The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
21-23 September, 2017, Bucharest, Romania

Design Lifecycle for Secure Cyber-Physical Systems based on Embedded Devices

Dmitry Levshun, Andrey Chechulin, and Igor Kotenko
St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences (SPIIRAS)
39, 14th Liniya, St. Petersburg, Russia,
St. Petersburg National Research University of Information Technologies, Mechanics and Optics (ITMO University),
49, Kronverkskiy prospekt, Saint-Petersburg, Russia
{levshun, chechulin, ivkote}@comsec.spb.ru, http://comsec.spb.ru

***Abstract*—The paper is devoted to the issues of design of secure cyber-physical systems based on embedded devices. It aims to develop a generalized approach to the design of secure systems based on embedded devices. Current approaches to design secure software and embedded devices are analyzed. The design lifecycle for secure embedded devices system is proposed. Advantages and disadvantages of the approach are analyzed. The correctness of design lifecycle for secure embedded devices systems is validated by its use in the development of the integrated cyber-physical security system.**

Keywords*— *cyber-physical systems, embedded devices, design of secure cyber-physical systems, security of embedded devices systems.

I. Introduction

Nowadays the integrated approach to providing cyber-physical security of critical infrastructures is widely spread. This approach is to combine heterogeneous sources of events of the physical and cyber levels within one system, as well as to ensure the resilience of such systems to attacks on them [1]. This approach allows to detect security incidents, scenarios of attacks, and provide anomalous activity detection, which previously was possible only at the stage of investigation, as well as to respond to them in real time.

In addition, there are national and international standards for the development of secure software, as well as solutions from leading companies in the domain of information technology. Besides that, the frameworks for design of the software architecture taking into account the need to ensure its security are widely spread.

We should note that the functionality of the embedded devices is determined not only by software but also by hardware. Connection between the software part of the device, on the one hand, and hardware, on the other, lead to the presence of additional constraints which have significant effect on the design process of such devices. This means that existing solutions to build secure software are not applicable to the design of secure systems based on embedded devices in full, and therefore there is necessity of their revision.

Also in practice the component approach to development of embedded devices is widely used. It is implemented, for example, within the context of the Arduino, Raspberry Pi, Beaglebone, and Intel Galileo. For this approach there are the techniques of designing secure embedded devices, which allow at the design stage of an embedded device to identify the list of possible attacks, which may affect the device, in accordance with the selected model of the intruder, and used software and hardware components [2]. However, the use of such methods and further consolidation within a single system, combining multiple secure embedded devices, does not allow to develop a secure system due to the need of considering the emergent properties of the system. This means that these techniques are also not applicable to the task fully and there is a need of their revision.

In addition, in recent years, the research in the field of methodologies for design and verification of networked embedded systems [3] are widely spread. The main purpose of these techniques is to provide developers of networked embedded systems with information about the applicability of certain interfaces and data transfer protocols to ensure appropriate level of reliability of the final system. The security of embedded devices is not an immediate goal of these methods, however, some aspects of security are considered when selecting interfaces and data transfer protocols. It is also important to note that methods of this type can ensure the reliability only for isolated embedded devices, without considering their interaction with remote servers, workstations, web services, etc. This means that the methodology of design and verification for networked embedded systems is not applicable to the task of providing security of not isolated systems based on embedded devices, and therefore also needs to be extended.

To summarize, it is important to note that at the moment there is no single generalized approach for solving the problem of designing secure systems based on embedded devices; and existing solutions have drawbacks and need improvement. This paper aims to develop a generalized approach to the design of secure systems based on embedded devices. The *main contribution* of this paper is to develop a unified technique to design secure systems based on embedded devices, taking into account the emergent properties of the protected system. The *novelty* of the proposed approach is in combination of solutions on the development of secure software with the techniques for design of secure embedded devices in a single design procedure. Thus as emergent properties are considered the properties that appear in the system based on interfaces and data transmission channels between elements of the system.

The paper presents main elements of the research and development performed; its structure is organized as follows. *Second section* discusses the main results of the previously fulfilled relevant research. *Third section* provides a general description of the developed approach. The correctness of the design life cycle for embedded devices system is verified by its use in the development of an integrated cyber-physical security system specified in *forth section*. Advantages and disadvantages of the approach are analyzed in *fifth section*. The main conclusions and further research directions are discussed in *sixth section*.

II. Related work

As one of the possible solutions for developing secure software, let us consider a solution from Microsoft, namely the Microsoft Security Development Lifecycle (Microsoft SDL) [4]. The approach of the company is divided into seven key sequential phases: training, requirements, design, implementation, verification, release and response. The fundamental phases from the point of view of the development of a secure system based on embedded devices are requirements and design. This is due to the fact that the immediate task of design technique is providing input on security requirements, quality gates/bug bars, security and privacy risk assessments on the phase of requirements. In addition, an equally important goal of the technique is providing input data on design requirements and threat model for the design phase.

Another possible solution for developing secure software is the solution from Cisco – Cisco Secure Development Lifecycle (Cisco SDL) [5]. The company's approach consists of six sequential phases: product security requirements, third party security, secure design, secure coding, static analysis, vulnerability testing. From the point of view of the development of a secure system based on embedded devices, the most important phases are product security requirements and secure design. So, on the secure requirement phase the gap analysis is done, whose main task is to identify the necessary changes in the system to achieve the safe state. And in the phase of the secure design the process of threat modeling is done to make assumptions for possible threats and ways to mitigate them. In addition, one of the interesting features of SDL Cisco compared to Microsoft SDL is a third party security phase, aimed at identifying possible threats from third party software, as well as to ensure registration and timely updates of this software.

To develop complex enterprise-level software systems the frameworks of appropriate level are usually used. One of such frameworks was the Zachman Framework [6]. In this framework, two classical approaches for the solution of analytical problems were used. The first approach is based on answering six key questions: what, how, when, who, where and why. It is important to note that basing on the answers to these questions one can form a holistic description of rather complex processes. The second approach consists of six consecutive development phases, namely identification, definition, representation, specification, configuration and instantiation. So the Zachman Framework has 6x6 matrix format in which columns correspond to question words, and rows are the phases of development. Each cell in the resulting table represents the corresponding simple model. The application of the Zachman Framework to design secure systems based on embedded devices allows to organize business logic of the system that provides the ability to form corresponding security requirements.

One possible approach for designing secure embedded devices is presented in papers [7, 8]. The essence of the technique proposed in these papers is to identify and account the list of possible harmful effects, to which the embedded device may be subject in accordance with the selected model of the intruder, and also by used hardware and software components, already in the design phase. In this approach, the protection tools are direct part of the embedded device, ensuring its security. Let us consider the main phases of the specified technique in more detail: (1) definition of functional requirements for the embedded device; (2) definition of non-functional requirements to the embedded device; (3) identifying the set of alternatives of component composition of the embedded device in accordance with the functional requirements; (4) choice of the optimal component composition of the embedded device from the point of view of non-functional requirements; (5) identification of the list of possible harmful impacts on the embedded device based on the static testing.

Thus, if the security level of an embedded device is sufficient, one can proceed to the stage of direct development. Otherwise, one should return to the first step and review the functional requirements. Unfortunately, a system based on the interaction of embedded devices, each of which is designed in accordance with the methodology for designing secure embedded devices, cannot be considered secure due to unique emergent properties occurring during operation of the system.

In order to develop a system of secure communications between embedded devices a variety of techniques are also used. One example of such techniques was presented within the framework of the European research project SecFutur [9], devoted to the development of systems with embedded devices. In this project it was proposed the use the topological approaches for building secure channels for data transmission between embedded devices. To solve this problem the calculation of the security of the path between two points in a network graph was performed basing on some numeric values of security assigned to the nodes. After it this characteristic served as the basis for changes of requirements on embedded devices. However, this approach does not take into account the interaction of systems of embedded devices with external systems (or takes into account the interaction only from embedded devices), which may cause problems at integrated protection of the network containing embedded devices.

III. General Approach

The developed approach combines solutions for developing secure software with models and techniques of designing secure embedded devices in a single integrated technique for designing secure systems based on embedded devices, taking into account the emergent properties of the protected system.

The first step of the proposed technique for designing secure systems based on embedded devices is the *definition of functional and non-functional requirements* for the developed system based on embedded devices. The functional requirements may be divided into requirements for embedded devices of the system, the requirements for the software of the system, and requirements for interfaces and data transfer protocols, on the basis of which the further cooperation of embedded devices and software in the framework of the designed system is performed. To limit the dimension of the final selection of possible alternatives of embedded devices and software of the system the non-functional requirements are formulated. Typically, valid non-functional requirements specify the range of cost, power consumption and dimensions of the embedded devices, and valid range of value and resource consumption of the software. Further, the requirements for embedded devices are taken into account in the implementation of the design technique for secure embedded devices (step 2), software requirements are used at implementing the design technique for secure embedded software (step 3), and the requirements for interfaces and data transfer protocols are used at implementation of the design technique for the secure embedded device system (step 4) as well as design technique for secure embedded devices (step 2), and the design technique for secure software (step 3).

The second step of the proposed technique is to *apply the technique of designing secure embedded devices*. In this step the functional requirements provided in the first step are analyzed to identify possible alternatives of component composition of embedded devices. In addition, the check of the correspondence of the obtained alternatives to the non-functional requirements, identified in the first step, is performed. It is important to note that according to the results of the analysis of functional and non-functional requirements, it may be concluded that the correspondence of some functional requirements may be done only partially or it may be totally impossible because of too strong restrictions of non-functional requirements.

To resolve this situation, the proposed technique provides production of relevant notification about necessity for making changes in the functional and/or non-functional requirements for the system based on embedded devices, as well as the recommendation for the refusal from some of them.

The list of possible alternatives of component composition of embedded devices directly depends on the quality of the knowledge base, which is used by the technique for design of secure embedded devices. However, as soon as the list of possible alternatives of component composition of embedded devices for the designed system is formed, among them the optimal one from the point of view of non-functional requirements will be selected. Further, in accordance with the technique for designing secure embedded devices, basing on static testing the list of possible harmful impacts on the model of the embedded device is analyzed, and the model is adjusted. Thus, at the completion of the second step of the proposed technique there will be generated the secure embedded device model, information about which will be transferred to the secure design technique for embedded system devices (step 4).

On the third step of the proposed technique *the analysis of security and privacy requirements and the design requirements to the software* of the system based on embedded devices is performed. Further, on the basis of the security and privacy risk assessments and threat model static testing process is performed. As in the previous step, while performing a technique of designing secure software there may be done a conclusion that the satisfaction of individual functional or non-functional requirements is partially possible or impossible at all. Usually this is due to the lack of computing performance used by embedded devices or with application of not cross-platform solutions. In such a situation, you will also receive a notification of the need for partial changes in the functional and/or non-functional requirements, and about refusal from some of them. Thus, at the completion of the third step of the proposed technique there will be compiled secure software model, information about which will be transferred to the design technique for secure embedded devices system (step 4).

The fourth step of the proposed technique is *the use of design technique for secure embedded device system*. The essence of this technique is the formation of an embedded devices system model based on the requirements to the interfaces and data transfer protocols, as well as on the formed in the previous steps secure embedded device model (step 2) and secure software model (step 3). When performing this technique, initially a list of possible alternatives of models of systems, based on embedded devices that meet functional requirements, is formed.

Then the conformity of the obtained alternatives to the restrictions imposed by non-functional requirements is checked, and among them the optimal one from the point of view of data requirements is selected. After it, basing on the static testing, the list of possible harmful impacts on the embedded device system model is analyzed, and the model is adjusted. Thus, at the completion of the fourth step of the proposed technique, the secure embedded device system model is generated, information about which will be transferred to the final stage of the implementation of secure system based on embedded device (step 7).

The fifth step of the proposed technique is an *embedded device manufacturing process*. In this step, basing on information about the secure embedded device model (step 2) and secure embedded devices system model (step 4), the real devices are produced. Their security is determined by application of the corresponding technique. After it secure embedded devices are transferred to the final stage of the implementation of secure system based on embedded devices (step 7).

The sixth step of the proposed technique represents the *implementation and verification phases of secure software*. In this step, basing on information about the secure software model (step 3) and the secure embedded devices system model (step 4), the software is developed. The protection of the software is determined by the application of the corresponding technique. After it the secure software is transferred to the final stage of the implementation of the secure system based on embedded devices (step 7).

The seventh (final) *step* of the proposed technique is the *implementation of the secure system based on embedded devices*. At this step the system is realized by locating the used embedded devices, laying of communication lines between them and their setup. In addition, the software and related protection mechanisms are installed and configured. The result of this step is the ready-to-use secure system based on embedded devices. The security of the system is determined by a set of techniques: the design technique for secure embedded devices at the level of embedded device model; design technique for secure software at the level of the software model; design technique for secure embedded devices on the level of embedded devices system model, as well as Software SDL at the software development phase. The combination of these techniques and instruments represents the lifecycle of development of secure systems based on embedded devices or Design Lifecycle of Secure Embedded Devices System (DLSEDS), shown in Figure 1.

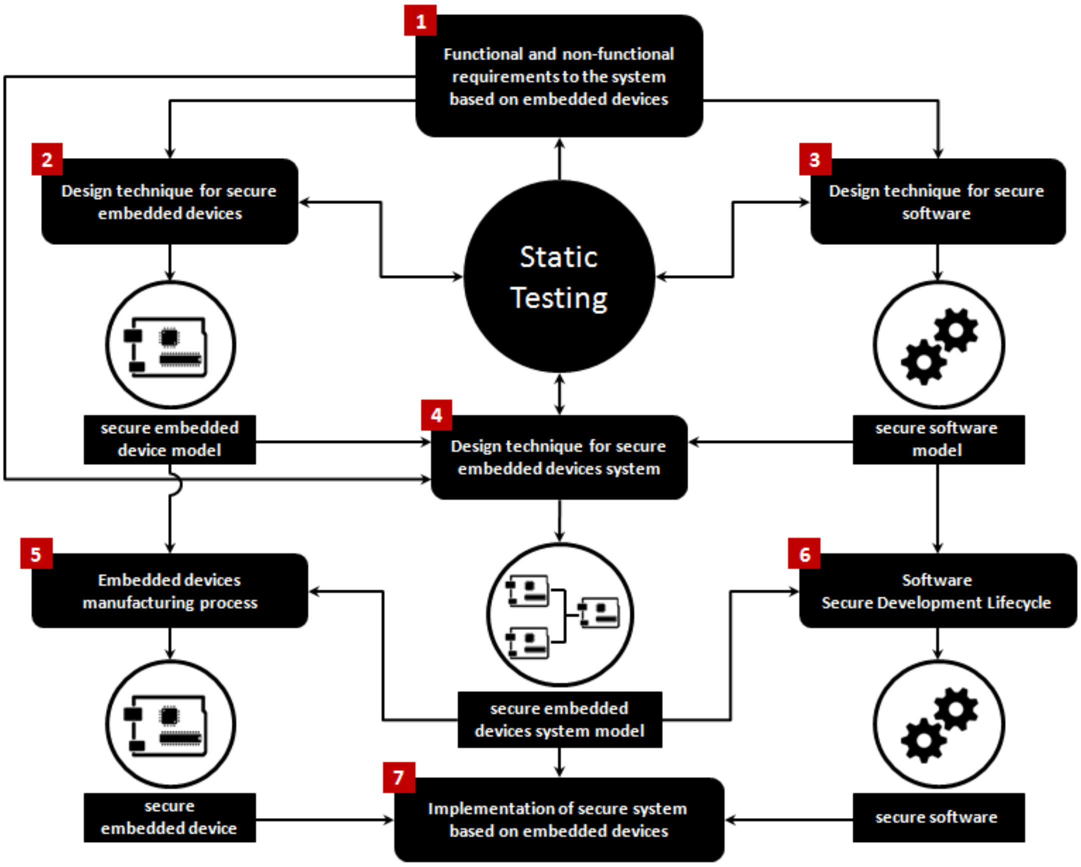
1
Functional and non-functional requirements to the system based on embedded devices
2
Design technique for secure embedded devices
3
Design technique for secure software
Static Testing
secure embedded device model
4
Design technique for secure embedded devices system
secure software model
5
Embedded devices manufacturing process
6
Software Secure Development Lifecycle
secure embedded devices system model
secure embedded device
7
Implementation of secure system based on embedded devices
secure software

Figure 1. Design Lifecycle of Secure Embedded Devices System

IV. Application of the Approach

The architecture of the integrated cyber-physical security system [10], designed with the use of DLSEDS, is shown in Figure 2. The integrated cyber-physical security system consists of four main modules: hardware interfaces (module 1), software interfaces (module 2), hubs (module 3), integrated cyber security system server (module 4).

Let is consider each module more in detail.

Module 1. Hardware Interfaces. These modules can be implemented as microcontrollers, that aimed to collect information from external sources and convert the external format of events to the internal format. The resulting stream of events goes to hub. To enhance to security of the proposed system the hardware interfaces use the encryption algorithms for protecting the channel between hub and them. Also they perform the mutual authentication procedure to protect the system against fake modules.

Module 2. Software Interfaces. These interfaces are used to collect data from computers and other cyber physical security systems by special drivers.

Module 3. Hubs. These modules can be implemented as high-performance microcontrollers. The aim of these modules is to collect data from software and hardware interfaces and to perform the data normalization and pre-processing; after that the data are stored and presented to the user by the web-interface (if the controlled system is small) or forwarded to the server of the integrated cyber-physical security system (if the controlled system is large).

Module 4. The server of the integrated cyber-physical security system. This module can be implemented as the computer (the required performance of which depends on the size of the controlled system). This module includes several components, namely: collection component (it receives data from the hubs), database component (it stores collected and processed data), data processing module (it correlates entry logs from the database to automatic detection of incidents, attack scenarios and anomalous activity) and visualization component (it shows the results to the operator and helps him to make decisions). The results of the work of the processing component also can be sent to the external systems (e.g., security information and event management systems) to provide the high level representation of the detected incidents, attack scenarios and anomalous activity.

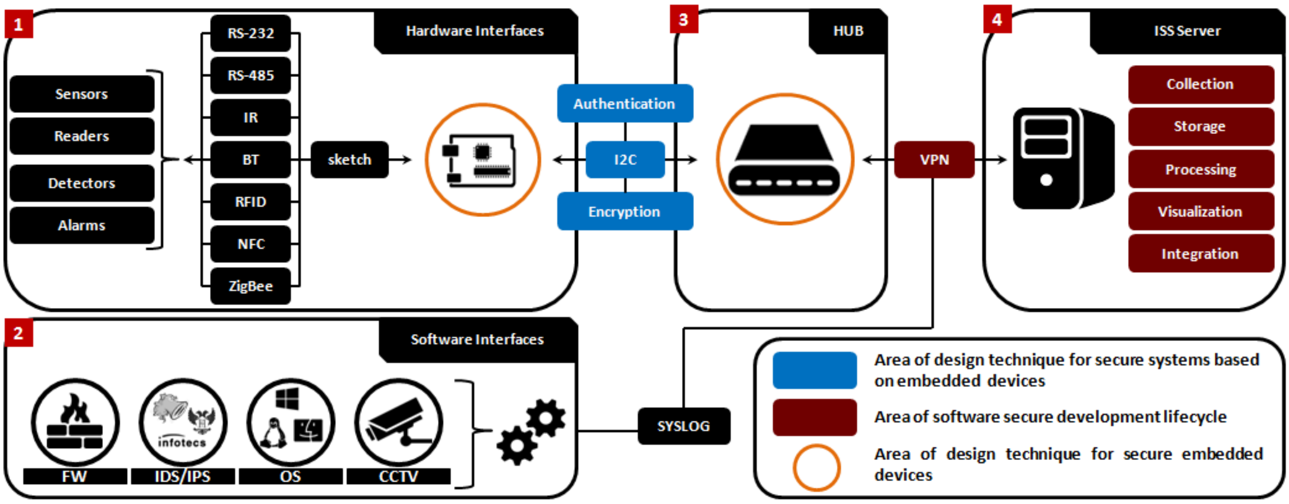

Figure 2. Integrated cyber-physical security system as the application of the DLSEDS

One of the important tasks for applications of DLSEDS for designing secure systems based on embedded devices is the formation of a secure environment of data transmission from detectors, alarms, sensors, readers and other external electronic components connected to embedded devices. However, the interaction of embedded devices with the specified data sources, as a rule, depends on the interface, supported by the given data source, and data transmission protocol, and thus it is not possible to affect the security of this cooperation within the framework of DLSEDS.

Alongside the area of DLSEDS may be divided into the area of design technique for secure systems based on embedded devices, the area of secure software development lifecycle and area of design technique for secure embedded devices. Thus, in the process of applying the design technique for secure systems based on embedded devices, it was decided to expand the functionality of the I2C protocol by mechanisms of the mutual authentication of embedded devices and encryption of transmitted data.

According to the results of the performed expert assessment, it was confirmed that the use of the developed approach allows to enhance the security of the developing system. Additionally, experts noted that DLSEDS allows to reduce the time spent on the development of secure systems based on embedded devices by automation of alternatives generation, taking into account the possible conflicts between the system elements and embedded devices.

V. Discussion

DLSEDS allows developers to design complex secure systems based on embedded devices without involvement of experts in the domain of embedded devices security. This assumes that the embedded devices of the system are also protected, and their composition is rational or optimal from the point of view of the requirements. Unfortunately, the list of possible alternatives of component composition of embedded devices, as well as information on supported interfaces and data transfer protocols depend on the quality of the knowledge base, which is used by the technique of designing secure embedded devices, as well as the technique of designing secure systems based on them. This means that the quality of the solution, provided by DLSEDS, directly depends on the completeness and relevance of the used knowledge base, and therefore DLSEDS still is not a full replacement for expert opinions.

An expert in the field of security systems based on embedded devices, having knowledge about the specific solutions and existing best practice, as a rule, chooses the component composition of embedded devices, as well as interfaces and data transfer protocols for their interaction with each other and the server software on a qualitatively higher level. On the other hand, DLSEDS may be useful to the expert as a tool to automate some routine tasks, as well as a source of solutions that differ from his (her) subjective preferences.

VI. Conclusion

In this paper we analyzed existing approaches to the development of secure software, as well as the techniques of designing secure embedded devices for their applicability for solving the problem of designing secure systems based on embedded devices. As the result of the performed analysis, the authors came to the conclusion about necessity to develop the own technique for the design of secure systems based on embedded devices, because none of the analyzed approaches or their combination allowed to solve the problem in full.

As a result the Design Lifecycle of Secure Embedded Devices System (DLSEDS) approach was developed. This approach represents the combination of the approach to the development of secure software, the technique for design of secure embedded devices, and the developed technique for designing secure systems based on embedded devices. The last suggested technique acts as a link, formulating requirements both to the technique for designing secure embedded devices and to the approach for developing secure software and also provides security of data transmission channels between the protected embedded devices.

The correctness of design life cycle for secure embedded devices systems was validated by its use in the development of the integrated cyber-physical security system.

In further research on this topic it is planned to conduct additional experiments on the use of DLSEDS, to expand the existing knowledge base on component composition of embedded devices, supported interfaces and data transfer protocols, and application of vulnerabilities database to improve the efficiency of the process of static testing.

Acknowledgment

This research is being supported by the grants of the RFBR (15-07-07451, 16-37-00338, 16-29-09482), partial support of budgetary subjects 0073-2015-0004 and 0073-2015-0007, and Grant 074-U01.

References

[1] V.A. Desnitsky, D.S. Levshun, A.A. Chechulin and I.V. Kotenko, “Design Technique for Secure Embedded Devices: Application for Creation of Integrated Cyber-Physical Security System,” *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, Vol.7, No.2, 2016. pp. 60-80.

[2] J.F. Ruiz, V.A. Desnitsky, R. Harjani, A. Manna, I.V. Kotenko and A.A. Chechulin, “A Methodology for the Analysis and Modeling of Security Threats and Attacks for Systems of Embedded Components,” *Proceeding of the 20th International Euromicro Conference on Parallel, Distributed and Network-based Processing (PDP 2012)*. Garching/Munich, February, 2012. pp. 261-268.

[3] F. Stefanni, “A Design & Verification Methology for Networked Embedded Systems,” *Ph.D. Thesis. University of Verona, Department of Computer Science, Italy*. April 7, 2011. 143 p.

[4] M. Howard, S. Lipner, “The Security Development Lifecycle. SDL: A Process for Developing Demonstrably More Secure Software,” *Microsoft Press, Redmond, Washington*, 2006. 320 p.

[5] Official Cisco Secure Development Lifecycle documentation. http://www.cisco.com/c/en/us/about/security-center/security-programs/secure-development-lifecycle.html, last visited on 22.02.2017.

[6] C. O’Rourke, N. Fishman, W. Selkow, “Enterprise Architecture Using the Zachman Framework,” *Published April 15th 2003 by Course Technology*. 752 p.

[7] V.A. Desnitsky, A.A. Chechulin, I.V. Kotenko, D.S. Levshun, M.V. Kolomeec, “Application of a Technique for Secure Embedded Device Design Based on Combining Security Components for Creation of a Perimeter Protection System,” *24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP 2016). Heraklion, Greece*, February, 2016. *IEEE Computer Society*. 2016. pp. 609-616.

[8] V.A. Desnitsky, I.V. Kotenko, A.A. Chechulin, “Configuration-based approach to embedded device security,” *Lecture Notes in Computer Science, Springer-Verlag. The Sixth International Conference "Mathematical Methods, Models and Architectures for Computer Networks Security" (MMM-ACNS-2012)*. October 17-19, 2012, *St. Petersburg, Russia*. pp. 270-285.

[9] Official website of SecFutur project. http://www.secfutur.eu/, last visited on 22.02.2017.

[10] I.V. Kotenko, D.S. Levshun, A.A. Chechulin, “Event correlation in the integrated cyber-physical security system,” *Proceedings of the 2016 XIX IEEE International Conference on Soft Computing and Measurements (SCM-2016), IEEE, St. Petersburg, Russia*, May 2016. pp. 484-486.