Security engineering for embedded systems – the SecFutur vision

[Vision Paper]

Sigrid Gürgens
Fraunhofer SIT
Darmstadt, Germany
guergens@sit.fraunhofer.de

Carsten Rudolph
Fraunhofer SIT
Darmstadt, Germany
rudolph@sit.fraunhofer.de

Antonio Maña
University Malaga, Spain
amg@lcc.uma.es

Simin Nadjm-Tehrani
Linköping University, Sweden
simin.nadjm-tehrani@liu.se

ABSTRACT

Security is usually not in the main focus in the development of embedded systems. However, strongly interconnected embedded systems play vital roles in many everyday processes and also in industry and critical infrastructures. Therefore, security engineering for embedded systems is a discipline that currently attracts more interest. This paper presents the vision of security engineering for embedded systems formulated by the FP7 project SecFutur [1].

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications—*Methodologies, tools*; C.3 [**Computer Systems Organization**]: Special-purpose and application-based systems—*Real-time and embedded systems*

General Terms

Design, Security

Keywords

Security engineering, embedded systems

1. INTRODUCTION

This paper describes a long-term vision on security engineering for embedded systems. Parts of this vision will be realized in the European research project SecFutur, but the scope of the discipline of security engineering is much wider than what can be achieved within one research project. Therefore, we publish this vision early in the project and ask researchers in the field of embedded systems as well as software and security engineering to interact with SecFutur in order to make strong and efficient security solutions available to the developers of complex embedded systems in a way that higher assurance can be achieved and specific security requirements are met.

S&D4RCES 2010 September 14, 2010, Vienna, Austria.

2. MOTIVATION

The focus of system engineering for embedded systems usually lies on resource-efficiency, cost reduction and functionality. Non-functional properties, if considered in the development of embedded systems, are mainly concerned with safety or performance. Security issues are either neglected, added as an afterthought or minimized due to cost or efficiency conditions in the development. One important reason for this situation is that in the past many embedded systems only had very restricted connectivity and operated in controlled environments. Thus, security was usually sufficiently provided by physical protection of the embedded devices. This situation has dramatically changed. Evolution of embedded systems towards devices connected via Internet, wireless communication or other interfaces as well as the trend towards always growing numbers of devices (Internet of Things) requires a re-consideration of embedded systems engineering processes. It is no longer possible to achieve the required level of security by adding security measures late in the development process. Security engineering needs to be part of the development in all stages of the process.

From a security point of view embedded devices are basically systems owned by a certain entity and operated in a potentially hostile environment. Hence requirements that could be assumed to be met by systems using physically protected devices need to be revisited. Possible examples can include non-repudiation of data, time or status of devices controlling sensor networks, legal requirements for calibration and gauging devices, security of payment systems or the enforcement of a particular behaviour of remote-controlled devices. A variety of specific international and national regulations and standards are relevant for such devices. Many of them require protection against manipulations.

Currently, a security engineering process for embedded systems that takes these considerations into account does not exist. However, a lot of building blocks are already available to be included in such a process. These building blocks

mainly provide efficient cryptographic algorithms, hardware security anchors or protection against some side-channel attacks. Many of them work on standard hardware and do not require additional expensive extensions or excessive resources. However, most existing building blocks only provide very basic security mechanisms for specific embedded use cases. The integration of these mechanisms in order to satisfy more complex security requirements is not trivial, and errors in applying or combining solutions can result in a low overall security of the system.

3. OBJECTIVES OF A NEW SECURITY ENGINEERING PROCESS

The main objective of security engineering shall be to support the development of dependable and secure embedded systems with predictable properties. The SecFutur project will develop a new security engineering process that can flexibly integrate security solutions into an overall framework for the hardware platform based design process and that can be incorporated into the existing design process with a minimum amount of changes. Security solutions will be provided in terms of security building blocks that integrate existing hardware and software security mechanisms in order to address complex security requirements as mentioned in the previous section. Figure 1 visualizes the main requirements for a security engineering process. The following list highlights four main goals a security engineering process should target and summarizes the vision of the SecFutur project with respect to these four statements:

Goal 1: Adequate support for embedded system developers to make *informed security design decisions* in the development process of embedded systems. As embedded systems will be more and more interconnected and become parts of the different ICT networks, higher security for embedded systems also means *increased trust in embedded areas of Future Internet technology.*
↪ The envisioned security engineering process will support the exploration of the security design space with respect to particular security requirements of embedded systems, based on characteristics and restrictions of available hardware and software. Users will be supported in realizing trustworthiness of a system throughout its life cycle by creation of meaningful and contextual interactions, tailor-made to the embedded services through the SecFutur development process and tools. This also has to include increased support for fulfillment of legal requirements e.g. for calibration and gauging of measurement devices.

Goal 2: Embedded systems development in most cases cannot rely on expensive taylor-made hardware and software in order to achieve high security. Therefore, a security engineering process needs to enable *efficient development of secure embedded systems* on the basis of existing hardware and software.
↪ This will be achieved by providing security building blocks for embedded systems, each implementing a specific complex non-functional property using proven efficient methods.

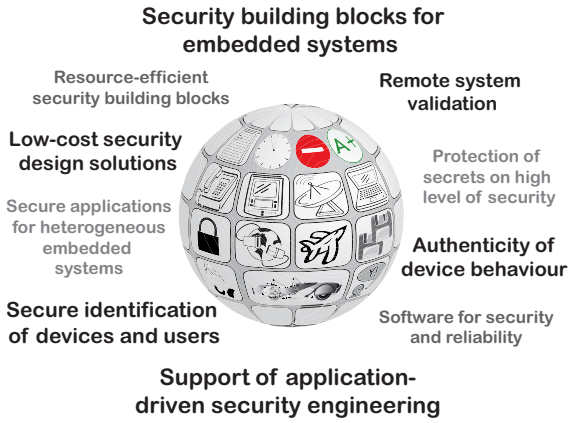

Figure 1: Main elements of a security engineering process for embedded systems

Goal 3: Security always needs to be seen relative to particular security requirements. Therefore, it is essential to provide *application driven security engineering* of embedded systems.
↪ In order to aim at the application view of security, the security engineering process will consider all stages of the development of embedded systems, and will demonstrate how application specific requirements are captured by combinations of security building blocks at different phases of the development process.

Goal 4: Security engineering can only be useful if applying the process indeed significantly *increases the security and thus the overall quality* of future designed embedded systems.
↪ This will be achieved by exact specification of security requirements and then using validated or verified security building blocks made available through the security engineering process. Furthermore, the engineering itself needs to be complemented by tools for security validation and testing.

4. APPLICATION DOMAINS

Embedded systems underpin developments in many technical areas such as automotive, avionics, telecommunications, plant automation, medical systems, and consumer electronics. Embedded devices create revenues for their owners and have essential roles in critical scenarios; hence their correct functioning is essential. Communication links as well as embedded devices themselves are increasingly subject to attacks already known from other IT infrastructures. Traditional protection based on physical separation from malicious environments is no longer sufficient. One reason is that an increasing number of embedded devices are equipped with communication interfaces and communicate via open network infrastructures (e.g. Internet, wireless, peer-to-peer), and helpful advances in user control such as comfortable browser-based configuration introduce new attack vectors, specifically if the devices are used in systems of systems. In networked embedded devices with interfaces to information infrastructures, dependable cooperative reasoning may in-

duce emerging security requirements in systems of systems. Mobility leads to new kinds of attack scenarios such as physical transportation of malware from one part of a system into another part across firewalls and other protection means.

Thus, identification of devices, control over the operational state of a device, security of the communication between devices, or even non-repudiation of actions taken at a particular time, are security requirements that are becoming increasingly important in today's embedded scenarios; at the same time these requirements are not supported by existing combinations of hardware and software. Furthermore, with the exception of mobile platforms, security issues are typically neglected in current research and development projects involving embedded systems.

Complex application scenarios like plant automation, remote metering, or critical communication depend on the secure and reliable operation of each single embedded component involved. Additionally, Future Internet thinking means to plan beyond specific pre-planned applications of intelligent devices.

The following paragraphs introduce some concrete areas and their particular security challenges. These use cases show the scope of the work in the SecFutur project.

4.1 Multi-service gateways

Domain

Security of the digital home. Service gateways. SOHO (small office, home) protection and network protection.

Use Case Description

This use case focuses on the home and small businesses networks where Operators/ISPs foresee an important emerging market. In this environment the Operator provides some embedded hardware to be installed in customer's home. This hardware, named usually a Service Gateway or Home Gateway, is the key element connecting the home network and the broadband network, and acting as a service platform in the digital home. Such a gateway will also serve as the basis for various additional services such as supporting smart grid applications within advanced energy distribution and control infrastructures. All these facts make it the main target of the use case.

Security Challenges

The Service Gateway is an embedded hardware which plays a critical role for various different services and therefore demands many security features. Any threat affecting this node is potentially harmful to the customer and/or operator. It shall be protected to assure both the security of the customer and the operator networks and data. This means providing features like protecting Service Gateway against unauthorized modification, controlling the security status of the hardware and installed software, and enhancing the security features of the services. Also, all the communications involving the Service Gateway must be secured, providing confidentiality, integrity and privacy. In this sense, identifying the devices by the operator network is an important aspect to be achieved. Last but not least, the strategic position of this device allows to consider it as a starting point to inspect the traffic flowing through for threat detection and prevention. This includes modeling the user's network traffic in order to detect anomalies such as bot-nets.

4.2 Legal calibration requirements

Domain

Metering devices with legal calibration requirements

Use Case Description

Devices for measurement (e.g. of electricity consumption, of petrol consumption in petrol stations, etc.) are subject to strict legal and technical requirements. One example of such devices are electronically calibrated devices for electricity consumption measurement. A successful licensing of the design of measurement devices is based on licensing procedures that are based on experiences with existing implementations.

Security Challenges

New technical approaches for the development of measurement devices need suitable adaptations of the regulations for the design licensing. This holds in particular for approaches that are based on a purely electronic implementation of security and dependability. Security requirements in this domain include credibility and reliability of the measurement devices. The fulfillment of these requirements is controlled and attested to by national calibration authorities. A license issued in one EU member state is valid in all other member states. Basis of the credibility of metering devices is the trust of the general public in the dependability of the measuring-technical systems. The public interest in the dependability and security of measuring devices is high, without sufficient actions being taken to ensure their dependability and security, electronic measuring instruments cannot be accepted as device according to legal regulations. Additionally, legal aspects of gauging have to be adapted to support the use of advanced technology in a secure and dependable way. Another considerable advantage of digital electronic metering devices is the ability to serve multiple clients with one device. For this the individual data and procedures of the clients have to be isolated from each other in a very reliable way. Without proof of sufficient security mechanisms this advantage cannot be achieved. in particular, legal regulations concerning data protection have to be fulfilled.

4.3 Ad-hoc mesh communication

Domain

Secure Ad-hoc wireless mesh communication for crisis management

Use Case Description

Interoperability and robustness of secure communication for blue light organizations even in the case of infrastructure communication blackout (i.e. collape of 3G and mobile communication) demand for new easily deployable ad-hoc communications for data-, voice-, video- and sensor-information.

Security Challenges

Reliable information dissemination and authenticity of information is crucial in crisis management communication. In addition to the need for communication security and authenticity, ease of deployment dramatically increases the challenges of information security. Therefore the communication system needs to cope with mobility, robustness and ad-hoc trusted network establishment. Additionally, effective blue light organization communication equipment shall

be open in terms of future applications and will be based on standard communication links such as WLAN because of future bandwidth needs. Therefore, crisis management communication has to cope with the challenge of general COTS based complex embedded systems and inherent high security needs. By designing sound security architectures and measures we aim to resolve these conflicts of objectives. The security challenges lie in the support for ease of deployment and interoperability in order to increase the potential for saving lives.

5. INNOVATION TOWARDS SECURITY ENGINEERING FOR EMBEDDED SYSTEMS

This section describes the short term approach and partly long-term vision to be realized during the next years by SecFutur and other related research and development.

1. The first key innovation is a new security engineering process and supporting tools especially tailored for the needs of embedded systems. It first provides a model framework for security aspects of embedded systems which can be integrated into existing engineering processes for embedded systems. Then, it supports the developer in identifying security requirements based on the model very early in the development process of embedded applications and systems. Finally, the developer is guided step-by-step through design decisions regarding security requirements to be addressed and considering restrictions to resources for the underlying devices and communication links. Non-security related design decisions and restrictions to the design space can be incorporated into the process at any level.

2. The second innovation will be a set of security building blocks for embedded systems running on established hardware platforms like 8051, ARM, AVR, 68k/Coldfire, Intel x86, 68HC12,and other 8/16/32-Bit-CPUs. By making use of existing hardware trust anchors (Trusted Platform Module (TPM), ARM TrustZone or M-Shield), these building blocks will provide assurance with respect to secure system states. As embedded systems require a tight integration of soft- and hardware, security building blocks will combine hardware trust anchors with suitable existing software security mechanisms. Furthermore, they will allow for means to isolate different applications owned by different stakeholders but running on the same platform. Novel security mechanisms for wireless communication in the event of existing infrastructure overloads such as *energy-efficient reliable manycast* on top of commodity devices will be considered as a building block.

3. Finally, the security engineering process needs to be complemented by a set of tools for security validation and testing to be used during design time, after re-configuration, evolution and updates, as well as at run-time for security monitoring. The tools will rely on existing validation and test tools and extend them with respect to security properties and also with respect to conditions and requirements imposed by security building blocks used in a specific embedded target of evaluation. The results will be demonstrated on three show cases in the area of set-top boxes, metering validation, and crisis management.

6. RELATED WORK

The focus of the embedded systems industry and research community in the last decade has been on the reduction of costs, down-sizing and faster time to market through efficient design processes and tools. This has led to both explosion of the embedded market and a drive for higher market shares by aiming for higher reliability and quality. However, embedded systems have been physically protected from massive security threats to such an extent that security attributes like confidentiality, integrity, and availability have hardly been regarded as a central feature in the design of such systems.

However, security engineering approaches for embedded systems can build on a large body of work in IT Security. In particular, a large variety of security mechanisms already exists and is ready to be deployed for embedded systems. These include hardware mechanisms such as crypto processors and TPMs, and software libraries. This is particularly true for embedded systems using standard architectures (e.g. resource-efficient PC architectures such as 8051, ARM, AVR, MIPS, PowerPC, 68k/Coldfire, Intel x86, 68HC12, C167). Remarkably, only in very few areas security mechanisms are included into embedded systems' development, and if so, they are tailored to specific applications. This was partly caused by the fact that traditionally embedded system development was much stronger concentrated on resolving safety and cost issues. These constraints will have a modified impact by supporting the design process of embedded systems in order to allow for the development of highly secure systems.

Even though embedded systems are becoming prevalent in the every day life of all citizens of the European union, there is not much focus on security-driven requirements of embedded systems [19]. Marwedel [12] pays a little attention on dependability requirements, more or less an exception of its kind. On the other hand, we are getting to a point where "networked and embedded systems security" is appearing on the agenda of major conferences with a distinct software profile (e.g. EMSOFT). Similarly, major textbooks on security are no longer entirely focusing on host and network security. For example the latest revision of Security Engineering by Ross Anderson [5] has several chapters devoted to the basic vulnerabilities of devices and systems used for banking, metering, and wireless mobile communication, signaling the increasing importance of this area.

Several concepts have been introduced as the central element for the definition of a development process for highly dynamic and heterogeneous systems with embedded elements. The component-based approach [10, 11, 16] has proven to be very appropriate for the type of scenarios relevant in embedded systems.

The main interest of component composition is to build new systems from their requirements by systematically composing reusable components. In general, this concept is not appropriate to represent security solutions because security mechanisms can not always be represented as units that can be connected to the rest of the system by means of well defined interfaces. This makes the research towards security engineering equally challenging and important. Many security mechanisms are not about "what" but about "how".

In fact, software components are good abstractions of functional elements, but security is a non-functional aspect. Being a system level concern it resembles safety or timeliness properties that are cross-cutting concerns [8, 18]. Finally, the use of components in embedded scenarios introduces important issues related to the fact that the systems are inherently distributed, heterogeneous and not under the control of a single entity.

There is very little work concerning the full integration of security and systems engineering from the earliest phases of software development. Although several approaches have been proposed for some integration of security, there is currently no comprehensive methodology to assist developers of security sensitive systems. Lack of support for security engineering in approaches for software systems development is usually seen as a consequence of: (i) security requirements being generally difficult to analyze and model, and (ii) developers lacking expertise in secure software development. All this becomes a special concern when considering complex security requirements such as those associated to applications in e-commerce, e-government and e-health scenarios.

Existing approaches are not comprehensive enough in the sense that they focus either on some special stage of development, e.g. on design or implementation, or on a specific security aspect such as access control. Moreover, they typically offer no guidance on how they can be integrated into current component or model based system development methods. Empirical studies confirm this view [13, 9]. Necessary are comprehensive and integrative approaches supporting the integration of security and systems engineering. Thus, work can build on results of the FP6 project SERENITY [2] which provides strong results on security engineering for applications in Ambient Intelligence environments.

One of the most interesting approaches for introducing security in development cycle is presented by Dimitrakos et al. [7]. Model Driven Security is a specialization of the MDA approach that proposes a modular approach combining languages for modelling system design with languages for modelling security. Dimitrakos et al. introduce an application for constructing systems from process models by combining a UML-based process design language with a security modelling language for formalizing access control requirements. They are able to automatically generate security architectures for distributed applications from models in the combined language. There are more commercial tools like "The ArcStyler MDA-SecurityTM Cartridge" [14] that captures the skills of security experts and makes them available for persons who need to deliver secure systems.

A further very active area of research is security engineering based on patterns [15, 17]. The Open Group is preparing a book on the subject [3]. Research into investigating a template for security patterns that is tailored to meet the needs of secure system development has been recently reported by Cheng et al. [6], where the UML notation is used to represent structural and behavioral aspects of design, and formal constraints to the patterns are used to enable verification. However, security patterns are often not precisely described and therefore, automated tools for classification, selection and composition are not yet available.

One basic standard for security engineering is the Systems Security Engineering Capability Maturity Model (SSE-CMM) [4]. The model highlights the relationship between security engineering and systems engineering, regarding the former as an integral part of the latter rather than as an end to itself.

This overview shows that, based on the variety of different approaches, several special security properties can be considered separately more or less rigorously in the development process. In contrast to these approaches, it will be necessary to provide an integrated security engineering process with tool support which allows rigorous treatment of security and reliability requirements and makes existing security solutions available for security engineering.

7. CONCLUSIONS

This paper describes the vision of security engineering for embedded systems. Some parts of this vision will be realized by running research and development projects, but obviously there are many open issues and there is the need for synchronisation and co-operation between the different projects and research directions. The goal of this paper is to encourage discussion and cooperations.

8. REFERENCES

[1] http://www.secfutur.eu.

[2] http://www.serenity-project.eu.

[3] http://www.opengroup.org/security/gsp.htm.

[4] Systems security engineering capability maturity model (sse-cmm), model description document, version 3.0. http://www.sse-cmm.org/model/ssecmmv2final.pdf.

[5] R. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, 2008.

[6] B. H. Cheng, S. Konrad, L. A. Campbell, and R. Wassermann. Using security patterns to model and analyze security requirements. Technical Report MSU-CSE-03-18, Department of Computer Science, Michigan State University, 2003.

[7] T. Dimitrakos, D. Raptis, B. Ritchie, and K. Stolen. Model based security risk analysis for web applications. In *In Proc. Euroweb 2002*, 2002.

[8] J. Elmqvist, S. Nadjm-Tehrani, and M. Minea. Safety interfaces for component-based systems. In *Proceedings of the 24th International Conference on Computer Safety, Reliability and Security (SAFECOMP)*, 2005.

[9] J. Jürjens. Towards development of secure systems using umlsec. In *Lecture Notes in Computer Science*, volume 2029, 2001.

[10] D. Llewellyn-Jones, M. Merabti, Q. Shi, and B. Askwith. Utilising component composition for secure ubiquitous computing. In *Proceedings of 2nd UK-UbiNet Workshop*, 2004.

[11] H. Mantel. On the composition of secure systems. In *Proc. of IEEE Symposium on Security and Privacy*, 2002.

[12] P. Marwedel. *Embedded System Design*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[13] A. Maña and G. Pujol. Towards formal specification of abstract security properties. Technical report, University Malaga, 2008.

[14] I. Objects. ArcStyler Cartridge guide for ArcStyler Version 3.x for MDA-security. http://www.interactive-objects.com/products/mda-extensions/mda-security-with-arcstyler.

[15] M. Schumacher and U. Roedig. Security engineering with patterns. In *Pattern Languages of Programs 2001*, 2001.

[16] Q. Shi and N.Zhang. An effective model for composition of secure systems. *Journal of-Systems-and-Software*, 43(3), 1998.

[17] C. Steel, R. Nagappan, and R. Lai. *Core Security patterns*. Pearson Education Inc., 2006.

[18] A. Tesanovic, S. Nadjm-Tehrani, and J. Hansson. *Component-Based Software Development for Embedded Systems - An Overview on Current Research Trends*, chapter Modular Verification of Reconfigurable Components. Springer Verlag, 2005.

[19] F. Vahid and T. Givargis. *Embedded System Design: A Unified Hardware/Software Introduction*. J. Wiley and Sons, 2002.