Recent Advances in SRE Research

S. K. Pandey[1], K. Mustafa [2]

[1]Department of Information Technology
Board of Studies, The Institute of Chartered Accountants of India, Noida- 201 301
E mail: santo.panday@yahoo.co.in
[2]Department of Computer Science
Jamia Millia Islamia (Central University), New Delhi-110 025
E mail: kmfarooki@yahoo.com

Abstract: The importance and the real potential of the Requirements Engineering is now being well recognized. A series of reversed as well as assorted researches are underway particularly on 'ways to incorporate security *right from the beginning*'. Researchers are doing excellent work in this area. In this paper, we review current Secure Requirements Engineering (SRE) research and try to identify current research directions, based on the recently published work. The research is considered with respect to technologies developed to address specific requirement tasks, such as elicitation, modeling, and analysis. Such a review enables us to identify mature areas of research, as well as areas that warrant further investigation. Finally, we highlight the hot current and future research topics, of significance and present a critical analysis.

Keywords: Software Security, Secure Requirements Engineering (SRE), Recent Work in SRE, Future Directions in SRE.

I. INTRODUCTION

Requirements are considered as foundation stone on which the entire software can be built. In earlier days, the requirements phase was not taken seriously, which caused many big software problems. These problems' nature and quality both continue to grow exponentially with the growth in software complexity and its versatility. The failure and success of any software depends upon the quality of requirements. It has been reported that about 70% of the software is not completed due to poor requirements [1] [3]. Poor requirements-definition is generally held 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 more than 70% of software development outsourcing failures in global 2000 companies is 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 added into an exiting system [5]. It is an emergent property that requires advance and rigorous planning during requirements phase with careful design. Earlier Software Security used to be considered only 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 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' phases [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.

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]. During this process, the team should consider how the essential and desirable 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, and plans needs to be incorporated in the SRS that is produced during the requirement phase.

In this paper, we review recent research directions in requirements engineering. The rest of the paper is organized as follows: In Section II, current research in SRE is briefly reported, whereas in Section III, we present the future research directions. Conclusion is reported in Section IV.

II. A SURVEY OF SRE RESEARCH

Various researchers are underway on ‘The different aspects of SRE’. However a rapid growth has been visualized recently. Some significant contributions bear weight and appear valuable among all. A selection from the trend setting research contributions are briefly described one by one for analysis on the advances, as follows:

Nancy R. Mead presented SQUARE method for requirements [8]. The Security Quality Requirements Engineering (SQUARE) method provides a systematic way to identify security requirements in a software development project. Authors described SQUARE and then discussed other methods used for identifying security requirements, such as the Comprehensive, Lightweight Application Security Process, the Security Requirements Engineering Process, and Tropos, and compare them with SQUARE [8].

Ivan Flechais, Cecilia Mascolo, M. Angela Sasse presented some ways for integrating security and usability into the requirements and design process [9]. In this paper, authors describe Appropriate and Effective Guidance for Information Security (AEGIS), a methodology for the development of secure and usable systems. AEGIS defines a development process and a UML meta-model of the definition and the reasoning over the system’s assets. AEGIS has been applied to case studies in the area of Grid computing [9].

Mamadou H. Diallo, Jose Romero-Mariona, Susan Elliott Sim, and Debra J. Richardson presented a comparative evaluation of three approaches: The Common Criteria, Misuse Cases, and Attack Trees [10]. They applied each of these approaches to a common problem, a wireless hotspot, and evaluated them for learnability, usability, completeness, clarity of output, and analyzability. They found that each approach has strengths and weaknesses, and that they can be complimentary when combined. The Common Criteria are difficult to learn and use, but are easy to analyze. Misuse Cases are easy to learn and use, but produces output that is hard to read. In contrast, Attack Trees produce clear output, but are difficult to analyze [10].

Sun-myung Hwang proposed some intelligent methods and procedures considering financial aspects and reducing the threat to the system by applying security engineering, and for building security countermeasure [11]. Requirements in security area are not same as with other research areas. Security-related requirements are listed into Protection Profile (PP). A protection profile defines an implementation-independent set of security requirements for a category of Target of Evaluations. Generally, PP contains functional requirements and security assurance requirements about the security of development environment for IT product or system and PP can be applied to development site. Sun-myung Hwang proposed some security-related check points for development site. It can be included into PP by analyzing ISO/IEC 15408 and ISO/IEC 21827 [12].

Johan Gr´egoire, Koen Buyens, Bart De Win, Riccardo Scandariato, Wouter Joosen presented a comparison between two processes [13]. In this paper, two high-profile processes for the development of secure software, namely OWASP’s CLASP and Microsoft’s SDL, are evaluated and compared in detail. The paper identifies the commonalities, discusses the specificity of each approach, and proposes suggestions for improvement [13]. Reijo Savola introduced a preliminary framework for security evaluation based on security requirement definition, behavior modeling and evidence collection [14].

Betty H. C. Cheng, Joanne M. Atlee reviewed current Requirements Engineering (RE) research and identify future research directions suggested by emerging software needs. First, they overviewed practitioners and experts on the state of the art in RE research. The research is considered with respect to technologies developed to address specific requirements tasks, such as elicitation, modeling, and analysis. Such a review enables us to identify mature areas of research, as well as areas that warrant further investigation. Next, they reviewed several strategies for performing and extending RE research results, to help delineate the scope of future research directions. Finally, they highlighted what they considered to be the “hot” current and future research topics, which aim to address RE needs for emerging systems of the future [15].

Chandan Mazumdar, et al proposed a quantitative information security risk analysis methodology [16]. The proposed methodology incorporates two approaches. The consolidated approach identifies risk as a single value for each asset. The detailed approach identifies the threat-vulnerability pair responsible for a risk and computes a risk factor corresponding to each security parameter for every asset. Once the risks to an asset are identified, appropriate safeguards in the form of tools installed at appropriate locations or the application of policies, guidelines and procedures by the management can be employed to protect the asset [16].

Corey Hirsch and Jean- Noel Ezingeard presented a case study on perceptual and cultural aspects of risk management alignment [17]. This case study offers an illustration of the mixed formal and informal ERM culture alignment mechanisms, ranging from committee structures to security fairs, surveys to spreadsheets [17].

Khaled M. Khan and JunHan specified security goals of Component Based Systems: with an end-user perspective [2]. This paper treats security from a software engineering point of view. Security issues of software components are usually handled at the two levels of development abstractions: by the security experts during the component design, and by the software engineers during the composition of an application system. Security experts identify the threats of the component, define the security policies and functions. On the other hand, the software engineers are more interested in the compositional impact and conformity of the security properties designed and implemented by the security experts. This paper identifies a third level of abstraction: security from the end-users' perspective. This paper argues that the end- users of the system should know the specific security objectives actually achieved at the system-level. This paper makes the following three specific contributions in this regard: (i)a need for a separate view of security at the end-user level; (ii) the formulation of security goals; (iii) the derivation of security goals for automatic processing [18].

Ashish Agarwal and Dr. Daya Gupta defined a process for security requirements elicitation, presenting techniques for activities like requirements discovery, analysis, prioritization and management [19]. This approach is different from Misuse Case approach and common criteria. As they consider both functional and non functional requirements to derive security requirements. Also discovering security requirements is integrated with discovering functional and non functional requirements. As mentioned in generic CAME Tool, MERU will be extended to help method engineer in construction of method which include security mechanism [19].

Jon Whittle, Duminda Wijesekera, and Mark Hartong presented a process for executable Misuse Cases for Modeling Security Concerns [20]. In this paper, they presented an executable misuse case modeling language which allows modelers to specify misuse case scenarios in a formal yet intuitive way and to execute the misuse case model in tandem with a corresponding use case model. Misuse scenarios are given in executable form and mitigations are captured using aspect-oriented modeling. The technique is useful for brainstorming potential attacks and their mitigations. Furthermore, the use of aspects allows mitigations to be maintained separately from the core system model. The paper, supported by a UML-based modeling tool, describes an application to two case studies, providing evidence that the technique can support red-teaming of security requirements for realistic systems [20].

Edward Bonver, and Michael Cohen emphasized for developing and retaining a security testing mindset [21]. This program stresses on continuous education through the use of a dedicated task force a security testing knowledge portal. Both help emphasizes the idea of thinking like an attacker by looking beyond the surface of a system's features and really questioning where vulnerabilities could exist and how they could be exploited [21].

Laurie Williams, Michael Gegick, and Andrew Meneely presented a methodology for Protection Poker of Structuring Software Security Risk Assessment and Knowledge Transfer [22]. They proposed the Protection Poker activity as a collaborative and informal form of misuse case development and threat modeling that plays off the diversity of knowledge and perspective of the participants. An excellent outcome of Protection Poker is that security knowledge passed around the team. Students in an advanced undergraduate software engineering course at North Carolina State University participated in a Protection Poker session conducted as a laboratory exercise. Students actively shared misuse cases, threat models, and their limited software security expertise as they discussed vulnerabilities in their course project. They observed students relating vulnerabilities to the business impacts of the system. Protection Poker lead to a more effective software security learning experience than in prior semesters. A pilot of the use of Protection Poker with an industrial partner will begin in October 2008 [22].

Bart De Win, Riccardo Scandariato, Koen Buyens, Johan Gr´egoire, and Wouter Joosen compared the CLASP, SDL and Touchpoints for the Secure Software Development Process [23]. In this paper, three high-profile processes for the development of secure software, namely OWASP's CLASP, Microsoft's SDL and McGraw's Touchpoints, are evaluated and compared in detail. The paper identifies the commonalities, discusses the specificity of each approach, and proposes suggestions for improvement [23].

[24] illustrates 'how software application security can be best, and most cost-effectively, achieved when developers monitor and regulate risks early on', integrating assessment and management into the development life cycle. This book identifies the two primary reasons for inadequate security safeguards: Development teams are not sufficiently trained to identify risks; and developers falsely believe that pre-existing perimeter security controls are adequate to protect newer software. Examining current trends, as well as problems that have plagued software security for more than a decade, this useful guide:

- Outlines and compares various techniques to assess, identify, and manage security risks and vulnerabilities, with step-by-step instruction on how to execute each approach

- Explains the fundamental terms related to the security process
- Elaborates on the pros and cons of each method, phase by phase, to help readers select the one that best suits their needs

Despite decades of extraordinary growth in software development, many open-source, government, regulatory, and industry organizations have been slow to adopt new application safety controls, hesitant to take on the added expense. This book improves understanding of the security environment and the need for safety measures. It shows readers how to analyze relevant threats to their applications and then implement time- and money-saving techniques to safeguard them [24].

Marco D. Aime, Andrea Atzeni, and Paolo C. Pomi presented AMBRA - Automated Model-Based Risk Analysis [25]. In this work, they showed a methodology based on existing standards, highlighting tasks automatically-performable, and describe how it is possible to automate these aspects in our model [25].

Jose Romero-Mariona, Hadar Ziv, and Debra J. Richardson presented a process CCARCH for Architecting Common Criteria Security Requirements [26]. They focused on a technique known as the Common Criteria, which allows for the development of security requirements. They extended the capabilities of Common Criteria beyond the requirements phase, to allow us to take security requirements into further stages of the cycle. In this paper, they describe CCARCH, a technique accompanied by a set of tools, that takes Common Criteria expressed security requirements to the architectural level. Their approach aids in making the usage of Common Criteria more beneficial and applicable [26].

Idongesit Mkpong-Ruffin, David Umphress, John Hamilton, and Juan Gilbert presented a model for Quantitative Software Security Risk Assessment [27]. This research uses empirical data that reflects the security posture of each vulnerability to calculate Loss Expectancy; a risk impact estimator. Data from open source vulnerability databases and results of predicted threat models are used as input to the risk model. Security factors that take into account the innate characteristics of each vulnerability are incorporated into the calculation of the risk model; resulting in an empirical assessment of the potential threats to a development effort based on the risk metric calculation [27].

Ivan Flechais, Cecilia Mascolo and M. Angela Sasse presented a process for integrating security and usability into the requirements and design process [28]. AEGIS has been presented as a development process that provides both usability and security. Through the definition of MOF-compliant semantics, they have described an asset model notation, capable of documenting security requirements. By modelling the context in which the system operates and the interactions of the operatives and the assets of the system, this notation also allows the documentation of usability needs. Finally, they have presented a case study in which AEGIS was taught and applied to a grid project. The case study highlighted that AEGIS is easy to learn, provides a clear means of documenting security requirements and is useful in identifying the role and importance of operatives in the system. Future work may include identifying issues concerning the resolution of conflicts in security requirements gathering, incorporating decision making support, improving tools support for AEGIS and also integrating AEGIS into Model Driven Architectures (Object Management Group, 2004) [28].

Haralambos Mouratidis presented a manifesto for Secure information systems engineering [29]. In this paper, they lay down the agenda for a discipline that is meant to promote research on increasing the development of secure information systems. In particular, they introduce areas related to the development of secure information systems; they identify limitations of existing approaches and the barriers that currently limit research and they discuss the characteristics for an engineering discipline for the development of secure information systems, its principles and the challenges that must be addressed [29].

Requirements in security area are not same with other research areas. Security-related requirements are listed into Protection Profile (PP). A protection profile defines an implementation-independent set of security requirements for a category of Target of Evaluations. Generally, PP contains functional requirements and security assurance requirements about the security of development environment for IT product or system and PP can applied to development site. Sun-myung Hwang proposed some security-related check points for development site can be included into PP by analyzing ISO/IEC 15408 and ISO/IEC 21827 [30].

Reijo Savola introduced a preliminary framework for security evaluation based on security requirement definition, behavior modeling and evidence collection [31].

Betty H. C. Cheng, Joanne M. Atlee reviewed current requirements engineering (RE) research and identified future research directions suggested by emerging software needs. First, they overviewed the state of the art in RE research. The research is considered with respect to technologies developed to address specific requirements tasks, such as elicitation, modeling, and analysis. Such a review enables us to identify mature areas of research, as well as areas that warrant further investigation. Next, they reviewed several strategies for performing and extending RE research results, to help delineate the scope of future research directions. Finally, they highlighted what they considered to be the “hot” current and future research topics, which aim to address RE needs for emerging systems of the future [32].

Charles B. Haley, Jonathan D. Moffett, Robin Laney, and Bashar Nuseibeh presented a Framework for Security Requirements Engineering [23]. This paper presents a framework for security requirements elicitation and analysis, based upon the construction of a context for the system and satisfaction arguments for the security of the system. One starts with enumeration of security goals based on assets in the system. These goals are used to derive security requirements in the form of constraints. The system context is described using a problem-centered notation, then this context is validated against the security requirements through construction of a satisfaction argument. The satisfaction argument is in two parts: a formal argument that the system can meet its security requirements, and a structured informal argument supporting the assumptions expressed in the formal argument. The construction of the satisfaction argument may fail, revealing either that the security requirement cannot be satisfied in the context, or that the context does not contain sufficient information to develop the argument. In this case, designers and architects are asked to provide additional design information to resolve the problems [33].

Guttorm Sindre Æ Andreas L. Opdahl presented a methodology for Eliciting security requirements with misuse cases [34]. This paper presents a systematic approach to eliciting security requirements based on use cases, with emphasis on description and method guidelines. The approach extends traditional use cases to also cover misuse, and is potentially useful for several other types of extra-functional requirements beyond security [34].

Tim Grance, Joan Hash, and Marc Stevens presented an idea for security considerations in the Information System Development Life Cycle [35]. A general SDLC is discussed in this guide that includes the following phases: initiation, acquisition/development, implementation, operations/maintenance, and disposition. Each of these five phases includes a minimum set of security steps needed to effectively incorporate security into a system during its development. An organization will either use the general SDLC described in this document or will have developed a tailored SDLC that meets their specific needs. In either case, NIST recommends that organizations incorporate the associated IT security steps of this general SDLC into their development process [35].

Security 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 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’ phases. But now, the need to consider security from the ground up is a fundamental tenet of secure system development [35].

III. RESEARCH DIRECTIONS IN SRE

SRE is a very active research area, with a wide variety of methods. At present, there is no consensus on a single ‘the best approach’ to security requirements engineering. However, many organizations intuitively feel that attention to this area will pay off in supporting their business goals [8]. Moreover, future work may include identifying issues concerning the resolution of conflicts in security requirements gathering, incorporating decision making support, and also integrating AEGIS into Model Driven Architectures [9].

There appears a need for the development of a framework that would assist in combining multiple security requirement methods. This framework could also provide guidance on when to use a particular technique or representation. Other future work includes extending the evaluation of requirements techniques. In this study they only considered specification, but they still need to consider elicitation, analysis, and traceability [10]. Authors proposed some methods for reducing the threat to the system by applying security engineering, and proposed a method for building security countermeasure [11]. But this method can’t cover all the cases. Therefore, more detailed research is needed, and the research for generalizing these processes may be proceeded, too [11].

In these days, some security countermeasures are used to protect development site. But the security countermeasures ought to be considered with consideration of applicable threats and security solutions deployed to support appropriate security services and objectives. Maybe this is one of our future works [12]. Some work may be initiated on combining the strong points of both approaches in order to develop an improved, consolidated process. This requires addressing, as well as validating, most of the areas of improvement that were discussed in the paper [13].

The current state of- the-art practice is limited to a too high abstraction level. In addition, the semantics of the transformation from non-functional aspects to the behavioral aspects requires future work [14]. Other potential RE tasks are to identify and document potential security threats. Specifically, the specifier identifies assets, identifies vulnerabilities in the context of potential threats, and specifies countermeasures to protect against these threats [15].

Although strategic, the threat-based approach to security requirements engineering is reactive and focuses on low level security requirements; there is no notion of a general security policy. An alternative approach would take a top down view of security requirements, and base requirements on organizational structures, such as lines of authority separation of duties, delegation, roles, groups, access policies, and so on [15].

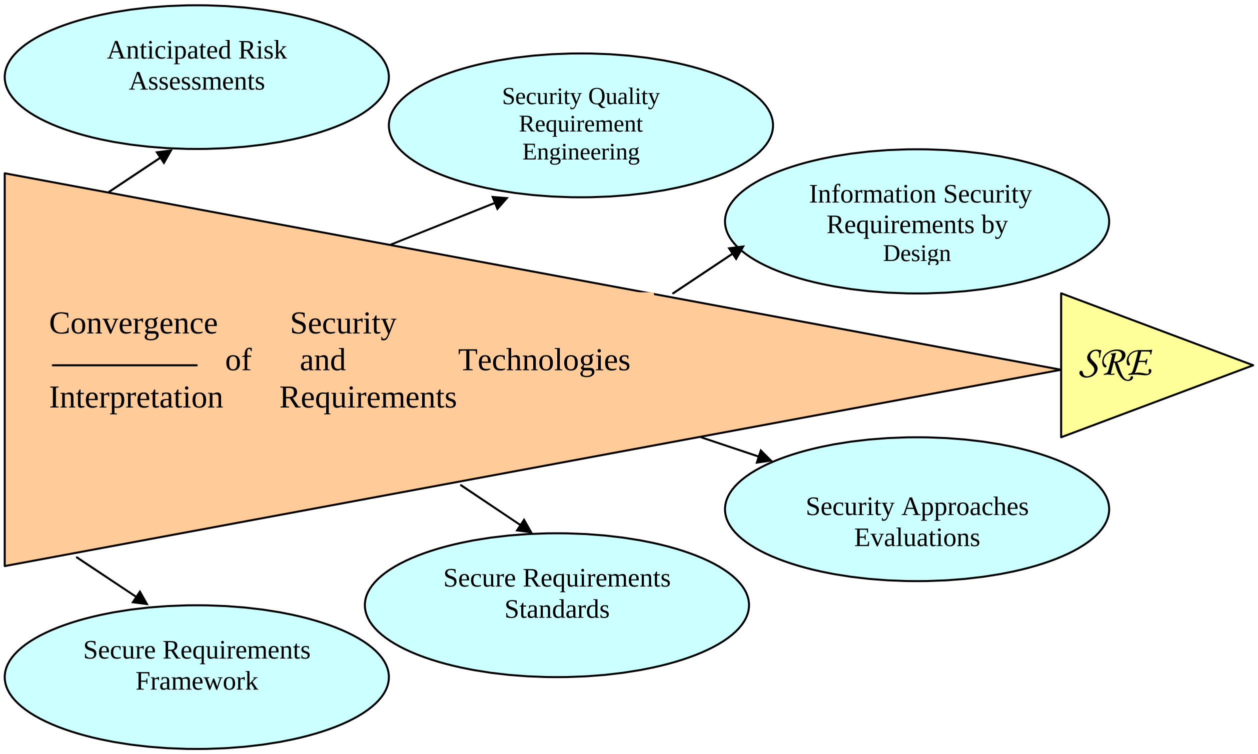
Anticipated Risk Assessments
Security Quality Requirement Engineering
Information Security Requirements by Design
Convergence ——— Interpretation of Security and Requirements Technologies
SRE
Security Approaches Evaluations
Secure Requirements Standards
Secure Requirements Framework

Fig. 1: SRE Research Directions

There is no consensus on the degree to which security requirements should be realized at the requirements level. Should specifiers go so far as to select and employ appropriate protections for identified threats, in the manner that user interfaces and timing deadlines are woven into behavioral specifications? Or should detailed security measures be optimized at design time along with other competing on functional requirements? These are open questions for the RE and security communities to resolve [15]. There is no doubt that further research is needed on ‘how to integrate RE technologies’, so that practitioners know ‘how to apply individual technologies effectively and synergistically’ [15].

IV. CONCLUSION

SRE research community has made significant progress along many fronts. At the same time, the demands placed on computing and the cyber infrastructure has increased dramatically, raising many new critical SRE research questions. Keeping in view, we presented a number of research areas in which further work is required, based on the published work of the year 2006, 07, 08, and 09. Realizing the SRE research significance, the efforts appear to be far less than desired. However, it is evident that directions being reported are conclusive, effective and efficient ways to incorporate security *right from the beginning* in the development life cycle.

REFERENCES

[1] James E. Powell, “IT pays a price for poor requirements practices”, Newswire, Enterprise Systems, Feb-7, 2008.

[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/a.knowledgestorm.com/shared/write/collateral/WTP/49705_52374_26971_MKS.pdf?ksi=129025 1&ksc=1298777634 (downloadable, March, 2007)

[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] Nancy R. Mead: How to compare the Security Quality Requirements Engineering (SQUARE) Method with Other Methods, Technical Note, CMU/SEI-2007-TN-021, Software Engineering Institute, Carnegie Mellon University.

[9] Ivan Flechais, Cecilia Mascolo, M. Angela Sasse: Integrating security and usability into the requirements and design process, International Journal of Electronic Security and Digital Forensics, Vol. 1, no. 1, 2007, pp 12-26.

[10] Mamadou H. Diallo, Jose Romero-Mariona, Susan Elliott Sim, and Debra J. Richardson: A Comparative Evaluation of Three Approaches to Specifying Security Requirements,www.di.unipi.it/REFSQ06/Papers/02%20Diallo.pdf (downloadable, March, 2007)

[11] Sun-myung Hwang: Intelligent Methods and Procedures of Countermeasure Design, in the proceedings of the IEEE International Conference on Multimedia and Ubiquitous Engineering 2007 (MUE'07) IEEE- 0-7695-2777-9/07, pp 5-17.

[12] Sun-myung Hwang, Special Checklist for Security Requirements in Software Development Site, in the proceedings of the IEEE International Conference on Multimedia and Ubiquitous Engineering 2007 (MUE'07) IEEE 0-7695-2777-9/07, pp 1172-1176.

[13] Johan Gr´egoire, Koen Buyens, Bart De Win, Riccardo Scandariato, Wouter Joosen: On the Secure Software Development Process: CLASP and SDL Compared, in the proceedings of the Third IEEE International Workshop on Software Engineering for Secure Systems 2007 (SESS'07) IEEE0-7695-2952-6/07.

[14] Reijo Savola: Requirement Centric Security Evaluation of Software Intensive Systems, in the proceedings of the IEEE 2nd International Conference on Dependability of Computer Systems, 2007, IEEE-0-7695-2850-3/07, pp 135-144.

[15] Betty H. C. Cheng, Joanne M. Atlee: Research Directions in Requirements Engineering, in the proceedings of the IEEE Conference on Future of Software Engineering 2007(FOSE'07), IEEE-0-7695-2829-5/07, pp 285-303.

[16] Chandan Mazumdar, Mridul Sankar Barik, Anirban Sengupta: Enterprise Information Security Risk Analysis: A Quantitative Methodology, Proceedings of the National Workshop on Software Security (NWSS 2007), N. Delhi, India, pp 1-12, 2007.

[17] Corey Hirsch, Jean- Noel Ezingeard: Perceptual and cultural aspects of risk management alignment: a case study, in the Journal of Information Systems Security, JISSec 4(1), Jan 2008, pp 3-20.

[18] Khaled M. Khan, JunHan: Specifying Security Goals of Component Based Systems: An End-User Perspective, in the proceedings of the IEEE Seventh International Conference on Composition-Based Software Systems, May 2008, pp 101-109.

[19] Ashish Agarwal, Dr. Daya Gupta: Security Requirements Elicitation Using View Points for Online System, in the proceedings of the IEEE First International Conference on Emerging Trends in Engineering and Technology, July 2008, pp 1238-1243.

[20] Jon Whittle, Duminda Wijesekera, Mark Hartong: Executable Misuse Cases for Modeling Security Concerns, in the proceedings of the ICSE'08, Leipzig, Germany, May 10–18, 2008, pp 121-130.

[21] Edward Bonver, Michael Cohen: Developing and retaining a security testing mindset, IEEE Security and Privacy, May 2008, pp 82-85.

[22] Laurie Williams, Michael Gegick, and Andrew Meneely: Protection Poker of Structuring Software Security Risk Assessment and Knowledge Transfer, ftp://ftp.ncsu.edu/pub/unity/lockers/ftp/csc_anon/tech/2008/TR-2008-21.pdf

[23] Bart De Win, Riccardo Scandariato, Koen Buyens, Johan Gr´egoire, and Wouter Joosen: On the Secure Software Development Process: CLASP, SDL and Touchpoints Compared, Preprint submitted to Elsevier, 16 January 2008.

[24] Douglas A. Ashbaugh: Security Software development, Assessing and Managing Security Risk, CRC Press, 23rd Oct, 2008.

[25] Marco D. Aime, Andrea Atzeni, and Paolo C. Pomi: AMBRA - Automated Model-Based Risk Analysis, in the proceedings of the QoP'07, Alexandria,Virginia, USA, October29, 2007, pp 43-48.

[26] Jose Romero-Mariona, Hadar Ziv, Debra J. Richardson: CCARCH: Architecting Common Criteria Security Requirements, in the proceedings of the Third International Symposium on Information Assurance and Security, July 2007, pp 349-354.

[27] Idongesit Mkpong-Ruffin, David Umphress, John Hamilton, Juan Gilbert: Quantitative Software Security Risk Assessment Model, in the proceedings of the QoP'07, Alexandria, Virginia, USA, October 29, 2007, pp 31-33.

[28] Ivan Flechais, Cecilia Mascolo and M. Angela Sasse: Integrating security and usability into the requirements and design process, Int. Journal of Electronic Security and Digital Forensics, Vol. 1, No. 1, 2007, pp 12-26.

[29] Haralambos Mouratidis: Secure information systems engineering: a manifesto, Int. Journal of Electronic Security and Digital Forensics, Vol. 1, No. 1, 2007, pp 27-41.

[30] Sun-myung Hwang, Special Checklist for Security Requirements in Software Development Site, in the proceedings of the IEEE International Conference on Multimedia and Ubiquitous Engineering 2007 (MUE'07) IEEE 0-7695-2777-9/07, pp 1172-1176.

[31] Reijo Savola: Requirement Centric Security Evaluation of Software Intensive Systems, in the proceedings of the IEEE 2nd International Conference on Dependability of Computer Systems, 2007, IEEE-0-7695-2850-3/07, pp 135-144.

[32] Betty H. C. Cheng, Joanne M. Atlee: Research Directions in Requirements Engineering, in the proceedings of the IEEE Conference on Future of Software Engineering 2007(FOSE'07), IEEE-0-7695-2829-5/07, pp 285-303.

[33] Charles B. Haley, Jonathan D. Moffett, Robin Laney, Bashar Nuseibeh: A Framework for Security Requirements Engineering, in the proceedings of the SESS'06, Shanghai, China, May 20–21, 2006, pp 35-40.

[34] Guttorm Sindre, E Andreas L. Opdahl: Eliciting security requirements with misuse cases, in the Journal of Requirements Eng, Issue 10, 2005, pp 34–44.

[35] Tim Grance, Joan Hash, Marc Stevens: Security Considerations in the Information System Development Life Cycle, Recommendations of the National Institute of Standards and Technology, Special Publication 800-64 REV. 1, 2006.