tree-like taxonomy, Lough proposed using four characteristics of attacks:
Improper validation: insufficient or incorrect validation results in unauthorised access to information or a system, Improper exposure: a system or information is improperly exposed to attack. Improper randomness: insufficient randomness results in exposure to attack. Improper deallocation: information is not properly deleted after use and thus can be vulnerable to attack

Piessens [16] proposed taxonomy of causes of software vulnerabilities with aim to help developers to focus on most frequently occurring causes of vulnerabilities. In this two level hierarchical taxonomy, top level is based on phases of SDLC: Analysis phase, Design phase, Implementation phase, Deployment phase and Maintenance phase. These phases are again subcategorized in two to six subcategories. Purpose of this taxonomy is practically very right as research reports indicate that many vulnerabilities are due to small numbers of causes. But it is difficult to assign vulnerabilities to SDLC phases because depending on level of abstraction classification can change. Number of phases is also a point discrepancy.
Gray [17] proposed a taxonomical framework comprising of ten classes by combining and extending work of Landwehr, Bishop and Wang .Proposed classes for program flaws are: Genesis, Time of introduction, Location, Execution environment, Quality impact, Method of discovery, Threat and exploitation scenarios, Monitoring and exploitation scenarios, Limitation and remediation scenarios, Elimination methods Purpose of this taxonomy is to classify vulnerability information to suit needs of different people at different positions with different point of view and diverse priorities. It is a flat taxonomy that limits practical adoptability for analysis purposes.
Jiwnani [18] proposed three dimensional vulnerability taxonomy with the aim to classify vulnerabilities to identify parts of system that have higher concentration of vulnerabilities. Taxonomy also aimed to identifY most common type of vulnerabilities so that testing and maintenance team can prioritize their efforts in more critical areas. Overall purpose was to develop more secure system in future by increasing testing efforts in vulnerability prone areas of system. This work focused on operating system vulnerabilities only. Jiwnani adopted two dimensions from Landwehr's [9] classification and introduced a third dimension. The three dimensions proposed were: Software development issues (Eight subclasses), Location of flaws in the system (Six subclasses), Impact of flaws on the system (Nine subclasses). Three dimensions further classified in various categories almost similar to Landwehr's scheme. Taxonomy was analysed by applying 1360 operating system vulnerabilities, results indicate that majority of vulnerabilities are associated with few areas and small number of software engineering issues. It signifies that by applying efforts in right direction systems can be secured in more efficient manner.
Pothamsetty & Akyol [19] categorize network protocol related vulnerabilities in classes and also offer engineering design, development and testing best practice countermeasures for each of these classes. For these they developed test technique taxonomy and best practices taxonomy besides vulnerability taxonomy. Classes in vulnerability taxonomy are: Clear Text Communication, Non-Robust Protocol Message Parsing, Insecure Protocol State Handling, Inability to Handle Abnormal Packet Rates, Vulnerability Arising From Replay and Reuse, Protocol Field Authentication, Entropy Problems. Taxonomy need to be manually updated to keep with newly discovered vulnerabilities and changing best practices. Generalization capabilities are cumbersome in view of ever increasing population of vulnerabilities.
Tsipenyuk et al proposed Fortify taxonomy [5] that organized coding errors in form of taxonomy to organize sets of security rules that help software developers in understanding causes and impact of security errors. This scheme gives an alternative to previously proposed schemes that focus only on operating system vulnerabilities. Eight classes proposed are: Input validation and representation, API abuse, Security features, Time and state, Errors, Code quality, Encapsulation, Environment. Classification claimed to be two level hierarchical but subclasses are not well defined.
Weber [20] proposed software flaw taxonomy which is very similar to Landwehr's classification by genesis. Purpose of this work was to help in development of code analysis tools to detect software security flaws. Taxonomy has two main classes intentional and inadvertent. Further intentional class has two subclasses malicious and non-malicious and inadvertent has five subclasses validation error, abstraction error, asynchronous flaws, subcomponent misuse/failure and functionality error. These subclasses are further categorized. Classification inherited same limitations from Landwehr's but author argued that taxonomy should be useful for its intended purpose instead of satisfying all standard properties. This taxonomy has the issues of ambiguity and mutual exclusiveness .
In [21] Seacord and Householder pointed out that most of the proposed vulnerability taxonomies do not address problem domain properly. They suggested that classification scheme should be based on engineering analysis of problem domain instead of published vulnerability reports. Their approach is to use attribute-value pairs to characterize

vulnerabilities. Their approach is inclined towards ontology development rather than taxonomy.

Hansman and Hunt [22]proposed a taxonomy that consist of four dimensions which provide a holistic taxonomy in order to deal with inherent problems in the computer and network field .The first dimension allows for classification of attack target .the second dimension classifies attack target, In third dimension vulnerabilities are classified and payloads are classified in fourth dimension. This taxonomy is a good start towards a taxonomy for computer and network attacks however is unable to classify blended attacks. Attacks that have targets that require other targets are not fully modelled in the taxonomy.

Kjaerland [23] proposed a taxonomy of cyber-intrusions from Computer Emergency Response Team (CERT) related to computer crime profiling, highlighting cyber-criminals and victims. In this research, attacks were analyzed using facet theory and multidimensional scaling (MDS) with Method of Operation, Target, Source, and Impact. Each facet contains a

number of elements with an exhaustive description. Kjaerland uses these facets to compare commercial versus government incidents. Kjaerland's taxonomy focuses on the motive of the attacker in an attempt to quantify why the attack takes place, and where the attack originated. Her taxonomy contains some limitations as she provides a high level view to the methods of operation without providing more details to the methods that can be used in identifying attack inception.

In [24] Bazaz & Arthur proposed taxonomy of vulnerabilities based on relationship between computer system resources, process and vulnerabilities. As vulnerabilities exploited due to violation of constraints and assumptions associated with resources, proposed classification express vulnerabilities in form of constraints and assumptions. Taxonomy has three levels in hierarchy; top level has three categories which represents resources: main memory, Input/output and Cryptographic resources. These top level categories divided in six subcategories which are also resources in form of components of higher level. These components are then subcategorized in different constraints and assumptions. Proposed approach is novel and promising in context of proposed framework but has limited scope to specific perspective and highly dependent on point of view.

IGURE et al[25] proposed a four level classification scheme. First level of classification is attack impact. Second level of classification is based on system-specific attack .Third level of classification comprises of system components (attack targets) Fourth level of classification was based on system features (source of vulnerability).

Chris Simmons et.al:[26] proposed a cyber attack taxonomy called AVOIDIT (Attack Vector, Operational Impact, Defence, Information Impact, and Target)to aid in identifying and defending against cyber attacks they used five major classifiers to characterize the nature of an attack, which are classification by attack vector, classification by attack target, classification by operational impact, classification by informational impact, and classification by defence. their fifth category, classification by defence, is used to provide the network administrator with information of how to mitigate or remediate an attack. AVOIDIT provides, through application, a knowledge repository used by a defender to classify vulnerabilities that an attacker can use AVOIDIT intends to provide a defender with vulnerability details to what encompasses an attack and any impact the attack may have on a targeted system .AVOIDIT is able to classify blended attacks by providing the ability to label various vulnerabilities of an attack in a tree-like structure. The defence strategies in the taxonomy presented a defender with an appropriate starting point to mitigate and/or remediate an attack. The plausible defences are enormous, so this taxonomy provides a high level approach to cyber defence.

References

Scott D.et.all[27] proposed A cyber conflict taxonomy it is an extensible network taxonomy organized as a plex data structure. Subjects of the taxonomy are entered as either Events or Entities and are then categorized using the categories and subcategories of Actions or Actors. Each of these categories is further subdivided into increasingly specific subcategories used to describe the defining characteristics of each subject and labelled lateral linkages are used to illustrate the associative relationships between Entities and Events.. this taxonomy can potentially identify actors across different events based on their similar method of operation, toolsets and target sets.

5. Summary of taxonomies studied

S.no	Taxonomy	Goal	Classification criteria	Limitation
1.	RISOS project 1976[7]	To categories operating system flaws	Operations of OS	A single flaw might have different classification
2.	PA, 1978[8]	Enabling discovery of security errors in system by using pattern directed approach	Formalized patterns to search for corresponding errors	The procedure for reducing defects to abstract patterns was not comprehensive.
3.	Landwehr, 1994 [9]	To consider possible sources of flaws from different perspective .Focused on nature of flaws	Generis, time of introduction ,location	Categorization by genesis is ambiguous, inability to classify some existing vulnerabilities.
4.	Aslam, 1995[10]	To organize vulnerability data being stored in a database	Faults at implementation level	Lacks the high level categories to classify design errors.
5.	Krsul, 1998[11]	Characterize operating system flaws	Assumptions made by programmer	Ambiguity in distinguishing between objects and attributes. fails to how assumptions lead to value
6.	Howard[12]	In gaining inside into the process of attacks	Attackers,tools,access,results,objectives	A taxonomy may not be of much practical value for information bodies such as CERT.
7.	Bishop, 1999[13]	Describe vulnerabilities in a form useful for IDS	Nature ,time of exploitation, effect, minimum number of components, source of identification.	Time of introduction can be non mutual exclusive for some vulnerability.
8	Du and Mathur 2000[14]	To develop a practical and usable categorization of software errors .	Three dimension based on operational viewpoint : By cause, By direct imact, By fix.	Classification scheme does not satisfy Mutual exclusiveness.
9	VERDICT, 2001[6]	Provide classification according to characteristics of attack	By characteristics of attack	Classification scheme does not satisfy Mutual exclusiveness, specifically categorization for attack vulnerabilities
10	Piessens, 2002[16]	To help developers to focus on most frequently occurring causes of vulnerabilities	Phase of SDLC	Difficult to assign vulnerabilities to SDLC ,because depending on level of abstraction classification can change.
11	Andy Gray,2003[17]	To classify vulnerability information to suit needs of different people at different position with different point of view and diverse priorities.	combination of existing taxonomies	Doesn’t offer any subclasses for any of the class,is a flat taxonomy limits practical adoptability for analysis purpose.
12	Jiwnani 2004[18]	To identify parts of system that have higher concentration of vulnerabilities	Software development issues, location of flaws in the system, impact of flaws in the system	Focused only on operation system vulnerabilities

13	Pothemsetty and Akyol, 2004[19]	To categorize network protocol related vulnerabilities	Cause of flaw	Generalization capabilities are cumbersome in view of ever increasing population of vulnerabilities.
14	Tsipenyuk, 2005[5]	To organize sets of security rules that help software developers in understanding cause and impact of security errors .	Errors in source code	Classification claimed to be two level hierarchical but subclasses are not well defined.
15	Weber, 2005[20]	To help in development of code analysis tools to detect software security flaws	Classify security flaw based on two main classes intentional and inadvertent	Issue of ambiguity and mutual exclusiveness.
16	Seacord, 2005[21]	To provide vulnerability classification based on engineering analysis.	Based on attribute value pair	A vulnerability may belong to multiple attributes
17	Hansman, 2005[22]	To provide holistic approach to classify attacks	Four dimension : attack vector, attack target, vulnerabilities and exploits, effect or payload of attack	Unable to classify blended attacks ,attacks that have vulnerabilities that require other targets are not fully modelled in taxonomy.
18	Kjaerland, 2006[23]	Focus on the motive of the attacker in an attempt to quantify why the attack takes place and where the attack originated	Method of operation,target, source and impact	Provide high level view to method of operation without providing more details to the methods that can be used in identifying attack inception.
19	Bazaz and Arthur, 2007[24]	To develop a framework for deriving verification and validation strategies to assess software security.	Computer system resources	Only provide classification of vulnerabilities that are in the form of violable constraints and assumptions .
20	Igure 2008[25]	To provide view of relationship between computer system resources, process and vulnerabilities	Attack vulnerability	Focused on classification only for known vulnerabilities.
21	AVOIDIT, 2009[26]	To characterize the nature or attack	Attack vector ,operational impact ,defence, information impact, target	Lack of defence strategies, Physical attack ommission
22	Cyber conflict, 2013[27]	To provide an organized formal model that can be used to measure the impact of attacks and different defence strategies both in specific scenarios and in large scale cyber conflicts.	Using the categories and sub categories of actions and actors	Taxonomy does not allow for any formal or empirical relationship among the entities beyond parent child relationship.

1.

2. 6. Conclusion:

3. The previous taxonomy attempts can definitely be counted as milestones along the timeline of complex task of vulnerability classification in view of multifaceted characteristics of vulnerabilities .There is need of standard vulnerability taxonomy for security assessment .In this paper study of previous efforts are reviewed that can be useful in the manner that it provides a direction to security experts while developing a taxonomy. It can be useful to identify which properties should be considered for developing a standard taxonomy. However there are many taxonomies developed to date but some prominent of them were analysed.

4.

5. References

6. [1] Bugtraq Mailing List, Retrieved march 2016 from http://www.securityfocus.com/archive
7. [2] Matt Bishop and David Bailey, "A Critical Analysis of Vulnerability Taxonomies" Tech. Rep. CSE-96-11, Department of Computer Science, University of California at Davis, 1996.
8. [3] James A. Whittaker and Herbert H. Thompson, How to Break Software Security: Effective Techniques for Security Testing, Addison-Wesley, 2003.
9. [4] Common Vulnerabilities and Exposures. [Online] https://www.cvedetails.com/browse-by-date.php.(accessed on10/04/2016).
10. [5] K. Tsipenyuk, B. Chess, andG. McGraw, "Seven Pernicious Kingdoms: A Taxonomy of Software Security Errors," IEEE Sec. & Privacy, vol. 3, no. 6, Nov.-Dec. 2005, pp. 81-84.
11. [6] Lough, Daniel. "A Taxonomy of Computer Attacks with Applications to Wireless Networks," PhD thesis, Virginia Polytechnic Institute and State University, 2001.
12. [7] R. P. Abbott et al., "Security Analysis and Enhancements of Computer Operating Systems," Tech. rep. NBSIR 76-1041,L awrence Livermore Lab., Ins!.. for Compo Sci. and Tech.INa!'1. Bureau of Standards, RISOS Project,W ashington, DC, Apr. 1976.
13. [8] R. Bisbey II and D. Hollingworth, "Protection Analysis: Final Report," ISIISR-78-13, USClInfo. Sci. Inst., Marina Del Rey, CA, May 1978.
14. [9] C E. Landwehr et aI., "A Taxonomy of Computer Program Security Flaws," ACM Comp. Surveys, vol. 26, no. 3, Sept. 1994, pp. 211-2S4.
15. [10] T. Aslam, "A Taxonomy of Security Faults in the Unix Operating System," M.S. thesis, Dept. of Compo Sci., Purdue Univ., Coast TR 95-09,1995.
16. [11] I. Krsul, "Software Vulnerability Analysis," Ph.D. dissertation, Purdue Univ.,Coast TR 98-09,1 998.
17. [12] Howard, John D. and Longstaff, Thomas A. "A Common Language for Computer Security Incidents," Technical report, Sandia National Laboratories, 1998.
18. [13] M. Bishop, "A Taxonomy of UNIX System and Network Vulnerabilities," Tech. rep. CSE-9S10, Dept. of Compo Science, UC Davis, May 1995.
19. [14] W. Du and A. P. Mathur, "Categorization of Software Errors that Led to Security Breaches," Proc. 21st Nat .[Irifo. Sys. Sec. Con[, 1998.
20. [15] S. Kamara et aI., "Analysis of Vulnerabilities in Internet Firewalls," Compo & Sec., vol. 22, no. 3,2003,p p. 214-32.
21. [16] F. Piessens. A taxonomy of causes of software vulnerabilities in internet software[C]. Supp lementary P roceedings of the 13th International Sympo sium on Software Reliability Engineering,2002.
22. [17] A. Gray, "An Historical Perspective of Software Vulnerability Management," Irifo. Sec. Teh. Rep., vol. 8, no. 4, Apr. 2003, pp. 34- 44.
23. [18] K. Jiwnani and M. Zelkowitz, "Maintaining Software with a Security Perspective," Proc. Int'l Con Software Maintenance, 3-6 Oct. 2002, pp. 194--203.
24. [19] V Pothamsetty and B. Akyol, "A Vulnerability Taxonomy for Network Protocols Corresponding Engineering Best Practice Countermeasures," Proc. 3rd lASTED Int'! Con! Commun. lnternet, and Irifo. Tech., 2004, pp. 168-75
25. [20] S. Weber, P. A. Karger and A. Paradkar. A Software Flaw Taxonomy.Aiming Tools At Security Software Engineering for Secure Systems-Building Trustworthy Applications (SESS'05) 2005.
26. [21]Householder, A. D., Seacord, R. C, "A Structured Approach to ClassifYing Security Vulnerabilities," CMu/SEI-200S- TN-Om, January 2005.
27. [22] S. Hansman and R. Hunt, "A Taxonomy of Network and Computer Attacks," Compo & Sec., vol. 24, no. 1, Feb. 2005, pp. 31-43.
28. [23] Kjaerland, M., "A taxonomy and comparison of computer security incidents from the commercial and government sectors".Computers and Security, 25:522–538, October 2005.
29. [24] Anil Bazazl and James D. Arthur2, Towards A Taxonomy of Vulnerabilities, Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS'07), IEEE Computer Society, Hawaii, 2007.
30. [25] V. M. Igure and R. D. Williams, "Taxonomies of Attacks and vulnerabilities in Computer Systems", IEEE Communications Surveys & Tutorials 1st Quarter 2008.
31. [26] 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: http://issrl.cs.memphis.edu/files/papers/CyberAttackTaxonomy IEEE Mag.pdf
32. [27] Scott D., Angelos S," Towards a Cyber Conflict Taxonomy", 5th International Conference on Cyber Conflict K. Podins, J. Stinissen, M. Maybaum (Eds.), 2013