Enhancing Integrity Protection for Industrial Cyber Physical Systems

Rainer Falk and Steffen Fries
Corporate Technology
Siemens AG
Munich, Germany
e-mail: {rainer.falk|steffen.fries}@siemens.com

***Abstract*—Cyber physical systems are technical systems that are operated and controlled using information technology. Protecting the integrity of cyber physical systems is a highly important security objective to ensure the correct and reliable operation and to ensure high availability. A comprehensive protection concept of the system integrity involves several axes: the component level ranging from sensors/actuator devices up to control and supervisory systems, planning and configuration management, and the system life cycle. It allows detecting integrity violations on system level reliably by analyzing integrity measurements from a multitude of independent integrity sensors, capturing and analyzing integrity measurements of the physical world, on the field level, and of control and supervisory systems.**

Keywords–system integrity, device integrity; cyber physical systems; Internet of Things, embedded security; cyber security.

I. INTRODUCTION

With ubiquitous machine-oriented communication, e.g., the Internet of Things and interconnected cyber physical systems (CPS), the integrity of technical systems is becoming an increasingly important security objective. Information technology (IT) security mechanisms have been known for many years, and are applied in smart devices (Internet of Things, Cyber Physical Systems, industrial and energy automation systems, operation technology) [1]. Such mechanisms target authentication, system and communication integrity and confidentiality of data in transit or at rest. System integrity takes a broader approach where not only the integrity of individual components (device integrity) and of communication is addressed, but where integrity shall be ensured at the system level of interconnected devices. This purpose is in particular challenging for dynamically changing cyber physical systems, that come with the industrial Internet of Things (IIoT) and Industrie 4.0. Cyber systems will become more open and dynamic to support flexible production down to lot size 1 (plug-and-work reconfiguration of manufacturing equipment), and flexible adaptation to changing needs (market demand, individualized products).

The flexibility starts on the device level where smart devices allow for upgrading and enhancing device functionality by downloadable apps. But also the system of interconnected machines is reconfigured according to changing needs.

Classical approaches for protecting device and system integrity target at preventing any changes, and compare the current configuration to a fixed reference policy. More flexible approaches are needed to protect integrity for flexibly reconfigurable and self-adapting CPSs.

This paper describes an integrated, holistic approach for ensuring CPS integrity. After summarizing system security requirements coming from relevant industrial security standard IEC62443 [1] in Section II, an overview for protecting device integrity and system integrity is described in Sections III and IV. The presented approach for integrity monitoring is an extensible framework to include integrity information from IT-based functions and the physical world of a CPS. This allows integrating integrity information from the digital and the physical world. A new approach for integrity monitoring of encrypted communications is described in Section V. An approach for evaluation in an operational security management setting is outlined in Section VI. Related work is summarized in Section VII, and Section VIII concludes the paper.

II. SYSTEM INTEGRITY REQUIREMENTS

A. Overview IEC62443 Industrial Security Standard

The international industrial security standard IEC 62443 is a security requirements framework defined by the International Electrotechnical Commission (IEC) and can be applied to different automation domains, including energy automation, process automation, building automation, and others. The standard has been created to address the specific requirements of industrial automation and control systems. It covers both organizational and technical aspects of security. In the set of corresponding documents, security requirements are defined, which target the solution operator and the integrator but also the product vendor.

As shown in Figure 1, different parts of the standard are grouped into four clusters covering

– common definitions and metrics;
– requirements on setup of a security organization (ISMS related, comparable to ISO 27001 [2]), as well as solution supplier and service provider processes;
– technical requirements and methodology for security on system-wide level, and

 ISBN: 978-1-61208-605-7

– requirements on the secure development lifecycle of system components, and security requirements to such components at a technical level.

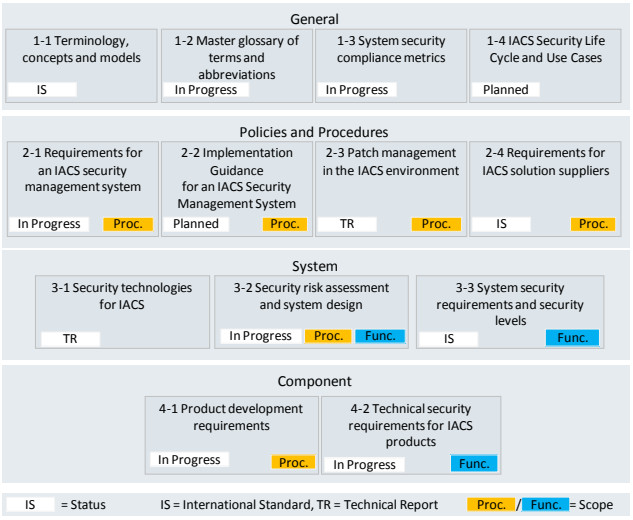

Figure 1. IEC 62443 Overview and Status

According to the methodology described in IEC 62443-3-2, a complex automation system is structured into zones that are connected by and communicate through so-called "conduits" that map for example to the logical network protocol communication between two zones. Moreover, this document defines Security Levels (SL) that correlate with the strength of a potential adversary as shown in Figure 2 below. To reach a dedicated SL, the defined requirements have to be fulfilled.

4 Security Level (SL)	
SL 1	Protection against **casual or coincidental** violation
SL 2	Protection against **intentional violation** using **simple means** with low resources, generic skills and low motivation
SL 3	Protection against intentional violation using **sophisticated means** with **moderate resources**, IACS specific skills and moderate motivation
SL 4	Protection against intentional violation using sophisticated means with **extended resources**, IACS specific skills and high motivation

Figure 2. IEC 62443 defined Security Level

B. IEC62443 Integrity Requirements

Part 3.3 of IEC62443 [3] defines seven foundational security requirements, including a specific foundational requirement on integrity.

IEC 62443 part 3-3 defines seven foundational requirements group specific requirements of a certain category:

– FR 1 Identification and authentication control
– FR 2 Use control
– FR 3 System integrity
– FR 4 Data confidentiality
– FR 5 Restricted data flow
– FR 6 Timely response to events
– FR 7 Resource availability

For each of the foundational requirements there exist several concrete technical security requirements (SR) and requirement enhancements (RE) to address a specific security level. In the context of communication security, these security levels are specifically interesting for the conduits connecting different zones.

Integrity requirements cover in particular the following areas:

– Overall system integrity
– Communication integrity
– Device integrity

The following examples from IEC 62443-3-3 [3] illustrate some of the foundational requirements:

– FR3, SR3.1 Communication integrity: "The control system shall provide the capability to protect the integrity of transmitted information".
– FR3, SR3.4 Software and information integrity: "The control system shall provide the capability to detect, record, report and protect against unauthorized changes to software and information at rest."
– FR3, SR3.8 Session integrity: "The control system shall provide the capability to protect the integrity of sessions. The control system shall reject any usage of invalid session IDs."
– FR5, SR 5.2 Zone boundary protection: "The control system shall provide the capability to monitor and control communications at zone boundaries to enforce the compartmentalization defined in the risk -based zones and conduits model."

III. PROTECTING DEVICE INTEGRITY

The objective of device integrity is to ensure that a device is not manipulated in an unauthorized way. This includes the integrity of the device firmware, of the device configuration, but also the physical integrity. Main technologies to protect device integrity are:

– Secure boot: A device loads at start-up only unmodified, authorized firmware.
– Measured boot: The loaded software modules are checked when they are loaded. Usually, a cryptographic hash value is recorded in a platform configuration register of a hardware of firmware trusted platform module (TPM) [4][5]. The configuration information can be used to grant access to keys, or it can be attested towards thirds parties.

 ISBN: 978-1-61208-605-7

- Protected firmware update: When the firmware of a device is updated, the integrity and authenticity of the firmware update is checked. The firmware update image can be digitally signed.
- Runtime integrity checks: During operation, the device performs self-test of security functionality and integrity checks to verify whether it is operating as expected. Integrity checks can verify the integrity of files, configuration data, software modules, and or runtime data as process list.
- Process isolation, kernel-based mandatory access control (MAC): Hypervisors or kernel MAC systems like SELinux [6], AppArmor [7], or SMACK [8], can be used to isolate different classes of software (security domains). An attack or malfunction one security domain does not affect other security domains on the same device.
- Tamper evidence, tamper protection: The physical integrity of a device can be protected, e.g., by security seals or by tamper sensors that detect opening or manipulation of the housing.
- Device integrity self test: A device performs a self-test to detect failures. The self-test is performed typically during startup and is repeated regularly during operation. Operation integrity checks: measurements on the device can be compared with the expected behavior in the operative environment. An example is the measurement of connection attempts to/from the device. Based on a Management Information Base (MIB) setting.

The functionality of some devices can be extended by extensions (App). Here, the device integrity has to cover also the App runtime environment: Only authorized, approved apps can be downloaded and installed. Apps are isolated during execution (managed runtime environment, hypervisor, container)

The known approaches to protect device integrity focus on the IT-related functionality of a device (with the exception of tamper protection). Also, a strong tamper protection is not common on device level. The main protection objective for device integrity shall ensure that the device's control functionality operates as designed. However, the integrity of input/output interfaces, sensors, and actuators are typically out of scope. In typical industrial environments, applying a strong tamper protection to the each control device, sensor, and actuator would not be economically feasible. So, protecting device integrity alone would be too limited to achieve the goal of protection the integrity of an overall CPS.

IV. System Integrity Monitoring

The next level of integrity is on the system level comprising a set of interconnected devices. The main approaches to protect system integrity are collecting and analyzing information on system level:

- Device inventory: Complete and up-to-date list of installed devices (including manufacturer, model, serial number version, firmware version, current configuration, installed software components, location)Centralized Logging: Devices provide logdata, e.g., using Open Platform Communication Unified Architecture (OPC UA) protocol [9], SNMP [10], or syslog protocol [11], to a centralized logging system.
- Runtime device integrity measurements: A device integrity agent provides information gathered during the operation of the device. It collects integrity information on the device and provides it for further analysis. Basic integrity information are the results of a device self-test, and information on the current device configuration (firmware version, patches, installed applications, configuration). Furthermore, runtime information can be gathered and provided for analysis (e.g., process list, file system integrity check values, partial copy of memory).
- Network monitoring: The network communication is intercepted, e.g., using a network tap or a mirror port of a network switch. A challenge is the fact that network communication is increasingly encrypted.
- Physical Automation process monitoring: Trusted sensors provide information on the physical world that can be used to cross-check the view of the control system on the physical world.
- Physical world integrity: trusted sensors (of physical world). Integrated monitoring of embedded devices and IT-based control systems, and of the technical process. Allow now quality of integrity monitoring as physical world and IT world are checked together.

The captured integrity information can be used for runtime integrity monitoring to detect integrity violations in real-time. Operators can be informed, or actions can be triggered automatically. Furthermore, the information is archived for later investigations. So, integrity violations can probably be detected later, so that corresponding counter-measures can be initiated (e.g., plan for an additional quality check of produced goods). The integrity information can be integrated in or linked to data of a production management system, so that it can be investigated under which integrity conditions certain production steps have been performed. Product data is enhanced with integrity monitoring data related to the production of the product.

A. System Overview

Agents on the system components acting as integrity sensors collect integrity information and optionally determine an integrity attestation of the collected information. To allow for flexibility in CPS, the approach puts more focus on monitoring integrity and acting when integrity violations are detected, than on preventing any change that has not been pre-approved by a static policy.

The approach is based on integrity sensors that provide integrity related measurements. An intelligent analysis

 ISBN: 978-1-61208-605-7

platform analysis this data using data analysis (e.g., statistical analysis, big data analysis, artificial intelligence) and to trigger suitable respondence actions (e.g., alarm, remote wipe of a device, revocation of a device, stop of a production site, planning for additional test of manufactured goods).

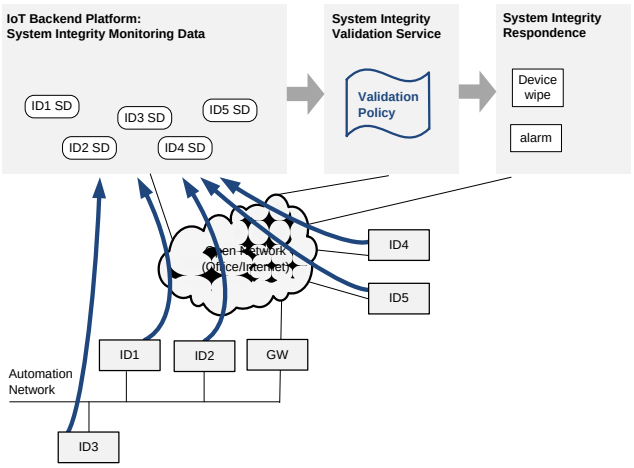

Figure 3. Validation of Device Monitoring Data

Figure 3 shows an example for an IoT system with IoT devices (ID1, ID2, etc.) that communicate with an IoT backend platform. The devices provide current integrity monitoring information to the backend platform. The devices can be automation devices that include integrity measurement functionality, or dedicated integrity sensor devices. The device monitoring system itself has to be protected against attacks itself, following IEC62443.

An integrity data validation service checks the obtained integrity measurement data for validity using a configurable validation policy. If a policy violation is detected, a corrective action is triggered: For example, an alarm message can be displayed on a dash board. Furthermore, an alarm message can be sent to the IoT backend platform to terminate the communication session of the affected IoT device. Moreover, the device security service can be informed so that it can revoke the devices access permissions, or revoke the device authentication credential.

B. Integrity Sensors

The integrity monitoring framework foresees to include a variety of integrity measurements. Depending on the specific application scenario, meaningful integrity sensors can be deployed. Depending on the evolving needs, additional sensors can be deployed as needed.

- Physical world (technical process)
- Physical world (alarm systems, access control systems, physical security as, e.g., video surveillance)
- Device world (malware, device configuration, firmware integrity)IT-based control systems (local, clod services, edge cloud)
- Infrastructure (communication networks)

Flexible extension with additional integrity sensors (even very sophisticated as, e.g., monitoring power fingerprint). The described approach is open to develop and realize sophisticated integrity measurement sensors. So the solution is design to allow evolution and innovation. Integrity sensors have to be protected against attacks so that they provide integrity measurements reliably.

C. Integrity Verification

The integrity monitoring events are analyzed using known data analysis tools. The system integrity can be monitored both online. In industrial environments, it is also important to have reliable information about the system integrity of a production system for the time period during which a certain production batch was performed. This allows performing the verification also afterwards to check whether during a past production batch integrity-violations occurred.

The final decision whether a certain configuration is accepted as correct is up to human operators. After reconfiguration, or for a production step, the configuration is to be approved. The approval decision can be automated according to previously accepted decisions, or preconfigured good configurations).

As integrity measurements are collected from a multitude of integrity sensors, integrity attacks can be detected reliably. Even if some integrity sensors should be disabled or manipulated to provide malicious integrity measurements, still other integrity sensors can provide integrity information that allows detecting the integrity violation. Checking integrity using measurements from independent integrity sensors and on different levels (physical level, field devices, control and supervisory systems) allows detecting integrity violations by checking for inconsistencies between independent integrity measurements.

V. INTEGRITY MONITORING OF ENCRYPTED COMMUNICATIONS

A specific part of monitoring the system integrity is the network communication. However, network communication is encrypted more-and-more, e.g., using the Transport Layer Security (TLS) protocol [12]. In contrast to earlier versions of the TLS protocol, the most recent version TLS1.3 [13], currently under development, supports only cipher-suites realizing authenticated encryption. Both confidentiality and integrity/authenticity of user communication is protected. No cipher suite providing integrity-only protection is supported by TLS version 1.3, anymore. So, only basic IP header data can be analyzed. This is not sufficient for integrity monitoring of TLS-protected industrial control communication.

A protocol specific solution to enable monitoring of encrypted communication channels by trusted middleboxes is provided by mcTLS [14]. With mcTLS, trusted middleboxes can be incorporated into a secure sessions established between a TLS Client and a TLS Server. Figure 4 shows the basic principle of mcTLS. A TLS authentication and key agreement is performed between a TLS client and a TLS server. As part of the handshake, the TLS client indicates those TLS middleboxes that shall be incorporated

 ISBN: 978-1-61208-605-7

within the TLS session. After the authentication and key between client and server has been completed, both the TLS client and the TLS server send (encrypted) key material of the established TLS session to the middleboxes. This allows the middleboxes to decrypt the data exchanged between TLS client and TLS server. Note that the integrity of the data exchange is cryptographically protected by message authentication codes. The keys for integrity protection are not made available to the middleboxes, so that the middleboxes can only decrypt the data, but not interfere with the contents of the data. So, data integrity is ensured end-to-end although middleboxes can decrypt the data.

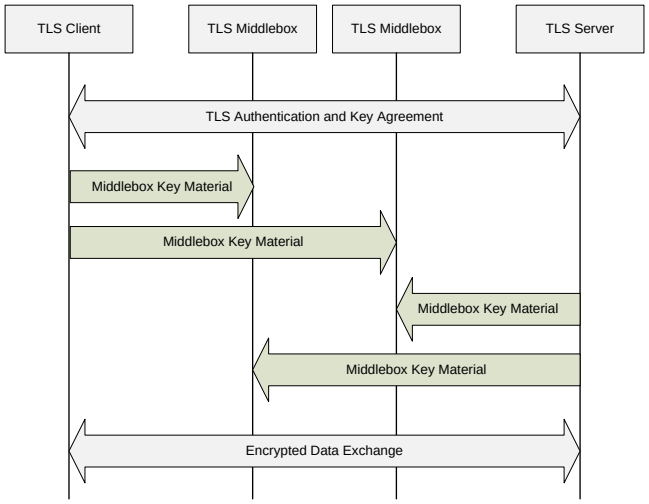

Figure 4. Multi-Context TLS

The basic principle is to perform an enhanced handshake involving middleboxes into the handshake phase of TLS, see Figure 4. Specifically, the middleboxes are authenticated during the handshake and thus known to both communicating ends. Moreover, each side is involved in the generation of the session key, which is also provided to the middlebox. There is also additional keying performed for the exchange of pure end-to-end keys. Specific key material known to the middlebox is used to decrypt the traffic and check the integrity. The end-to-end based keys are used to protect integrity end-to-end. The latter approach ensures that the middlebox can only read and analyze the content of the communication in the TLS record layer, but any change done by the middlebox is detected by an invalid end-to-end integrity check value. This approach has the advantage that it provides an option to check the associated security policy during the session setup and at the same time monitor traffic as an authorized component. The drawback is that the solution focuses solely on TLS and cannot be applied to other protocols without changes.

The TLS-variant mcTLS allows middleboxes to analyze the TLS-protected communication, e.g., to detect potential security breaches This approach enables communication checking the contents of the communication session without breaking end-to-end security. So, with mcTLS, the contents of encrypted data communication, in particular of industrial control communication, can be checked.

VI. EVALUATION

The security of a cyber system can be evaluated in practice in various approaches and stages of the system's lifecycle:

– Threat and risk analysis (TRA) of cyber system
– Checks during operation to determine key performance indicators (e.g., check for compliance of device configurations).
– Security testing (penetration testing)

During the design phase of a cyber system, the security demand is determined, and the appropriateness of a security design is validated using a threat and risk analysis. Assets to be protected and possible threats are identified, and the risk is evaluated in a qualitative way depending on probability and impact of threats. The effectiveness of the proposed enhanced device authentication means can be reflected in a system TRA.

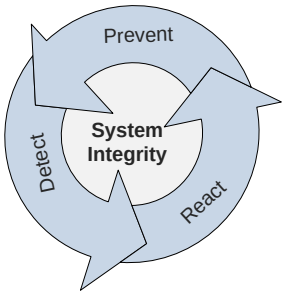

Figure 5. Prevent Detect React Cycle

So, the main evaluation of security tools is coming during security operation, when as part of an overall operational security management appropriate technologies are chose that, in combination, reduce the risk to an acceptable level.

The new approach presented in this paper provides an additional component of a security architecture that reduces the risk of integrity violations. Compared to existing solutions covering IT-related aspects only, the integrity of the control application and the physical world are included. The solution approach does not intend to have a single technology, but it realizes a system-oriented approach that can evolve as part of the security management life cycle covering prevent, detect, and response, see also Figure 5.

VII. RELATED WORK

A security operation center (SOC) is a centralized unit for detecting and handling security incidents. Main functionalities are continued security monitoring reporting, and post-incident analysis [15][16]. Security incident and Event management (SIEM) systems can be used within a SOC to analyze security monitoring data. Compliance management systems support a centralized reporting of server configuration in data centers.

Host-based intrusion detection systems (HIDS) as SAMHAIN [17] and OSSEC [18] analyze the integrity of

 ISBN: 978-1-61208-605-7

hosts and report the results to a backend security monitoring system. Network based intrusion detection systems (NIDS) capture the network traffic, e.g., using a network tap or a mirroring port of a network switch, and analyze the traffic. Examples are SNORT [19] and Suricata [20].

Two main strategies can be followed by an intrusion detection system (IDS): Known malicious activities can be looked for (signature based detection), or any change compared to a learned reference network policy is detected (anomaly detection).

An "automotive thin profile" of the Trusted Platform Module TPM 2.0 has been specified [21]. A vehicle is composed of multiple control units that are equipped with TPMs. A rich TPM manages a set of thin TPMs, so that the vehicle can be represented by a vehicle TPM to the external world. The vehicle's rich TPM can check the integrity of the vehicle by verifying attestations provided by thin TPMs.

Approaches to utilize the context information on the CPS operation, device capabilities, device context to enhance the authentication of a single device, have been described by the authors of this paper in previous work [22]. The effect of an integrity attack on the degradation of a control system has been investigated by Mo and Sinopoli [23].

VIII. CONCLUSION

Ensuring system integrity is an essential security feature for cyber physical systems and the Internet of Things. The security design principle of "defense in depth" basically means that multiple layers of defenses are designed. This design principle can not only be applied at the system level, but also at the level of a single security mechanism.

This paper proposed a framework for ensuring system integrity in flexibly adaptable cyber physical systems. With new concepts for flexible automation systems coming with Industrial IoT / Industrie 4.0, the focus of system integrity has to move from preventing changes to device and system configuration in having transparency on the device and system configuration and checking it for compliance. This paper focused on integrity of devices, communication, and cyber systems. Integrity in a broader sense covers the whole life cycle, including development, secure procurement, secure manufacturing, and supply chain security.

REFERENCES

[1] IEC 62443, "Industrial Automation and Control System Security" (formerly ISA99), available from: http://isa99.isa.org/Documents/Forms/AllItems.aspx 2017.09.26

[2] ISO/IEC 27001, "Information technology – Security techniques – Information security management systems – Requirements", October 2013, available from: https://www.iso.org/standard/54534.html 2017.09.26

[3] IEC62443-3-3:2013, "Industrial communication networks – Network and system security – Part 3-3: System security requirements and security levels", Edition 1.0, August 2013

[4] Trusted Computing Group: "TPM Main Specification", Version 1.2, , available from http://www.trustedcomputinggroup.org/resources/tpm_main_specification 2017.09.26

[5] Trusted Computing Group, "Trusted Platform Module Library Specification, Family 2.0", 2014, available from http://www.trustedcomputinggroup.org/resources/tpm_library_specification 2017.09.26

[6] SELinux, "Security Enhanced Linux", available online: https://selinuxproject.org/page/Main_Page 2017.09.26

[7] AppArmor, "AppArmor Security Project", available online: http://wiki.apparmor.net/index.php/Main_Page 2017.09.26

[8] SMACK, "Simplified Mandatory Access Control Kernel", available online: https://www.kernel.org/doc/html/latest/admin-guide/LSM/Smack.html 2017.09.26

[9] OPC Foundation, "OPC Unified Architecture (UA)", available online: https://opcfoundation.org/about/opc-technologies/opc-ua/ 2017.09.26

[10] J. Case, R. Mundy, et al., "Introduction and Applicability Statements for Internet Standard Management Framework", RFC3410, available online: https://tools.ietf.org/html/rfc3410 2017.09.26

[11] R. Gerhards, "The Syslog Protocol", RFC5424, March 2009, available online: https://tools.ietf.org/html/rfc5424 2017.09.26

[12] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, Aug. 2008, available from http://tools.ietf.org/html/rfc5246 2017.09.26

[13] E. Rescorla: "The Transport Layer Security (TLS) Protocol Version 1.3", Internet draft (work in progres), September 2017, available online: https://tlswg.github.io/tls13-spec/draft-ietf-tls-tls13.html 2017.09.26

[14] D. Naylor, K. Schomp, et al., "Multi-Context TLS (mcTLS), Enabling Secure In-Network Functionality in TLS," available from http://mctls.org/ 2017.09.26

[15] B. Rothke, "Building a Security Operations Center (SoC)", RSA Conference, 2012, available from https://www.rsaconference.com/writable/presentations/file_upload/tech-203.pdf 2017.09.26

[16] McAffee Foundstone® Professional Services, "Creating and Maintaining a SoC", Intel Security Whitepaper, available from: https://www.mcafee.com/us/resources/white-papers/foundstone/wp-creating-maintaining-soc.pdf 2017.09.26

[17] R. Wichmann, "The Samhain HIDS", fact sheet, 2011, available from http://la-samhna.de/samhain/samhain_leaf.pdf 2017.09.26

[18] OSSEC, "Open Source HIDS SECurity", web site, 2010 - 2015, available from http://ossec.github.io/ 2017.09.26

[19] "SNORT", web site, available from https://www.snort.org/ 2017.09.26

[20] "Suricata", web site, available from https://suricata-ids.org/ 2017.09.26

[21] Trusted Computing Group, "TCG TPM 2.0 Automotive Thin Profile", level 00, version 1.0, 2015, available from http://www.trustedcomputinggroup.org/resources/tcg_tpm_20_library_profile_for_automotivethin 2017.09.26

[22] R. Falk and S. Fries, "Advanced Device Authentication: Bringing Multi-Factor Authentication and Continuous Authentication to the Internet of Things", The First International Conference on Advances in Cyber-Technologies and Cyber-Systems, CYBER 2016, October 9 - 13, 2016 - Venice, Italy , available from http://www.thinkmind.org/index.php?view=article&articleid=cyber_2016_4_20_80029 2017.09.26

[23] Y. Mo and B. Sinopoli, "On the Performance Degradation of Cyber-Physical Systems Under Stealthy Integrity Attacks", IEEE Transactions on Automatic Control 61.9 (2016): 2618-2624.

 ISBN: 978-1-61208-605-7