Threat and Risk-Driven Security Requirements Engineering

Holger Schmidt, Technical University of Dortmund, Germany

ABSTRACT

In this paper, the author aims to present a threat and risk-driven methodology to security requirements engineering. The chosen approach has a strong focus on gathering, modeling, and analyzing the environment in which a secure ICT-system to be built is located. The knowledge about the environment comprises threat and risk models. As presented in the paper, this security-relevant knowledge is used to assess the adequacy of security mechanisms, which are then selected to establish security requirements.

Keywords: Domain Knowledge, Risk Analysis, Secure Software Engineering, Security Requirements Engineering, Threat Analysis

INTRODUCTION

Security describes the inability of the environment to have an undesirable effect on an ICT-system (Røstad, Tøndel, Line, & Nordland, 2006). Expressed from the software development point of view, the software of an ICT-system must be constructed in a way such that the ICT-system is protected from the environment.

From the definition of the term security alone, the importance of the environment in which an ICT-system is integrated when we deal with security becomes clear. The adequacy of an ICT-system’s security mechanisms is strongly influenced by its intended operational environment. Thus, knowledge about the environment must be collected to support or even allow the decision for particular security mechanisms. Especially knowledge about the malicious part of the environment, i.e., the part that can attack an ICT-system in order to nullify or bypass its security mechanisms, must be gathered, modeled, and assessed.

The gathering and modeling parts involve threat analysis techniques such as Microsoft’s STRIDE (Hernan, Lambert, Ostwald, & Shostack, 2006), attack trees by Schneier (1999), and the approach by Fernandez, Red M., Forneron, Uribe, and Rodriguez G. (2007). Threat analysis is applied to identify threats that exploit vulnerabilities of security mechanisms. The assessment part is related to risk analysis techniques such as CORAS by Braber, Hogganvik, Lund, Stølen, and Vraalsen (2007). Risk analysis is used to determine the probability that security mechanisms will work correctly in the intended (malicious) environment.

When constructing secure ICT-systems, it is instrumental to take the environment into account right from the beginning of the software development. Consequently, we focus in this paper on early security requirements engineering.
We consider our security engineering process using patterns (SEPP) (Hatebur, Heisel, & Schmidt, 2007; Schmidt, 2010). There, security requirements are elicited, analyzed, and documented so that security mechanisms adequate to establish the security requirements can be selected. SEPP makes use of special patterns called security problem frames for security requirements and concretized security problem frames for security mechanisms. Thus, security requirements are strictly separated from solutions, i.e., security mechanisms.

Results from applying threat and risk analysis techniques heavily influence the process of selecting security mechanisms to establish security requirements. In this paper, we analyze the impact of threat and risk analysis on security requirement engineering. We extend SEPP by a threat and risk-driven procedure to select adequate security mechanisms. We illustrate the procedure using an example software development.

The rest of the paper is organized as follows: In the next section, we introduce a concrete security-critical software development problem. We use this example then to demonstrate security problem frames and concretized security problem frames. Following this, we present the threat and risk analysis extensions to SEPP’s requirements analysis phase. Afterwards, we validate this threat and risk-driven approach. Finally, we consider related work followed by a summary and directions for future research.

**CASE STUDY**

We use the following software development problem as a case study to demonstrate the techniques presented in this paper.

A secure text editor should be developed. The text editor should enable an author to create, edit, open, and save text files. The text files should be stored confidentially.

The informal security requirement (SR1) can be described as follows:

Preserve confidentiality of text file except for its file length for honest environment and prevent disclosure to malicious environment.

Note: We decide to focus on storing text files confidentially. The given software development problem can also be interpreted such that the security requirement also covers confidential editing operations, e.g., confidential clipboard copies. To simplify matters, this is not covered in the security requirements analysis presented in this paper. For the same reason, the create and edit functionality of the secure text editor is not covered in our case study. Practically, it is very difficult to develop 100% confidential systems. Hence, as an example, we discuss an SR that allows the secure text editor to leak the text file length.

**Problem Frames for Security Requirements Engineering**

In earlier publications (cf., Hatebur, Heisel, & Schmidt, 2008; Schmidt, 2010), we presented special patterns defined for structuring, characterizing, and analyzing problems that occur frequently in security engineering. Similar patterns for functional requirements have been proposed by Jackson (2001). They are called problem frames. Accordingly, our patterns are named security problem frames (SPF). SPF consider security requirements. Furthermore, for each SPF, we defined a set of concretized security problem frames (CSPF) that take generic security mechanisms (e.g., encryption to keep data confidential) into account to prepare the ground for solving a given security problem. Since CSPFs involve first solution approaches, they consider concretized security requirements. Problem frames, SPF and CSPFs are graphically depicted by frame diagrams. Figure 1 shows an instance of the frame diagram of the CSPF confidential data storage using password-based symmetric encryption. Instances of frame diagrams are called problem diagrams.
Client-Server Based LBS Architecture: A Novel Positioning Module for Improved Positioning Performance
[www.igi-global.com/article/client-server-based-lbs-architecture/46084?camid=4v1a](www.igi-global.com/article/client-server-based-lbs-architecture/46084?camid=4v1a)

Securing EPR Data Using Cryptography and Image Watermarking
[www.igi-global.com/article/securing-epr-data-using-cryptography/66368?camid=4v1a](www.igi-global.com/article/securing-epr-data-using-cryptography/66368?camid=4v1a)