A Reference Architecture for Integrating Safety and Security Applications on Railway Command and Control Systems

Extended Abstract

Henk Birkholz, Christoph Krauß, Maria Zhdanova
Fraunhofer Institute for Secure Information Technology SIT
Darmstadt, Germany
{firstname.lastname}@sit.fraunhofer.de

Don Kuzhiyelil
SYSGO AG
Klein-Winternheim, Germany
don.kuzhiyelil@sysgo.com

Tolga Arul, Markus Heinrich, Stefan Katzenbeisser, Neeraj Suri, Tsvetoslava Vateva-Gurova
TU Darmstadt
Darmstadt, Germany
{arul,heinrich,katzenbeisser}@seceng.informatik.tu-darmstadt.de,{suri,vateva}@deeds.informatik.tu-darmstadt.de

Christian Schlehuber
DB Netz AG
Frankfurt am Main, Germany
christian.schlehuber@deutschebahn.com

KEYWORDS

Safety, Security, MILS, Railway Command and Control Systems

ACM Reference Format:
Henk Birkholz, Christoph Krauß, Maria Zhdanova, Don Kuzhiyelil, Tolga Arul, Markus Heinrich, Stefan Katzenbeisser, Neeraj Suri, Tsvetoslava Vateva-Gurova, and Christian Schlehuber. 2018. A Reference Architecture for Integrating Safety and Security Applications on Railway Command and Control Systems: Extended Abstract. In *Proceedings of 4th International Workshop on MILS: Architecture and Assurance for Secure Systems (MILS'18).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In critical infrastructures such as railway systems, the continuous and resilient availability of safety critical functions residing on actuator and sensor components must be ensured. Since these components are also more and more connected using the Internet Protocol (IP), they additionally require security functions to provide protection against attackers. Moreover, the railway infrastructure is highly distributed, with its critical components residing at the track side easily accessible to attackers. Thus, a continuous proofing that the safety-critical systems are not manipulated is required, too.

The (safety) certification of such safety-critical systems covers both the hardware components and corresponding software components that compose a specific safety-critical application. Since security functions are currently not in use, they are not part of the certification. However, the integration of security functions is imperative to provide the basis for preventing or detecting manipulations of the system. In essence, co-residing security functions are required to retain and assure the trusted interoperability of safety critical systems integrated in the rapidly growing number of newly deployed control networks based on the IP. Thus, it is required that a given safety certification (and the given guarantees) must not be violated by the integration of security functions.

In this paper, we present the first results of the ongoing HASELNUSS project[1] by introducing the Haselnuss Reference Architecture (HRA) for Railway Command and Control Systems (CCS), that allows uncertified security functions to reside on the same hardware device as certified safety functions; without voiding the certification of these safety functions.

2 ARCHITECTURE OF RAILWAY SYSTEMS

Control and safety systems take a central role in the safe operation of trains in European rail networks since a long time. In the early days, around 1900, the safety of trains was ensured by the usage of mechanical interlocking systems. Since then the interlocking systems have experienced a steady evolution, which resulted in the current electronic interlocking system which are computerized systems implementing the the safety logic of the interlocking. As a part of this evolution, also the general architecture and behavior of the interlocking systems evolved; while only a minimum of interaction with external systems was required in the beginning, modern electronic interlocking systems or operating control centers are connected to a wide variety of systems. Partly also public communication links are used for these connections.

Current signaling systems can in general be divided into three layers:

Operation Layer On this layer, the operators are working at specialized workstations and tell the interlocking system, which route has to be built and in which direction the trains

MILS'18, June 2018, Luxembourg
2018. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] https://haselnuss-projekt.de/

have to move. This is done via the workstations, which consist of a safe display of the controlled area. These workstations are also connected to several communication systems like GSM-R or the telephone network. Besides the operators, also SOC/NOC systems as well as disposition systems are located. The buildings where these systems are located have to fulfill special requirements regarding physical security. Also the personnel is trained to perform a safe railway operation.

Interlocking Layer On the interlocking layer, most of the safety systems are located. The layer is located between the operation layer and the field element layer and checks the commands from the operation layer for validity and if they respect the safe operation. In addition, it monitors the components on the field element layer for correct operation and in case of anomalies, falls into an error state. On this layer, systems like the interlocking itself and the European Train Control System (ETCS) are located. The layer is connected to the operation layer and the field element layer via a wide area network owned or leased by the railway operator. Components on this layer are developed according to several safety standards like EN 50126 [2] and only the required functionality is available. Additionally, these components are built redundant, which means that in case of a defect one of the standby systems comes in place and the maintenance personnel is notified to replace the failing component. The data networks and the power supplies are redundant, too. According to the size of the facility, the building is equipped with a battery or also a generator, which is started if the energy provider is not able to provide sufficient energy.

Field Element Layer On this layer, the field elements are located. These are elements like points, signals, axle counters, or other equipment of this type. Each element is controlled by an object controller, which is connected to the interlocking layer via a network connection. The communication between the layers is secured by a security gateway that applies integrity-protection, encryption, etc. For the key distribution a Public Key Infrastructure (PKI) is in place.

An exemplary illustration of such an architecture can be seen in Figure 1.

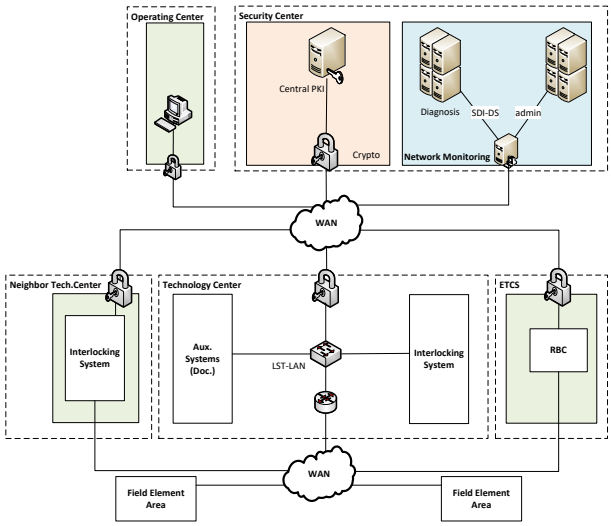

Figure 1: Exemplary Architecture of Signaling Systems [10]

3 SECURITY GOALS

Introduction of networked IT-based components and IP networks into formerly closed railway infrastructures changes their risk landscape. In emerging CCS architectures such as the one shown in Figure 1 hazardous situations can result from random hardware faults and software bugs or be caused by actions of a malicious attacker. For this reason, security goals in addition to safety goals have to be considered during system design.

For the HRA, we define the following security goals [6]:

- *Availability:* A railway CCS should at any time be able to provide its required functionality and data, i.e., to generate safe routes, send and receive signals and commands over the network, log critical events, etc. This requires provisions against Denial-of-Service (DoS) attacks that can be carried out on a network or a cyber-physical layer and block or delay time-critical operations. In safety systems, availability guarantees are usually achieved through redundancy. In case of a motivated attacker, this might not be enough, especially if redundant components "fail-safe" silently, and the attack stays undetected until the system limit is reached.
- *Integrity:* Considering that a railway CCS is a highly distributed and complex system, it requires the protection against any unauthorized modification of its data (configurations, commands, access credentials, etc.) as well as software and hardware components. If such modifications stay undetected, the correct operation of the CCS can be disrupted in multiple ways.
- *Authenticity:* It is necessary to be able to verify the trusted origin of safety- and security-critical data and components in order to prevent, e.g., that tampered software or hardware is deployed in a railway CCS or reactions build on the falsified information.
- *Confidentiality:* Data transferred by safety applications in a CCS are not considered confidential. Apart from safety assets, the electronic interlocking system architecture contains security assets such as access credentials or cryptographic keys for the PKI that need to be protected from unauthorized disclosure or use.
- *Accountability:* Any action performed by a CCS should be traceable to an authorized entity responsible for this action.
- *Non-repudiation:* An authorized entity in a CCS should not be able to deny its actions.
- *Auditability:* Security-related events need to be recorded in an auditable form (including time, source, user, etc.).

4 HASELNUSS REFERENCE ARCHITECTURE

The Haselnuss Reference Architecture (HRA) can be integrated in railway systems such as object controllers for field elements, we call them Haselnuss nodes.

For the integration of safety and security applications on the same hardware platform, a certifiable MILS (Multiple Independent Levels of Safety and Security) operating system or a separation kernel (SK) [12] is used. By making use of the separation capabilities of the SK [8], the existing safety application is spatially and temporally separated from the newly introduced security applications. Spatial separation is required to ensure that the security application will not affect the integrity of safety application's code and data. Temporal separation is required to ensure that the temporal behavior of safety application is not affected by the security applications and thus, not influencing the real-time guarantees to be fulfilled by the safety application. Of course, the hardware platform used for the integration shall be fast enough to reserve the CPU time required for the safety application to meet its deadlines and at the same time have remaining CPU time that can be made available to the security applications to perform its functions.

Information channels to the safety application are realized making use of the communication objects provided by the SK that allows precise control over the information flows in the system. This partitioned architecture based on the certifiable SK allows to provide evidence of non-interference between the high assurance safety applications (i.e., Safety Integrity Level (SIL) 4) and the security applications which does not contribute to the safety of the system (and thus having a lower SIL). This freedom from interference evidence is needed to keep the existing certification of safety application when integrated with the security applications. The SK that we use is also certifiable at the same assurance levels (i.e., SIL 4) as the safety application, e.g., an railway object controller[2].

The MILS template is used to run the critical infrastructure's safety application(s) on the same hardware as the security applications that are created to protect the safety functionality against attacks. MILS's separation allows us to define exact contact points of information flow between the safety application and the security application such as an Intrusion Detection System (IDS). This structures the safety case, where the influence of security has to be investigated and freedom of interference with the safety has to be proven.

The HRA provides several security functions. This includes, amongst others, mutual authentication of Haselnuss nodes with the interlocking system, protecting the software integrity of Haselnuss nodes at boot- and runtime, integrity reporting and remote attestation of Haselnuss nodes, remote software update of Haselnuss nodes, and an IDS.

There are vital transition points in the life-cycle of a safety-critical system and one prominent example (next to on-boarding or enrollment) are software updates. Three types of information elements, composed into manifests, increase the trust in a software update significantly: the integrity evidence created by the device to be updated (the Attestor) before the update, proving that the safety-critical device is in a state that warrants the deployment of potentially confidential information as part of an software update. Evidence about the acquisition of a signed manifest of a software update, a composite of the firmware or the pointer to a trusted source of firmware and corresponding metadata [7], created also by the Attestor. The integrity evidence of the new operational state of the safety-critical device that just completed the transition procedure of a software update. Appropriated measures can be selected with an significantly improved confidence, if these key performance indicators about the targets integrity can be provided to the owner or maintainer of safety-critical infrastructure.

The integrity and remote attestation function provides a continuous proofing function of the platform integrity. It currently includes the integration of a secure boot process and a time-based uni-directional attestation (TUDA) [4] procedure. The TUDA protocol defined in the HRA utilizes the Trusted Platform Module (TPM) version 2.0, a Hardware Security Module (HSM) specified by the Trusted Computing Group (TCG) [11]. The TUDA protocol is also used to illustrate the complete continuous proofing work-flow from creating integrity evidence (Attestor role), streaming it to a management system, and appraising the evidence (Verifier role) to confirm the integrity of software components. In this proof-of-concept, the implementation of TUDA is used to provide and assess integrity evidence both for security functions and safety functions via an integrated solution. Since TUDA only provides an assurance of the system's software integrity at boot time, a health monitoring functionality complements this security function with runtime integrity monitoring.

The functional architecture of the Health Monitor includes components for non-invasive data collection, runtime analysis of applications (if the integrity state has changed) and reporting. The ultimate goal is to make a railway object controller resilient against malicious attacks that cannot be detected or prevented using boot-time mechanisms. This can be achieved using a hypevisor-based monitoring techniques that analyze the integrity state of applications and detect anomalies [1, 9]. For this purpose, amongst other state parameters, the information channels to and from safety applications are controlled via new security applications adding the desired security properties to the currently implemented availability measures, without impact on the assurance level of the safety applications. The communication objects of the SK are extended with monitoring capabilities. In addition, system services are monitored to detect failures or attacks. The capability of systems to recover after failures or attacks can help reducing the service downtimes. For example, depending on a particular scenario, it may be desirable for an application in case of a failed configuration or code update to automatically resume to an older software version instead of failing or trigger some other recovery mechanisms. In this regard, the applicability of approaches from adaptive systems similar to ones proposed in [3, 5] is analyzed for HRA.

The IDS provides a defense mechanism against network-based attacks. By collaborating among the HRA instances in a defined area of the critical infrastructure, the IDS is enabled to detect adverse commands and configurations of the infrastructure's actuator and sensor components. It includes a concept to fine-tune the IDS on the critical infrastructure's network topology and utilized protocols. In this way, the IDS is enabled to leverage context information of the controlled infrastructure to enhance the intrusion detection accuracy. In a second step, counteractions on detected intrusions are defined that respect the safety functions of the critical infrastructure. We carefully design the intrusive counteractions such that they do not alter the network channel properties beyond the specification that the safety application is anyway required to tolerate.

[2] https://www.sysgo.com/solutions/safety-security-certification/

For example, in case of a connection loss or temporary network breakdown, the system is fail-safe already. We plan to evaluate the extent of this interference in a test-bed that will be built during the project.

In addition to these functions, the system itself is hardened against attacks. Due to the co-location of the safety application with other services, mechanisms to prevent violations of the confidentiality of the system through a covert- or side-channel attack stemming from the usage of shared resources (e.g., [14] and [13]) will be considered. These mechanisms will consider the probability of a side-channel attacker process co-residing with the safety and security applications that can exploit the cache of the underlying system to leak information. The underlying SK of the HRA already employs measures to prevent cache covert-channel and side-channel exploits. As a defense mechanism, the cache can be flushed at every context switch when involving an application that deals with confidential data. This mechanism makes the cache unavailable as a covert channel in respect to the particular application. Such an approach erases the cache footprint left by the application and with that eradicates the cache-based covert-channel threat. Moreover, the proposed HRA ensures that the co-residency of compartments is established statically and cannot be changed during runtime which reduces the probability of a malicious co-resident. HRA does not make use of the memory de-duplication feature for better memory utilization eliminating a class of side-channel attacks based on the Flush+Reload strategy relying on memory de-duplication.

5 CONCLUSION AND FUTURE WORK

Safety and security are historically two different and isolated worlds. Safety certification, especially in the railway sector, does not consider security measures. Moreover, one of the greatest challenges of the ongoing railway digitalization, is how to guarantee the transportation safety of the new IT-based railway systems now open to malicious attackers? Can state-of-the-art IT components and IP networks be used in railway scenarios to increase efficiency but without putting people's lives at risk? The proposed Haselnuss Reference Architecture tries to answer this question positively, equipping safety applications with the necessary security functions.

As a work in progress, the Haselnuss Reference Architecture first needs to be fully specified, including handling of system upgrades or security incidents, and implemented. It then will be tested in a realistic railway test-bed being currently built as well to analyze the applicability and achievable security of our approach. For example, effects related to timing of safety applications introduced by the SK and additional security components will be analyzed. Certain aspects of HRA will be formally evaluated, too. In addition, the possible certification of our solution will be evaluated together with the responsible authorities, e.g., the German Federal Railway Office (EBA). In this context, the freedom of interference between safety and security will be investigated. If the actions of security are transparent to the safety application, we believe that we can keep the safety-case. This could be possible if the security only affects network channel properties that the safety is already able to tolerate, such as a defined threshold of latency or a certain amount of message loss or channel failure.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their insightful feedback. The work presented in this paper has been partly funded by the German Federal Ministry of Education and Research (BMBF) under the project "HASELNUSS: Hardwarebasierte Sicherheitsplattform für Eisenbahn-Leit- und Sicherungstechnik" (ID 16KIS0597K).

REFERENCES

[1] A. M. Azab, P. Ning, E. C. Sezer, and X. Zhang. 2009. HIMA: A Hypervisor-Based Integrity Measurement Agent. In *Computer Security Applications Conference, 2009. ACSAC '09. Annual.* 461–470. https://doi.org/10.1109/ACSAC.2009.50

[2] CENELEC - European Committee for Electrotechnical Standardization. 2010. *EN50126 - Railway applications - The specification and demonstration of Reliability, Availability, Maintainablity and Safety (RAMS) Part 1: Basic requirements and generic process.* Number EN 50126:1999 E. CENELEC Central Secretariat, rue de Stassart, 36, B-1050 Brussels.

[3] M. Dinkel, S. Stesny, and U. Baumgarten. 2007. Interactive Self-Healing for Black-Box Components in Distributed Embedded Environments. In *Communication in Distributed Systems (KiVS), 2007 ITG-GI Conference.* 1–12.

[4] Andreas Fuchs, Henk Birkholz, Ira McDonald, and Carsten Bormann. 2017. *Time-Based Uni-Directional Attestation.* Internet-Draft draft-birkholz-i2nsf-tuda. The Internet Engineering Task Force (IETF). https://datatracker.ietf.org/doc/draft-birkholz-i2nsf-tuda/

[5] D. Garlan, S. W. Cheng, A. C. Huang, B. Schmerl, and P. Steenkiste. 2004. Rainbow: Architecture-based Self-adaptation with Reusable Infrastructure. *Computer* 37, 10 (Oct 2004), 46–54. https://doi.org/10.1109/MC.2004.175

[6] Markus Heinrich, Tsvetoslava Vateva-Gurova, Henk Birkholz, Maria Zhdanova, Don Kuzhiyelil, and Christian Schlehuber. 2017. *Requirements Analysis.* Deliverable D1. Project "HASELNUSS" (ID 16KIS0597K).

[7] B. Moran, H. Tschofenig, H. Birkholz, and J. Jimenez. 2018. *Firmware Updates for Internet of Things Devices - An Information Model for Manifests.* Internet-Draft draft-ietf-suit-information-model-00. The Internet Engineering Task Force (IETF). https://tools.ietf.org/html/draft-ietf-suit-information-model-00

[8] Sven Nordhoff and Holger Blasum. 2017. *Ease Standard Compliance by Technical Means via MILS.* Zenodo. https://doi.org/10.5281/zenodo.571175

[9] Carbone M. Lee W. Payne, B.D. 2007. Secure and flexible monitoring of virtual machines. In *Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC 2007).*

[10] Christian Schlehuber, Markus Heinrich, Tsvetoslava Vateva-Gurova, Stefan Katzenbeisser, and Neeraj Suri. 2017. A Security Architecture for Railway Signalling. In *International Conference on Computer Safety, Reliability, and Security.* Springer, 320–328. https://doi.org/10.1007/978-3-319-66266-4_21

[11] Trusted Computing Group 2016. *Trusted Platform Module Library Specification* (Family 2.0, Level 00, Revision 01.38 ed.). Trusted Computing Group.

[12] Sergey Tverdyshev. 2017. Security by Design: Introduction to MILS. In *3rd International Workshop on MILS.*

[13] Yuval Yarom and Katrina Falkner. 2014. FLUSH+RELOAD: A High Resolution, Low Noise, L3 Cache Side-channel Attack. In *Proc. of USENIX Security.* 719–732.

[14] Y. Zhang, A. Juels, M. Reiter, and T. Ristenpart. 2012. Cross-VM side channels and their use to extract private keys. In *Proc. of CCS.* 305–316.