A Critical Review of Fault Tolerance: Security Perspective

Anshul Mishra[1], Dr. Devendra Agarwal[2], Dr. M. H. Khan[3]

School of Computer Application, BBDU[1], Lucknow, India
Director (Engg.) at BBDNIIT, BBDU[2], Lucknow, India
Professor, Department of C.S. E., I.E.T[3]., Lucknow, India

***Abstract*— This Security is an important issue of any software system. Security factors play a valuable and appropriate role in software security estimation process. Software security is affected with security attributes as well as fault. An effort through fault perspective is to identify the involved factors of fault and its probable impact on design parameters to quantify security. Fault tolerance is taken as key concept to security assessment. It is to be identified that qualifications of security attributes improved through inspect indemnity, discriminating, vulnerability and attacks in design development process. This research bridges the gap between object oriented design parameters, faults and security factors. In this paper fault is discussed as a security factor of software security. A constant state of the protected software enhances additional security.**

***Keywords*— Security, Fault, Fault Tolerance, Confidentiality, Integrity, Availability**

I. INTRODUCTION

Security plays an increasingly important role for software system. Security concern must inform every phase of software development from problem domain to solution domain [22, 2]. Software security estimates provides the help for degree of protection and assess the impact. Microsoft has stated that above 50% of the security related problem for any firm has been found at design level of software development process. Software security touch points are based on good software engineering and involve explicitly pondering security throughout the software lifecycle. Security estimation of software may heavily affect to security of the final product. The experts made an effort in this regards to develop the security estimation guidelines, view and concept. There are some probabilities that original code segment may have some security flaws, anomalies that may influence security at different phase.

To develop secure software system is the causative process of different steps and reflections of each phase are the matter of study to measure the perfect impacts of security. Security is a continuous process for every phase in development life cycle [6]. Security is a multidimensional attribute. Software security is about understanding software security risk and how to manage them. A quantitative approach can be much better than conceptual method to develop a technique which can assess the actual level of security assessment. Security enhancement techniques are extremely desirable for improving the internal structure, design simplicity and other feature of software. Before going for further discussion we will discuss about fault and try to drive relation between security and fault tolerance.

II. FAULT: SECURITY PERSPECTIVE

All Unscathed factor that is fault which determines the probability of occurrences attacks and also play a striking responsibility with other factors of security. Faults are common even today. Fault is the cause of software failure while failure is the effect that occurs as a result of the fault. Capability of fault is signified by the hiatus of unsecure software [3]. A security estimation process provides an accurate hint to measurement of software faults. The faults are found during the testing and the failure is when the system stops working. Fault attacks can be deployed in software which generally help to avoid, detect and correct faults. During development of software, faults and flaws are introduced either from the implementation or from the design of the software.

It relies on software attributes that how much system's software is protected; exactly the fault decides for how longer software will be protected. Fault is not only the factor that makes things hard to understand but with enough difficulty anything can become harder to understand. In this paper, we conduct a study regarding impact of fault to security estimation and their efficiency. There is need to develop a new approach to deal with a word of fault in software. Some other attributes of security is shown in figure 1. The required goal of security is to introduce measures and procedures that preserve confidentiality, integrity, availability, and other attributes such as authenticity, fault, and non-repudiation.

Figure 1. Security factors outlook

III. Correlation Between Security And Fault Tolerance

The security assessment is helpful for software developers, risk management team and executives of the company It definitely needs thoughtful subtle of security including security measurements, classifications and security attributes. Security attributes may decrease the cost and impinge between problem domain to solution domain at each phase of development life cycle [4, 5, and 20]. A level-2 heading must be in Italic, left-justified and numbered using an uppercase alphabetic letter followed by a period. For example, see heading “C. Section Headings” above.

Software Security is an external software attribute that reduces faults and effort required for secured software. Security must encompass dependable protection and secured the software system against all relevant concerns including confidentiality, integrity, availability, non repudiation, survivability, accessibility despite attempted compromises, preventing, misuse and reducing the consequences of unforeseen threats [13, 15]. Fault tolerance is direct associated to security attributes such as confidentiality, integrity, availability, non-repudiations, and survivability. Fault tolerance thought will efficiently improve the security. Fault tolerance is frequently essential, but it can be riskily error-prone because of the added efforts that must be involved in the programme procedure. A consistent quantitative estimate of security is highly enviable at an early stage of software development life cycle. Fault tolerance is direct associated to reliability and security.

Fault prevention and fault tolerance intend to present the ability to deliver an accurate service. Controlling and Monitoring can work mutually to enforce the security policy. Fault tolerance is the ability of a system to continue secures the software module and presence of software faults. The development of high assertion software has been dominated by work on two split theme: security and fault tolerance. Here, we introduced a conceptual summarize in term of errors, fault, and attack surface that measures observed effectiveness and simplicity of use with respect to security mean shown in Figure 2.

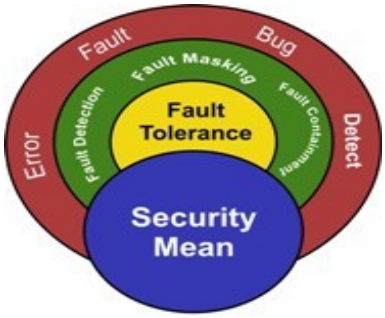

Figure 2. Affiliations between Security and Fault Tolerance

Fault tolerance attributes as a fault masking, fault detection and fault consideration effective to security policy [16, 12]. Fault tolerance implies a savings in development time, cost and efforts; also it reduces the number of components that must be originally developed.

IV. Related Work

The interest in software fault tolerance has been growing as software faults have become the major contributor to system failures. Study of security, experts says that fault, stability, complexity and reliability are an essential attributes which impinge on the depth of security. A long period of time fault tolerance has been used as a mean of improving and controlling the security of software system. Fault tolerance is the dynamic method that’s used to keep the interconnected systems together, sustain security and availability in systems [25]. Fault tolerance can be categorized in three level; Hardware, Software, and System faults [26].

Experts frequently supports and sustain that fault tolerance theory should be applied at early stage in the development process of software [10, 11]. Software fault tolerance is alarmed with techniques required to facilitate a system to tolerate software 'bugs' or faults. Fault tolerance is often old synonymously with graceful degradation which aims to detect, separate and resolve problems pre-emptively. Not all of the security faults existing in software systems are identifiable during the fault analysis [28, 40]. Many experts and researchers in the area recommended that fault tolerance is the ability of a system or component to continue normal operation despite the software fault and the similar is concise in table 1:

Sr. No.	Expert/ Researchers	Contribution with Fault tolerance
1.	Giovannie & Mauro 2002 [19]	➢ Demonstrate multivariate models for predict fault proneness of module of different software packages
2.	B.B. Madan & K.S. Trivedi 2003 [1]	➢ Quantified the security attributes of an intrusion tolerant system.
3.	A. Avizienis& C. Landwehr 2004 [21]	➢ Provided a new concept about security, fault tolerance and Dependability
4.	C. .N. Zhang & H. Zhong 2004 [8]	➢ Introduced an integrated approach for database security and fault tolerance
5.	R. Tirtea G. Deconinck & R. Belmans 2006 [18]	➢ Established relations between adaptive fault tolerance, quality of services and security
6.	Yuming 2007 [7]	➢ Empirical studies on design matrices are able to predict low security faults
7.	S. Bohacek, J. Hespanha, J. Lee, C. Lim & K. Obraczka 2007 [24]	➢ Developed a framework in terms of the security, fault tolerant, delay and throughput trade off
8.	D. B. Sharp, A. Nayak & N. Goel 2007 [27]	➢ Demonstrate a quantifiable definition for secure software with respect to integrity and fault
9.	A. R. Naghsh Nilchi, A. Vafaei & H. Hamidi 2008 [32]	➢ Introduced a qualitatively security considerations and challenges in application development with fault detection system
10.	Suraya & R. A. Khan 2009 [23]	➢ Provided the fault Proneness Model and validate for predict the fault
11.	R. Mehresh, Shambhu, J. Upadhyaya &	➢ Introduced a new approaches for analyzing the performance of a secure and

Sr. No.	Expert/ Researchers	Contribution with Fault tolerance
	K. Kwiat 2010 [33]	fault tolerant system
13.	D. Mougouei & M. Almasi 2012 [34]	➢ Introduced a measurement model for evaluating the degree of fault tolerance (FTMM) in security requirements
14.	C. Wang, C. Jiang & X. Liu 2012 [35]	➢ Developed Fuzzy logic based algorithms for secure and fault tolerant system
15.	D. F. T. Paz, J. Antonio P. Espinoza & J.Juan G. Hernandez 2014 [36]	➢ Proposed a low complexity distributed system for storing files in the cloud with fault tolerance and security
16.	M. Rodriguez & C. A. Kamhoua 2015 [30]	➢ Provided qualitative analysis of design diversity in fault tolerant and secure systems
17.	Hossein & Omid 2015[9]	➢ Demonstrate a new theme machine learning technique represents a reliable, fault tolerance, secure framework
18.	M. Rodriguez & A. Kwiat 2015 [31]	➢ Explored and Analyzed design diversity with respect to its fault tolerance
19.	M. Diaz & P. Henríquez 2016 [29]	➢ Developed a noval method for early bearing fault detection based on dynamic stability measure

In the context of all this investigation fault tolerance is considered as a security solution and its effect the activist direction to other security attributes. A number of researchers addressed the fault tolerance in the context of security and design. Fault tolerance is a noticeable matter for security. Figure 3 illustrates the fault tolerance contribution in different perspective such as method, framework, algorithms and concept.

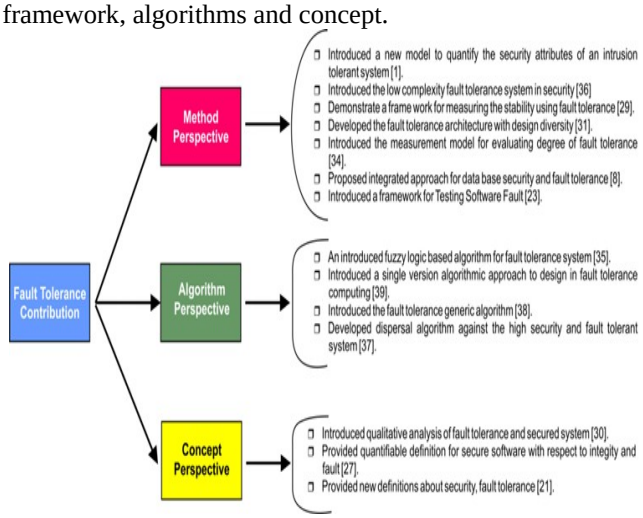

Figure 3. Fault Tolerance Contribution Taxonomy

V. PURPOSE

Security is key that helps to protect the data and resources. To improve security is to gain a better secured system. There is common agreement between researchers and security practitioners to integrate security at the preliminary stage of software development life cycle in order to develop secured software. [17]. Such as the necessity arises security factors were recognized and composed for their role in software development life cycle. Security estimation is achievable through the help of finding new security factors which directly and indirectly affects security features of software. Security is associated with three important security pillars which can be conveniently concise by the acronym CIA (confidentiality, integrity, availability) [14]. Software practioners, project managers, and developers remain under complete stress on account of their inability to deliver secure software system. This way of research is being appropriate at design process and aiming to improve one step in the safety of software. Security assessment is achievable with the help of finding innovative security factors which directly and indirectly affect the security features of software. The criterion for security estimation is a key to know the quality of software. Following steps form one such process when performed iteratively, incrementally, and in parallel with the other activities and tasks:

- Identify the Security Factors
- Identified fault tolerance factors having impact on security attributes and its behaviour are best suited in Object Oriented Design Perspective
- Identify Design Parameters and related metrics
- Analysis of best security practices
- Establishing the co-relation between security factors and design parameters with fault tolerance mechanism
- To establish a relation for quantified values

VI. CONCLUSIONS

Security assessment must be a compulsory at early stage of development life cycle. As such no framework, models, and metrics has been available in the literature that estimates software security of object oriented design by taking fault into consideration. Fault is parallel to availability, survivability and reliability. Fault tolerance is a complete result of security which contributes in measure the fault. Fault is not a single measurable property some other attributes acquaintances to security. This research paper concludes that how will give a view to show the effect of security factors with fault.

REFERENCES

[1] B. B. Madan and K. S. Trivedi, " A Method for Modelling and Quantifying the Security Attributes of Intrusion Tolerant System", ElSEVIER, DOI : 10.10/16/j.peva.2003.07.003, 2003.

[2] Davis, N. Humphrey and W. Redwine, "Mc. Graw and G. Processes for producing Secure Software Security and Privacy", IEEE, Vol.2, Issue 3, pp. 18-25, May 2004.

[3] S. A. Khan and R. A. Khan, "A framework to Quantify Security: Complexity Perspective", International Journal of Information and Education Technology, Vol 2, No.5, October 2012.

[4] Engineering Safety Requirements, Safety Constraints, and Safety Critical Requirements, Available at: http://www.jot.fm/issues/issue_2004_03/column3/ last visit Oct 17, 2014.

[5] T. Anderson and J. C. Knight, "A Framework for software Fault tolerance in Real time System", IEEE Transaction on software Engineering, Vol. 9, No.3, May 1983.

[6] R. A. Khan and S. A. Khan, "A Roadmap for Security", International Journal of Computer Science & Emerging Technologies", Vol. 1, Issue 1, June 2010.
[7] Y. Zhou and Hereton " Emperical analysis of Object Oriented Design Matrics for Predicting High and low Severity Faults", IEEE Transaction on software engineering, vol.32, Oct 2006.
[8] C. N. Zhang and H. Zhong, "An Integrated Approach for Database Security and Fault Tolerance", IEEE, DOI: 10.1109/ITCC.2004.1286560, 24 Aug 2004.
[9] H. Karimia and Omid, "Implementing a Reliable, Fault Tolerance and Secure Framework in the Wireless Sensor-Actuator Networks for Events Reporting", ELESIVER, International Conference on Advanced Wireless, Information, and Communication Technologies, 2015.
[10] R. K. Choudhary and R. Ahmed, "Classical Software Fault Tolerance Schemes by an Object-Oriented Technique", International Journal of Computing Science and Communication Technologies, Vol.1, No.1, July 2008.
[11] Wilfredo Torres, "Software Fault Tolerance: A Tutorial", 2000.
[12] W. Heimerdinger and C. Weinstock, "A Conceptual Framework for System Fault Tolerance", Technical Report CMU/ SEI-92-TR33, ESC-TR-92-033 SEI., OCT 1992.
[13] G. H. Walton, T.A. Long Taff and R. C. Linder, "Computational Evaluation of Software Security attributes", IEEE, 1997.
[14] M. Al. Kuwaiti, N. Kuriakopoulos and S. Hussain, "A Comparative Analysis of Network Dependability, Fault Tolerance, Relaibility, Security and Survivability", IEEE, Communication Survey and Tutorials, Vol.2, 2009.
[15] P. Neuman, "Practical Architecture for survivable system and networks", Phase Two Project 1688, SRI International, Menlo Park, California, June 2000.
[16] D. Prahan, "Fault Tolerance Computer System Design", 2nd Edition, Prentice Hall. Inc., Upper Saddle River NJ, pp. 5-14, 1996.
[17] S. A. Khan and R. A. Khan, "Integrity Quantification Model for Object Oriented Design", ACM SIGSOFT Software Engineering Notes, Vol. 37, No. 2, Mar 2012.
[18] R. Tirte, G. Deconinck & R. Belmans, "Fault Tolerance Adaptation Requirements vs. Quality of Service, Real Time and Security in Dynamic Distributed Systems", IEEE, DOI: 10.1109/RAMS.2006.1677390, 28 Aug 2006.
[19] Giovanni and Mauro, "An Emperical Evaluation of Fault Proneness Models", ACM, Digital Library, International Conference of Software engineering, 24th Conference, New York, May, 2002.
[20] A. D. T. Amma, V. R. Pramod and N. Radhika, "ISM for Analyzing the Interrelationship between the Inhibitors of Cloud Computing", vol. 2, No. 3, 2012.
[21] A. Avizienis & C. Landwehr, "Basic Concept and Taxonomy of Dependable and Secure Computing, IEEE, Doi 10.1109/itcc.2004.1286560, 2004.
[22] P. T. Devanbu and S. Stubblebine, "Software Engineering for Security: A Road Map", in Proceeding of the Conference on the future of Software Engineering, pp. 227-239, 2000.
[23] Surya and R. A. Khan, "Fault Proneness model of Object Oriented Software: Design Phase Perespective", International Conference onComputer Engineering and Application, Vol. 2, 2009.
[24] S. Bohacek, J. Hespanha, J. Lee, C. Lim & Katia Obraczka, "Game Theoretic Stochastic Routing for Fault Tolerance and Security in Computer Networks", IEEE, 2007.
[25] A. Sari and Murat A. K. Kaya, "Fault Tolerance Mechanism in Distributed Systems", Scientific Research Publication, pp. 471-482, 17 Dec 2015.
[26] Short and R.A. "The Attainment of Reliable Digital System through the use of Redudency- A Survey", IEEE, Computer Group News, pp.22-17, Jan 2006.
[27] D. B. Sharp and A. Nayak and N. Goel, "Fault Tolerance & Testable Software Security: A Method of Quantifiable Non Malleability with Respect to Time", IEEE, DOI: 10.1109/CCECE.2007.386, 2007.
[28] D. Mougouei, "Goalbased Modeling Approach to Develop Security Requirements of Fault Tolerant Security-Critical Systems", IEEE, Computer and Communication Engineering (ICCCE), Aug 2012.
[29] M. Diaz & P. Henríquez, "Novel Method for Early Bearing Fault Detection Based on Dynamic Stability Measure", IEEE, DOI: 10.1109/CCECE.2007.386, 2016.
[30] M. Rodriguez, K. A. Kwiat & C. A. Kamhoua, "Modeling Fault Tolerant Architectures with Design Diversity for Secure Systems", IEEE, DOI: 10.1109/MILCOM.2015.7357618, 17 Dec 2015.
[31] M. Rodriguez & A. Kwiat, "On the Use of Design Diversity in Fault Tolerant and Secure Systems: A Qualitative Analysis", IEEE, DOI: 10.1109/CISDA.2015.7208639, 2015.
[32] A. R. Naghsh Nilchi, A. Vafaei & H. Hamidi, "Evaluation of Security and Fault Tolerance in Mobile Agents" IEEE, DOI: 10.1109/WOCN.2008.4542509, 2008.
[33] R. Mehresh, Shambhu, J. Upadhyaya & K. Kwiat, "A Multistep Simulation Approach toward Secure Fault Tolerant System Evaluation", IEEE, 2010.
[34] D. Mougouei & M. Almasi, "A Goalbased Modeling Approach to Develop Security Requirements of Fault Tolerant Security Critical Systems", IEEE, 2012.
[35] C. Wang, C. Jiang & X. Liu, "Fuzzy Logic Based Secure and Fault Tolerant Job Scheduling in Grid", IEEE, Vol. 1, Issue S1, 2012.
[36] D. F. T. Paz, J. Antonio P. Espinoza & J.Juan G. Hernandez, " A Low Complexity Fault Tolerant Document Storage System", IEEE, 2014.
[37] H. Lahkar and M. C R, " Toward High Security and Fault Tolerance Dispersed Storage System with optimized Information Dispersal Algorithm", Vol.2, Issue 2, 2014.
[38] X. Pia, Z. Xingshe, "Security Driven Fault Tolerant Scheduling Algorithm for High Dependable Distributed Real Time System", IEEE, 2012.
[39] G. K. Saha, "A Single-Version Algorithmic Approach to Fault Tolerant Computing Using Static Redundancy", CLEI ELECTRONIC JOURNAL, Vol. 9, No. 2, DEC 2006.
[40] S. A. Khan and R. A. Khan, **"**Object Oriented Design Security Quantification", Journal of Global Research in Computer Science, Volume 2, No. 4, 2011.

BIOGRAPHY

Anshul received the MCA degree from Dr. R. M. L. Avadh University, Faizabad, in 2008. He is enrolled as research scholar in BBDU , Lucknow. His research interests include Software testability, Software Quality Estimation, Data Dictionary.

Dr. Devendera Agarwal is currently working as Prof. & Director (Engg.) at BBDNIIT (BBD Group), Lucknow. He has over 17 years of teaching & 5 years of industrial experience. He has done his B.Tech in Computer Science from Mangalore University in 1993, M.Tech from U.P. Technical University, Lucknow in 2006, and Ph.D. from Shobhit University, Meerut in 2013. He has over 10 research papers with 4 students pursuing Ph.D.

Dr. M. H. Khan, Professor, Department of Computer Science and Engineering at IET Lucknow UP. Obtained his MCA degree from Aligarh Muslim University (Central University) in 1991. Later he did his PhD from Lucknow University. He has around 28 years rich teaching experience at UG and PG level. His area of research is Software Engineering. Dr. Khan published numerous articles, several papers in the National and International Journals and conference proceedings.