Modeling Requirements for Security-enhanced Design of Embedded Systems

Alberto Ferrante, Igor Kaitovic and Jelena Milosevic
ALaRI, Faculty of Informatics, Università della Svizzera italiana
Via G. Buffi 13, 6904, Lugano, Switzerland

Keywords: Modeling, UML Profiles, Security Requirements, Embedded Systems Design.

Abstract: Designing an embedded system is a complex process that involves working on both hardware and software. The first step in the design process is defining functional and non-functional requirements; among them, it is fundamental to also consider security.
We propose an effective way for designers to specify security requirements starting from User Security Requirements. User Security Requirements are high-level requirements related to security attacks that the system should be able to withstand. We also provide a mechanism to automatically translate these User Requirements into System Security Requirements, that include a detailed description of security solutions. For expressing requirements we use Unified Modeling Language (UML); specifically, we create a UML profile to describe user requirements and we use model-to-model transformation to automatically generate system requirements. We show the effectiveness of the modeling scheme and of the translation mechanism by applying our methodology to a case study based on wearable devices for e-health monitoring.

1 INTRODUCTION

The use of embedded systems has been continuously increasing in the last years both with respect to the number of application areas and to the complexity of platforms designed: embedded systems have become fundamental in ICT. This increase in their use, even in new, critical, and non-protected environments, has not been accompanied by a proper development of methodologies for making them secure (Viega and Thompson, 2012). Additionally, security for these systems proved to be more difficult and more critical than security for general purpose systems. Criticality is given both by the kinds of data stored in mobile devices (e.g., phone bookmarks and personal user data) and by the kind of damages that an attack to a mobile system may produce. However, security is not yet perceived as a significant problem by embedded systems users.

Any design starts with requirement specifications. For each embedded system, there exist functional requirements and non-functional requirements, such as the ones on cost, performance, time, and area. In the aim of designing secure embedded systems, it is fundamental to add security to these non-functional requirements. In this way, in fact, security can be not only included along with the functional system design, thus providing better protection, but it can also be optimized along with it. Security specification is a first step toward a comprehensive approach to the design of secure embedded systems. This step, as also mentioned in the ISO/IEC 15408-3 (ISO/IEC, 2009) standard, is fundamental to limit the introduction of vulnerabilities in the system. Security requirements can be specified by considering two main stand points: the one of security solutions and the one of attacks that the system should be able to withstand (i.e., the impact of these attacks on the system should be mitigated or, in the best case, completely prevented). The former should be chosen by considering the latter and a trade-off among impact of each attack and cost of a suitable security solutions.

In the rest of the paper we refer to the requirements related to security attacks as *User (Security) Requirements*; we refer to solution-level requirements as *System (Security) Requirements*. During system design, User Requirements need to be translated into system ones. System Requirements, in turn, are implemented. The possibility to express security requirements as related to security attacks that the system should be able to withstand, would give system designers the possibility to define security requirements easily, by starting from security standards, penetration tests, and security assessment methods, such

Ferrante A., Kaitovic I. and Milosevic J..
Modeling Requirements for Security-enhanced Design of Embedded Systems.
DOI: 10.5220/0005050003150320
In *Proceedings of the 11th International Conference on Security and Cryptography* (SECRYPT-2014), pages 315-320
ISBN: 978-989-758-045-1

as the ones discussed in (Köster et al., 2009). Therefore, a translation mechanism from user to System Requirements is needed as well as a translation from System Requirements to a choice of suitable implementations.

In this paper, we propose a suitable format, based on UML, for specifying security requirements; we also propose a methodology for translating the User Security Requirements into System Security Requirements. System Security Requirements obtained in such a way can be integrated directly into the embedded systems design methodology described in (Ferrante et al., 2013), where requirements are translated into set of implementations and optimized by means of design space exploration.

The remaining part of this paper is organized as follows: Section 2 discusses the related work; Section 3 describes in detail the methodology that we have devised, the UML profile that we propose to use, and the automatic translation process; Section 4 discusses the application of the methodology on a case study related to wearable devices for e-health applications; Section 5 provides a summary of the obtained results and outlines future work.

2 RELATED WORK

In (Kocher et al., 2004) it is suggested that design methodologies for secure embedded systems should include techniques for specifying security requirements in a way that can be easily communicated to the design team and evaluated throughout the design cycle; any attempt to specify security requirements needs to address the desired *level of security*. Security is currently specified by system architects in a vague and imprecise manner (Kocher et al., 2004).

During embedded system architecture design, techniques to map security requirements to different alternative solutions and to explore the associated trade-offs in terms of cost, performance, and power consumption, would be invaluable in helping embedded system architects understand and make better design choices (Kocher et al., 2004). Various approaches that incorporate security from the early stage of system design are based on Unified Modeling Language (UML) (Object Management Group, 2011) as it is standardized as well as the most commonly used modeling language in industry and academia. UML can be easily extended and tailored for a specific domain through the concept of profiles. A profile is a set of tags and stereotypes that can be applied to standard UML notation as additive extensions. For example a stereotype can be defined to extend the notation of a UML class in order to model a specific type of a hardware component. In (Jürjens, 2003) a language named UMLsec is proposed as an extension of UML for modeling security requirements. A number of other UML profiles have been defined for various fields including security, such as (Bouaziz and Coulette, 2012; Rodriguez et al., 2006). In (Ferrante et al., 2013) and in (Roudier et al., 2013), embedded systems design methodologies that include security in the optimization process, are described. In the former, security requirements are defined in UML by following a given template and in the form of required security solutions with the associated parameters; in the latter, SysML-Sec, an extension of SysML, is proposed and used to model requirements.

There exist languages for specifying security requirements, such as i* (Yu, 1997) and SI* (Massacci et al., 2010). These two languages are designed to describe security requirements in information systems, where multiple actors are involved. Other works, such as (Markose et al., 2008), propose languages and methods for modeling security threats in embedded systems.

The goal of our work is to define a framework that can be used to specify security requirements as well as for automatically or semi-automatically translating security requirements from the *security attacks* level to the *security solutions* ones. i* and SI* can be used to specify the security requirements of the environment in which the considered embedded system will be used and, therefore, to derive the security requirements of the system that will, in turn, be specified by using our language. UMLsec, SysML-Sec and the other existing UML profiles, are more in line with our goals, even though, they do not distinguish between different level of security requirements nor provide mechanisms for relating and translating User to System Security Requirements. Our profile overcomes this gap and can be used along with the existing UML profiles for modeling security.

3 METHODOLOGY

The goal of our approach is to simplify modeling of security-related requirements. Therefore, we devised a framework in which a separation of concerns between system designers and security experts is considered: security experts define and keep updated security libraries for different application fields (e.g., mobile devices) and system designers (assumed not to be a security expert) rely on these to model security requirements of the system. More specifically, each security library contains a list of possible User Re-

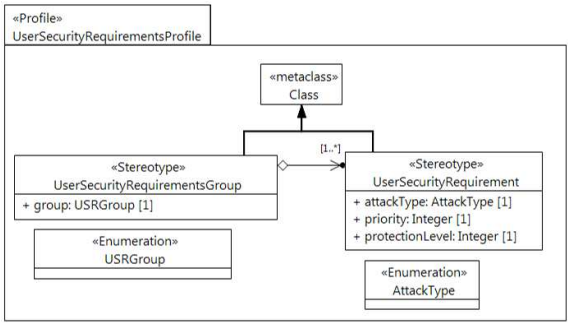

Figure 1: User Security Requirements profile.

quirements and implied System Requirements. System designer choses a subset of User Requirements and assigns desired priorities and security levels. The methodology is composed of:

- **Security Requirements Framework** - a framework for specifying and modeling security requirements and relating User and System Requirements.
- **Security Library** - a library defined by the security expert using the formalism provided by the framework; it contains a set of User and System Requirements and appropriate mappings.
- **Automated Requirements Translation** - a process that, for a subset of User Security Requirements chosen by the designer, generates alternative sets of System Security Requirements in an automated fashion.

In scope of this work we fully define and implement the framework and the mapping process. The security library should be set for each specific domain by security experts.

The framework is implemented in Papyrus (The Eclipse Foundation, 2013b), a widely accepted UML editor for Eclipse (The Eclipse Foundation, 2013a). A model-to-model transformation mechanism has been used to implement the translation of User Security Requirements into the System ones. The implementation has been done by using VIsual Automated model TRAnsformations (VIATRA) framework (The Eclipse Foundation, 2013c). Even though the current framework could be designed will less advanced engine than VIATRA, we chose it in order to support further extensions and enhancements of our methodology.

3.1 Security Requirements Framework

The security requirements framework is implemented as a UML profile divided into three sub-profiles: User Security Requirements Profile, System Security Requirements Profile, Security Requirement Mapping Profile. The User Security Requirements Profile, depicted in Figure 1, extends the notion of a UML class to model User Security Requirements with following attributes: Attack type, Priority, Protection level.

Attack types are defined and enumerated as a part of the Security Library definition to describe various types of attacks (e.g., Impersonation and Eavesdropping). Security requirement priority defines the importance of a requirement. This property is used when only a subset of requirements can be implemented due to mutual exclusions or resources constrains. Protection level is defined by the security expert and might have different interpretations for different attacks. User Requirements groups combine a number of related User Security Requirements. A User Requirement can belong to any number of groups while a group (e.g., Network related attacks) must contain at least one requirement.

System Security Requirements profile depicted in Figure 2 defines a set of grouped and related stereotypes for modeling various types of System Security Requirements. The profile is defined in a generic manner so that various security domains can be covered. These requirements can be further related to solutions that are more specific in the same way as it is done in (Ferrante et al., 2013).

Finally, Security Requirements mapping profile depicted in Figure 3 provides a mean to relate User and System Requirements. A User Requirement may imply one or more System Requirements (i.e., multiple security solutions may be needed to withstand certain attacks). Thus, a User Requirement is related to a group of System Requirements that includes a set of complementary System Requirements. Multiple User Requirements may be mapped to the same (or similar) System Requirement so that one System Requirement is included in several groups. Finally, alternative system requirements might exist for some User Requirements. Thus, a User Requirement can be related to multiple (but at least one) alternative groups of System Requirements.

3.2 Security Library

For a specific domain, a Security library contains a list of User Requirements and implied System Requirements. User Requirements are divided into groups for the convenience of designers; a group contains User Requirements that have similar characteristics (e.g., all attacks related to the network connection). Each User requirement is related to multiple groups of System Requirements.

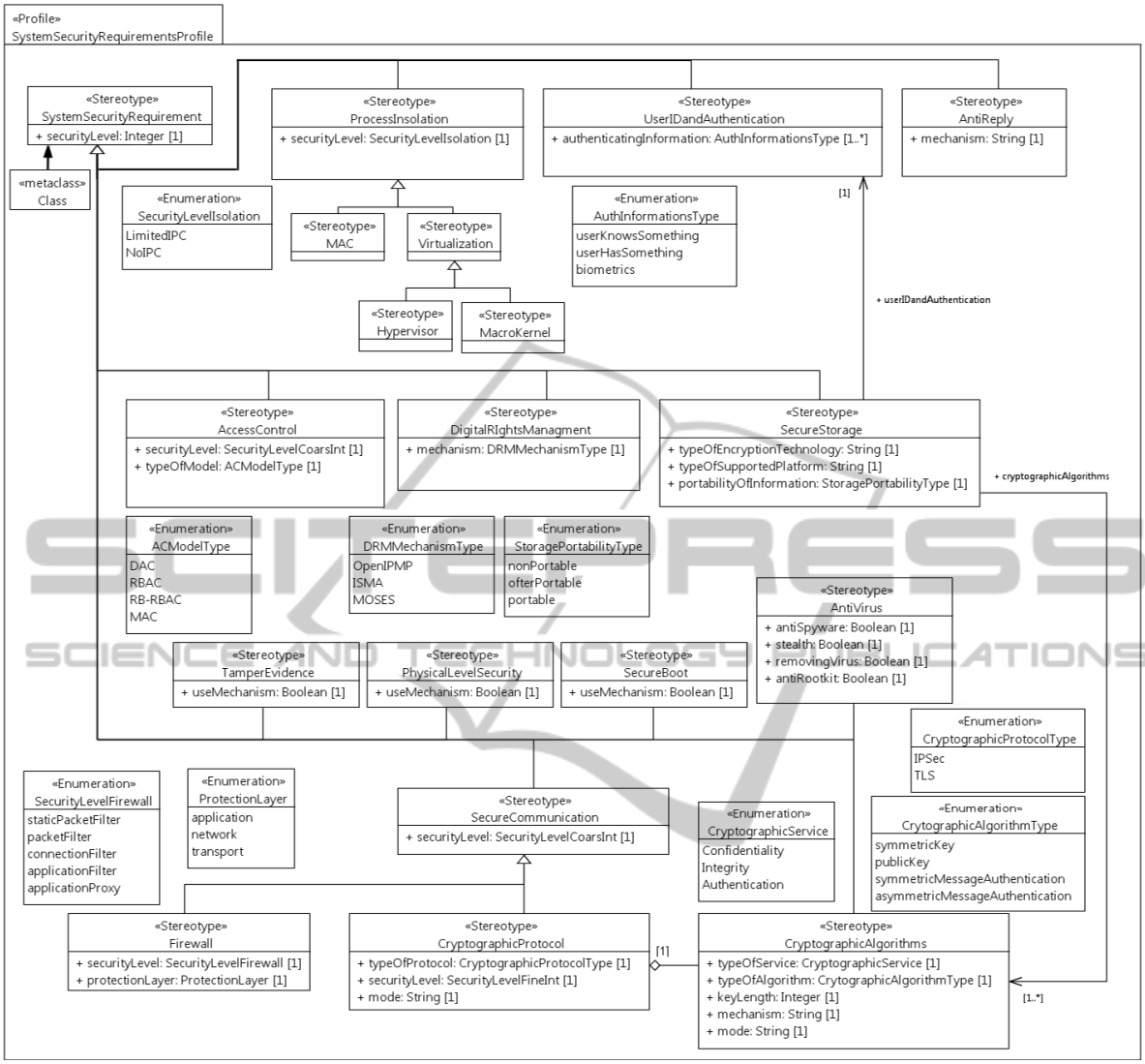

Figure 2: System Security Requirements profile.

3.3 Automated Requirements Translation

A procedure for translating User into System Requirements generates a set of UML diagrams describing alternative sets of System Security Requirements taking a set of User Requirements and a Security library as inputs. A system designer choses a set of User Requirements and describe their properties in an XML format. Starting from an empty set of System Requirements, the translation engine runs a loop in which it (1) takes one User Requirement at a time, (2) identifies implied System Requirement groups and (3) merges a set of System Requirements from the previous iteration considering all possible alternative solutions eith these System Requirement groups.

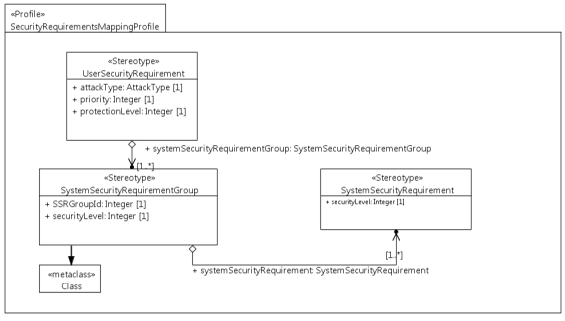

Figure 3: Mapping profile.

4 CASE STUDY

As a case study we consider wearable wireless devices used in Body Area Networks (BAN) for e-health

Table 1: Textual representation of the Security Library used in the case study. In the table, User Requirements are Mapped to System Requirements; System Requirements shown in different table rows represent alternative mappings for the corresponding User Requirement.

Group	User Requirements	System Requirements
Network-related	Eavesdropping of network connection	Symmetric encryption: AES, 128 bit key Encryption key exchange: pre-shared key
		Symmetric encryption: AES, 128 bit key Encryption key exchange: DSA
		Symmetric encryption: Trivium, 80 bit key Encryption key exchange: pre-shared key
	Data modification through man-in-the-middle	Data authentication: HMAC-SHA256 Authentication key exchange: DSA Host authentication: Public-key algorithm, 1024-bit key
		Data authentication: Poly1305-AES Encryption key exchange: Pre-shared key Host authentication: Trivium, 80-bit key Host authentication key exchange: pre-shared
	Impersonation	Host authentication: public-key algorithm, 1024 bit key Anti-replay: counter
		Host authentication: Trivium, 80-bit key Host authentication key exchange: pre-shared Anti-replay: counter
Local	Data stealing through physical access to the device	Local storage encryption: AES, 128 bit key
		Local storage encryption: Trivium, 80 bit key

```
<?xml version="1.0" encoding="UTF-8"?>
<user_requirement id="user_requirement_1">
    <attack_type id="Eavesdropping of network connection"></attack_type>
    <priority val="1"></priority>
    <security_level level="1"></security_level>
</user_requirement>
<user_requirement id="user_requirement_2">
    <attack_type id="Data modification through man-in-the-middle"></attack_type>
    <priority val="1"></priority>
    <security_level level="2"></security_level>
</user_requirement>
<user_requirement id="user_requirement_3">
    <attack_type id="Impersonation"></attack_type>
    <priority val="2"></priority>
    <security_level level="1"></security_level>
</user_requirement>
```

Figure 4: XML code representing the User Requirements of the case study.

monitoring. Security issues in BANs for telemedicine and e-health are particularly important because sensitive medical information must be protected from unauthorized use for personal advantage and fraudulent acts that might be hazardous to a user's life (e.g., alteration of system settings, drug dosages, or treatment procedures) (Poon et al., 2006). Although they carry sensible information, these devices are usually quite limited on computational resources and on power. Moreover, they are often connected to remote locations by means of a local gateway, that, in most of the cases, is a smartphone. Specific security solutions are available for this kind of devices, such as the ones listed in (Di Pietro and Mancini, 2003).

We consider the security library containing the requirements shown in Table 1. In (Kargl et al., 2008; Ameen et al., 2012; Poon et al., 2006) a number of attack types that could happen in e-health monitoring systems are given; among them, the ones that are used in this case study are shown in column User Requirements. As an example of the library, a part of it related to the *Eavesdropping of network connection* is represented in Figure 5.

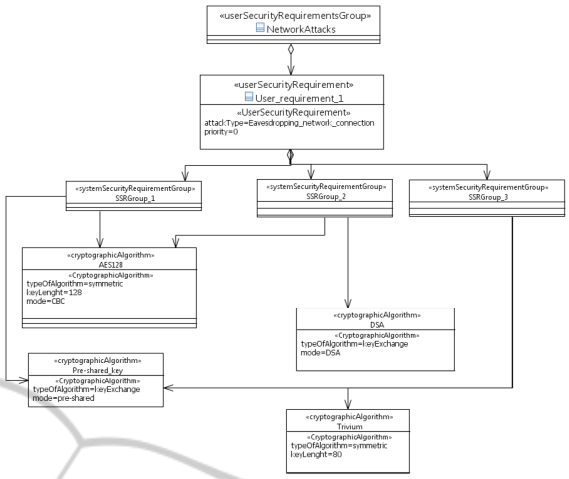

Figure 5: Part of the Security Library representing System Requirements related to the *Eavesdropping of network connection* User Requirement.

The User Requirements are represented in the XML code shown in Figure 4. Both User Requirements no. 1 and no. 4 are related to eavesdropping of network connection and, therefore, they are represented by *user_requirement_1*; User Requirement no. 2 and 3 are represented by *user_requirement_2* and *user_requirement_3*, respectively. We assigned priorities to the requirements in the XML file in such a way that eavesdropping and data modification are more important than impersonation. Furthermore, we used the optional security level field to specify that we require lowest possible security for *user_requirement_1* and *user_requirement_2*, while we require medium security for *user_requirement_3*.

The result of the translation process is a set of UML class diagrams, each of them representing an alternative set of System Requirements. Each set is defined as a combination of System Requirement groups and included System Requirements where each group is related to one of the User Requirements.

5 CONCLUSIONS AND FUTURE WORK

We presented a methodology to: (1) specify user and system security requirements, (2) create a security library by relating the two, and (3) automatically translating user security requirements to system security requirements. User requirements are specified by considering the attacks that the system should be able to withstand; the system requirements are defined considering the protection mechanisms that can be applied against these attacks. The implementation is done by defining a set of UML profiles and apply-

ing model-to-model transformations.

Future work will be on optimizing the translation mechanism in such a way that similar System Requirements derived from different User Requirements are recognized and merged.

REFERENCES

Ameen, M., Liu, J., and Kwak, K. (2012). Security and privacy issues in wireless sensor networks for healthcare applications. *J. Med. Syst.*, 36(1):93–101.

Bouaziz, R. and Coulette, B. (2012). Applying security patterns for component based applications using uml profile. In *Computational Science and Engineering (CSE), 2012 IEEE 15th International Conference on*, pages 186–193.

Di Pietro, R. and Mancini, L. V. (2003). Security and privacy issues of handheld and wearable wireless devices. *Commun. ACM*, 46(9):74–79.

Ferrante, A., Milosevic, J., and Janjusevic, M. (2013). A security-enhanced design methodology for embedded systems. In *ICETE SECRYPT 2013*, Reykjavik, Iceland. ICETE.

ISO/IEC (2009). *ISO/IEC 15408-3 – Evaluation criteria for IT security – Part 3: Security assurance components*.

Jürjens, J. (2003). *Secure Systems Development with UML*. Springer Verlag.

Kargl, F., Lawrence, E., Fischer, M., and Lim, Y. Y. (2008). Security, privacy and legal issues in pervasive ehealth monitoring systems. In *Mobile Business, 2008. ICMB '08. 7th International Conference on*, pages 296–304.

Kocher, P., Lee, R., McGraw, G., and Raghunathan, A. (2004). Security as a new dimension in embedded system design. In *Proceedings of the 41st annual Design Automation Conference*, DAC '04, pages 753–760, New York, NY, USA. ACM. Moderator-Ravi, Srivaths.

Köster, F., Nguyen, H., Obermeier, S., Brändle, M., Klaas, M., Naedele, M., and Brenner, W. (2009). Information security assessments for embedded systems development: An evaluation of methods. In *8th Annual Security Conference*, Las Vegas, USA.

Markose, S., Liu, X., and McMillin, B. (2008). A systematic framework for structured object-oriented security requirements analysis in embedded systems. In *Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on*, volume 1, pages 75–81.

Massacci, F., Mylopoulos, J., and Zannone, N. (2010). Security requirements engineering: The si* modeling language and the secure tropos methodology. In Ras, Z. and Tsay, L.-S., editors, *Advances in Intelligent Information Systems*, volume 265 of *Studies in Computational Intelligence*, pages 147–174. Springer Berlin Heidelberg.

Object Management Group (2011). *Unified Modeling Language Infrastructure*, 2.4.1 edition. http://www.omg.org/spec/UML/2.4.1.

Poon, C. C. Y., Zhang, Y.-T., and Bao, S.-D. (2006). A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health. *Communications Magazine, IEEE*, 44(4):73–81.

Rodriguez, A., Fernandez-Medina, E., and Piattini, M. (2006). Security requirement with a uml 2.0 profile. In *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*.

Roudier, Y., Idrees, M. S., and Apvrille, L. (2013). Towards the model-driven engineering of security requirements for embedded systems. In *MODRE 2013, International Workshop on Model-Driven Requirements Engineering, 15 July 2013, Rio de Janeiro, Brazil*, Rio de Janeiro, BRAZIL.

The Eclipse Foundation (2013a). Eclipse development environment. http://www.eclipse.org.

The Eclipse Foundation (2013b). Papyrus UML modeling tool. http://www.eclipse.org/modeling/mdt/papyrus.

The Eclipse Foundation (2013c). VIATRA2, VIsual Automated model TRAnsformations. http://www.eclipse.org/gmt/VIATRA2/.

Viega, J. and Thompson, H. (2012). The state of embedded-device security (spoiler alert: It's bad). *Security Privacy, IEEE*, 10(5):68 –70.

Yu, E. (1997). Towards modelling and reasoning support for early-phase requirements engineering. In *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*, pages 226–235.