Table of Contents

Chapter 1: Introduction ....................................................................................................... 3
  1.1 Integration of Security with Software Engineering ............................................ 3
  1.2 Security Requirement Elicitation ........................................................................ 3
  1.3 Motivation and Problem Statement .................................................................... 4
  1.4 Summary ............................................................................................................. 4

Chapter 2: Thesis Objective and Approach ........................................................................ 5
  3.1 Discussion ................................................................................................................. 5
  3.2 Thesis Statement ....................................................................................................... 5
  3.3 The approach ............................................................................................................. 5
  3.4 The approach benefits: .............................................................................................. 6
  3.4 The approach evaluation ........................................................................................... 7

Chapter 3: Status and Work Plan ........................................................................................ 8

References ........................................................................................................................... 9
Chapter 1: Introduction:

1.1 Integration of Security with Software Engineering:

During the last decade, software systems security has become an increasingly growing concern due to the number of security incidents reported on software security attacks [1]. Attackers exploit software vulnerabilities causing threats to the systems such as stealing sensitive information and causing denial of service. As a result, developing secure software systems has acquired much attention from the research community lately [21].

Secure software development includes integrating security in different phases of the Software development lifecycle (SDLC) such as requirements, design, implementation, and testing. An example for secure software development lifecycle is shown in figure 1. Realizing security early, specifically in requirement phase, is important so that security problems can be tackled soon enough before going further in the process and avoid rework [5]. Thus, several approaches have been proposed that upgrade previous requirement engineering approaches to let it support security such as goal oriented [9], agent oriented [20], and use case based [2]. Such approaches are security requirement engineering approaches.

![Figure 1: An example for secure software development lifecycle [24]](image)

1.2 Security Requirement Elicitation:

In order to integrate security with the requirement engineering, we have to consider security requirements. The basic task of security requirement engineering is to identify and document requirements needed for developing secure software system. Satisfying such *security requirements* should lead to more secure software system [16]. Security requirements can be elicited by analyzing the assets to be protected and the threats from which these assets should be protected [4]. In figure 2, relationships are expressed between assets and services, threats, security requirements and security mechanisms. Assets are vulnerable to threats which necessitate elaborating security requirements. Such security requirements will need security mechanisms which satisfies security requirements because it counters the threats and protects the assets. [4]
1.3 Motivation and Problem Statement

Security requirements need to be explicit, precise, adequate, non-conflicting with other requirements and complete [9]. However, practicing security requirement engineering requires security experts. The analyst should have background on how to identify and analyze the system assets, threats, vulnerabilities and countermeasures. Not all developers have such knowledge and experience in order to deal with security problems. One of the challenges for secure software systems development methodologies is how to enable developers who are not security experts to use them [32]. Thus, the approach used in the security requirement engineering should be effective in eliciting adequate security requirements in addition to assisting non-security experts to use it.

1.4 Summary:

Developing secure software systems has acquired much attention from the research community lately. Security requirement engineering approaches have evolved which aim at realizing security early in the Software development lifecycle. It is used to identify and document requirements which are needed in order to design secure software system. Assets and threats are analyzed in order to elicit security requirements. Security requirements need to be explicit, precise, non-conflicting with other requirements and complete; moreover, practicing security requirement engineering requires security experts. Thus, Security requirement engineering approaches need to assist non security experts to effectively elicit adequate security.
Chapter 2: Thesis Objective and Approach:

3.1 Discussion

Problem frames [12] can help model security requirements effectively, however, we have a gap between security modeling approaches that are based on problem frames. Although such approaches are based on the same model, no integration is presented in the literature that bridges them together. We believe that these approaches complement each other. For example, abuse frames [14] help model threats and thus it can be used to model the informal threat descriptions presented in [17]. Security problem frames [18] present helpful catalog for modeling security problems; however, the analyst can make better use of such patterns catalog if it adapts abuse frames [14] to help relate threats with corresponding security requirements.

3.2 Thesis Statement:

*Develop a methodology that assists software developers in eliciting adequate security requirements during the requirement engineering phase that is based on problem frames, abuse frames and security problem frames.*

3.3 The approach:

The approach intends to adapt security to a requirement engineering methodology that is based on problem frames. Abuse frames and security problem frames will be utilized for this purpose.

The approach proposed is built upon works of [14], [17], and [18]. It iterates through the following steps:

1- Model the system context and functional requirements of the software system using problem frames.
2- Identify the system assets which can be represented as domains in problem frames as in [17].
3- Determine threats by using abuse frames [14] to model threat descriptions presented in [17]. Threats will be searched from a catalog that incorporates both abuse frames [14] and security problem frames [18].
4- If a search hit is found, instantiate the matched abuse frame and the security problem frame to fit the related subproblem. Instantiation will include modifying phenomena and interfaces of the security problem frame in order to map to the specific subproblem.
5- If there are no hits, Crosscut threats with the functional requirements to identify vulnerabilities. If we have vulnerabilities, we will derive the security requirement by constraining the functional requirement of the subproblem or by adding trust assumptions leading to counter the vulnerability as in [17].

The approach is represented by an activity diagram shown in figure 3:
3.3 The approach benefits:
Such approach utilizes different methodologies based on problem frames that complement each other in order to model security requirements. It intends to assist the analyst during security requirement elicitation through the requirement engineering phase. We believe that such integration would achieve consistency in models as all the models used are problem frames based. Our contribution will be in interfacing security problem frames with Abuse Frames and adapting them into requirement engineering methodology based on Problem Frames. The approach will be further refined in order to better integrate such models together.

Figure 3: Activity diagram for the proposed approach
3.4 The approach evaluation

The approach will be evaluated by showing how it can help elicit security requirements for case studies that use problem frames in modeling software systems. For example, we can make use of the case study presented in [33] where bank transactions are modeled using problem frames. Also, we can compare with the case study presented in [17] where human resource system is modeled. Such models will be inputted into step 2 of our approach till the end. After progressing through the approach steps, adequate security requirements for the system are expected to be elicited.
Chapter 3: Status and Work Plan:

We are currently applying the methodology on case study presented in [17] to make initial validation to our methodology. Our future work plan will start by refining the initial methodology we are proposing. Then, we will tackle problems of integrating models such as abuse frames and security problem frames into the proposed methodology. Finally, we will apply the methodology on case studies to verify its applicability in capturing security requirements. Table 1 shows a timeline for the thesis proposal.

Table 1: Thesis Timeline

| Fall 08                  | Spring 09             |
|-------------------------|-----------------------|
| Literature survey and initial methodology development | November 08 – January 09 |
| Refining methodology    | February – April 09   |
| Adapting models into methodology |                  |
| Apply methodology on suitable case studies |                  |
| Thesis Writing          |                      |
References:

[1] CERT, http://www.cert.org/stats/cert_stats.html

[2] G. Sindre and A.L. Opdahl, “Eliciting security requirements with misuse cases”, Requirements Engineering 10, Springer-Verlag London Ltd, 2005

[3] Jon Whittle, Duminda Wijesekera, Mark Hartong, Executable misuse cases for modeling security concerns, Proceedings of the 30th international conference on Software engineering, Leipzig, Germany, 2008

[4] D. Firesmith. “Security Use Cases”. Journal of Object Technology, Vol. 2, No. 3, 53-64, 2003.

[5] Sindre, G., Firesmith, D.G. and Opdahl, A.L., “A Reuse-Based Approach to Determining Security Requirements.” In: Proc. 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ’03), 2003

[6] F. Swiderski and W. Snyder. Threat Modeling. Microsoft Press, 2004.

[7] Suvda Myagmar, Adam J. Lee, and William Yurcik, “Threat Modeling as a Basis for Security Requirements”, IEEE Symposium on Requirements Engineering for Information Security (SREIS’05), 2005

[8] E. Oladimeji, S. Supakkul and L. Chung, "Security Threat Modeling: A Goal-Oriented Approach,” Proc. SEA’06, Dallas, TX, 2006

[9] A. van Lamsweerde, “Elaborating Security Requirements by Construction of Intentional Anti-models,” Proc. 26th Int’l Conf. Software Eng. (ICSE 04), IEEE CS Press, 2004

[10] A. van Lamsweerde and E. Letier, “Handling Obstacles in Goal-Oriented Requirements Engineering”, IEEE Transactions on Software Engineering, Special Issue on Exception Handling, 2000

[11] Laurent A. Hermoye and Axel van Lamsweerde, and Dewayne E. Perry. "A Reuse-Based Approach to Security Requirements Engineering", 2006

[12] M. Jackson. Problem Frames. “Analyzing and structuring software development problems.” Addison-Wesley, 2001.

[13] M. Jackson. Problem Frames, “Problem Frames and Software Engineering,” Expert Systems, Volume 25, Number 1, pp. 7-8(2), 2008

[14] Lin, L., Nuseibeh, B., Ince, D., Jackson, M., & Moffett, J. "Introducing Abuse Frames for Analyzing Security Requirements," In Proceedings of the 11th IEEE International Requirements Engineering Conference (RE’03). Monterey CA USA, pp. 371-372, 2003
[15] C.B. Haley, J.D. Moffett, R. Laney, and B. Nuseibeh, “A Framework for Security Requirements Engineering,” Proc. 2006 Software Eng. for Secure Systems Workshop with the 28th Int’l Conf. Software Eng., 2006.

[16] C.B. Haley, J.D. Moffett, R. Laney, and B. Nuseibeh, “Security Requirements Engineering: A Framework for Representation and Analysis”, IEEE Transactions on Software Engineering, 2008.

[17] C.B. Haley, R. Laney, and B. Nuseibeh “Deriving Security Requirements From Crosscutting Threat Descriptions,” in Proceedings of the Third International Conference on Aspect-Oriented Software Development, Lancaster, UK, March 22–26, 2004

[18] D. Hatebur, M. Heisel, and H. Schmidt, “A pattern system for security requirements engineering,” in Proceedings of the International Conference on Availability, Reliability and Security (ARES), pp.356-365, IEEE, 2007.

[19] P. Giorgini, F. Massacci, J. Mylopoulos, N. Zannone, Requirements Engineering meets Trust Management: Model, Methodology, and Reasoning, in: Proc. of iTrust-04, LNCS 2995, 2004.

[20] L. Liu, E. Yu, and J. Mylopoulos. Security and privacy requirements analysis within a social setting. In Proc. of RE’03, pages 151–161, 2003.

[21] K.R.V. Wyk and G. Mcgraw, "Bridging the Gap between Software Development and Information Security", presented at IEEE Security Privacy, pp.75-79, 2005

[22] L. Chung, B. Nixon, E. Yu, and J. Mylopoulos, Non-Functional Requirements in Software Engineering. Kluwer Academic, 2000.

[23] Inger Anne Tøndel, Martin Gilje Jaatun, Per Hakon Meland, “Security Requirements for the Rest of Us: A Survey”, IEEE Software, vol.25, no.1, pp.20-27, 2008

[24] A. Apvrille and M. Pourzandi, "Secure Software Development by Example," IEEE Security &Privacy, vol. 3, no. 4, pp. 10–17, 2005

[25] J. Pauli and D. Xu, "Integrating Functional and Security Requirements with Use Case Decomposition", Proc. of the 11th International Conference on Engineering of Complex Computer Systems (ICECCS 2006). Palo Alto, CA, USA, 2006

[26] S. Ardi, David Byres, P. H. Meland, I.A. Tondel, N. Shahmehri “How Can the Developer Benefit From Security Modeling”, In Second International Conference on Availability, Reliability and Security (ARES’07), IEEE, 2007.

[27] Common Criteria for Information Technology Security Evaluation v2.3, ISO/IEC 15408: August, 2005

[28] Bruce Schneier, “Attack Trees,” Dr. Dobb's Journal December 1999
[29] Information Assurance Technology Analysis Center (IATAC)/ Data and Analysis Center for Software “Software Security Assurance: A State-of-the-Art Report,”, 2007

[30] Meledath Damodaran, “Secure Software Development Using Use Cases and Misuse Cases,” Issues in Information Systems, No. 1 : pp. 150–154, 2006

[31] N. Yoshioka, H. Washizaki, and K. Maruyama, "A survey on security patterns", Progress in Informatics, No. 5, pp. 35-47, March 2008

[32] H. Mouratidis “Secure Information Systems Engineering: A Manifesto” International Journal of Electronic Security and Digital Forensics 1(1) pp. 27-41, 2007

[33] V. Banagala, “Analysis of Transaction Problems Using the Problem Frames Approach”, in Proc. of ICSE 2006, China, pp. 5-12, 2006