Security Assurance – The Requirements Way

S. K. Pandey, S. Rehman, K. Mustafa, S. I. Ahson
Department of Computer Science
Jamia Millia Islamia (Central University), New Delhi-110025, INDIA
santo.panday@yahoo.co.in, shabana.infosec@gmail.com,kmfarooki@yahoo.com,drsiahson@yahoo.com

Software Engineering technologies seem to support, and demand as well, the adequate level of security assurance in software projects. Requirements are considered as foundation stone on which the entire software is built. In earlier days, the requirements phase was not taken seriously, which caused the many big software problems. These problems' nature and quality both continue to grow exponentially with the growth in software complexity and its versatility. There is no doubt that security is now a 'vibrant burning issue'. It needs to address prominently and no escape way is feasible. The failure and success of any software depends upon the quality of requirements. It is observed that about 71% of the software is not completed due to poor requirements [1] [3]. Poor requirements-definition is responsible for almost half of the failures when it comes to translating what users need into ICT reality [2]. A 2003 research report from Meta Group (since acquired by Gartner) indicates that 60%-70% of software development outsourcing failures in global 2000 companies are due to poor requirements gathering, analysis and planning [4]. Studies indicate that more than 60% failure rate for software projects in the US, with poor requirements as one of the top five reasons. Studies also show a high percentage of project schedules overruns, with 80% due to creeping requirements [5].

Contrary to the perception, experts are now of the opinion that security cannot be feasibly added into an exiting system [5]. It is an emergent property that requires advance planning during requirements phase with careful design. Earlier Software Security was an after thought, which used to compound itself during later stages. Generally, it used to be taken as post development process; and had been a matter of concern only when s penetrated by attackers. Barry Boehm and Victor R. Basili, famous software experts from University of South California and University of Maryland observed that finding and fixing a software problem after delivery is often 100 times more expensive than finding and fixing it during the requirements and design phase [6]. But now, the need to consider security from the ground up is a fundamental tenet of secure system development [7]. We can reduce the cost and efforts by implementing the security aspect right from beginning i.e. from requirement phase onwards.

If you have taken care of security in requirements phase ,then this could be avoided.
Attacks
Bug
Virus
Vulnerability
Errors

The importance of the requirements engineering has been well recognized and now several researches are underway on ‘ways to incorporate security right from beginning’. The requirements phase is the foremost opportunity for the product team to consider how security will be integrated into a development process, identify key security objectives and otherwise maximize software security [7]. In continuation to this process, the team should need to consider how the security features and assurance measures of its software will integrate with other software, likely to be used together with its software. The requirements team’s overall perspective of security goals, challenges, policies, and plans need to be incorporated in the SRS that is produced during the requirement’s phase.

The requirements are more often incomplete mainly due to invertible changes in those requirements or because Stakeholders do not see a pressing need for a particular aspect of the software, such as security. The possibility of integrating security right from requirements phase necessarily depends on the specifier’s ability to anticipate and model with a high degree of accuracy and thoroughness the full range of different environment states with which the software will be confronted [8]. Requirements level security assurance is possible through vaccination of the SRS, which means maximum number of vulnerabilities and threats corresponding to each requirement can be taken care of right from requirements phase itself. There is no doubt that to ensure better security assurance processes, a *precisely specified and highly prescriptive* ‘framework, method ,or road map’ are generally used and have been found to be handy and quite fruitful [9]. Further, it become evident through the explanation of the researchers that a little work on this pertinent issue has been reported. So, there is a necessity to have a security framework that should be prescriptive in nature and can be easily usable to assure secure development processes.

In 1990, failure due to a single line of buggy code in AT & T’s 4ESS switch caused systems drop roughly 50% of long distance over a period of nine hours and $60 million loss [10][11]. Another incident of computer security reported to the CERT coordination center in recent years due to a single class of programming flaws buffer overruns [12]. USA alone looses about $38B in security lapses and tracking of virus incidents alone runs into the range of $80B per year worldwide [13]. These losses are incurred despite an estimated security market size of $36B expected by the year 2007-08 [13]. There are no foolproof solutions in sight. As discussed earlier, finding and fixing a software problem after delivery is often 100 times more expensive than finding and fixing it during the requirements and design phase [6]. The idea to introduce security 'right from beginning' may well reduce the cost and provide the security assurance as well. It will also be helpful to increase the productivity and to reduce security engineering in later phases. Hence there is no doubt to say that cost effective and efficient security assurance can be achieved by secure requirements.

References

[1] John Pescatore, "First Take FT-23-5794", Gartner Research, July 2004.

[2] Stephen Bell Wellington: "Poor requirements-definition equals ICT failure", Computer World, Thursday, 9 November, 2006.

[3] "Stop the seeds of project failure", BCS Project Management Article, www.bcs.org, September 2007.

[4] Nari Kannan , CEO and co-founder of Ajira "Agile Outsourcing: Requirements Gathering and Agile Methodologies" http://www.sourcingmag.com/content/c061002a.asp

[5] An Innovative Approach to managing Software Requirement http://projectmanagement.knowledgestorm.com/shared/write/collateral/WTP/49705_52374_26971_MKS.pdf?ksi=1290251&ksc=1298777634

[6] Barry Boehm, Victor R. Basili, "Software Defect Reduction Top 10 List", Software Management, Jan 2001, pp 135-137.

[7] Steve Lipner, Michael Howard, "The Trustworthy Computing Security Development Lifecycle", Microsoft Corporation, 2006.

[8] The Information System Security Engineering Process, Chapter-3, IATF Release 3.1- Sep 2002.

[9] Raees A. Khan: "Quality Estimation of Object Oriented Code-A Design Metrics Perspective-", Ph.D. Thesis, Department of Computer Science, Jamia Millia Islamia, May 2004.

[10] I. Peterson, "Fatal Defect: Chasing Killer Computer Bugs", Vintage Books, New York, pp. 210-216, 1996.

[11] Anup K. Ghosh, "Addressing New Security and Privacy Challenges, IT Pro pp. 10-11, May/June 2002.

[12] C. Cowan & Coleagues, "Stachgard: Automatic Adaptive Detection and Prevention of Buffer-Overflow attack", Proc. 7th usenix Security Symp., Usenix Assoc, San Diego, Calif, 1998.

[13] Prem Chand, "Building India as the Destination for Secure Software Development – Next Wave of Opportunities for the ICT Industry", LNCS Volume 3803/2005, pp 49-65, Springer Berlin / Heidelberg, 2005.