OCORA

Open CCS On-board Reference Architecture

(Cyber-) Security Strategy

Gamma Release

Document ID: OCORA-40-010-Gamma

Version: 1.00

Date: 04.12.2020

Status: Final

Revision history

Version	Change Description	Name (Initials)	Date of change
0.01	Initial draft for formal review	Roger Metz	2020-11-02
0.02	Security Cluster 1st review integration	Roger Metz	2020-11-12
1.00	OCORA formal review comments integration	Roger Metz	2020-12-04

Table of Content

References

The following references are used in this document:

[1] OCORA-10-001-Gamma – Release Notes

[2] OCORA-30-001-Gamma – Introduction to OCORA

[3] OCORA-30-002-Gamma – Problem Statements

[4] OCORA-90-002-Gamma – Glossary

[5] OCORA-40-009-Gamma – (Cyber-) Security Overview

[6] EN 50126-1:2017 - Railway Applications – The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS) - Part 1: Generic RAMS Process

[7] EN 50129:2018 - Railway applications -Communication, signalling and processing systems -Safety related electronic systems for signalling

[8] EN 50159:2010 - Railway applications - Communication, signalling and processing systems - Safety-related communication in transmission systems

[9] prTS 50701 - Railway application - Cybersecurity

[10] IEC 62443 3-3 - Industrial communication networks – Network and system security –Part 3-3: System security requirements and security levels

[11] NIST 800-53 - Security and Privacy Controls for Federal Information Systems and Organizations

[12] NIST 800-30 - Guide for Conducting Risk Assessments (July 2002 and September 2012)

[13] ISO 27001 - Information Security Management

[14] ISO 27005 - Information Technology - Security techniques - Information Security Risk Management

1 Management Summary

The security strategy aims to provide security related information like security protection goals, security principles and security guidelines to the OCORA architects and security mangers to ensure the security awareness during the development process.

In the next phase of the OCORA security workstream more architectural information and results will be available to finalize the complete OCORA security documentation for the OCORA Release 1.0.

2 Introduction

2.1 Document Context and Purpose

This document is published as part of the OCORA Gamma release, together with the documents listed in the release notes [1]. It is the first release of this document and it is still in a preliminary state.

Subsequent releases of this document and topic specific documentation will be developed in a modular and iterative approach, evolving within the progress of the OCORA collaboration.

This document aims to provide the reader with Security Strategies.

2.2 Why should I read this Document?

This document addresses experts in the railway security and architecture domain, interested in security strategies.

The reader will be able to provide feedback to the authors and can, therefore, engage in shaping OCORA the security approach.

Prior to reading this document, it is recommended to study the Release Notes [1], the Introduction to OCORA [2] and the Problem Statements [3]. The reader should also be aware of the Glossary [4] and the OCORA-40-009-Gamma – (Cyber-) Security Overview [5].

3 Security Protection Goals

Railway operators pursues a unified approach to securing information and information processing systems. The objective of security is to protect installations and systems on vehicles that use or contain ICT (Information and Communication Technology) against improper interference.

Security is not only protection against attacks from the outside world, but also against attacks from the inside and includes prevention of unwanted manipulation. The need for protection and the risk situation characterize the requirements, which are continuously updated.

Due to the increased merging of IT and classic operations technology (train control, etc.) and the interconnection of all relevant components, security in general and as an enabler of safety is becoming increasingly important. Thus, the two disciplines, security, and safety, must be considered in their interdependence.

The definition of the security protection goals needs to be based on the specifications of the national authority for transportation (BAV, BMK, EBA, etc.), in particular the requirements for protection against unauthorized access and availability and the standard of the Federal Office for National Economic Supply, i.e. the standards IEC 62443 [10] and prTS 50701 [9].

The primary objectives of security are to ensure that security incidents do not result in safety incidents and to protect from cyber blackmail. In addition, security must support the reliable operation of a railway operator, in particular high availability, and integrity or, in the event of damage, i.e. "graceful degradation", while maintaining financial viability. Graceful degradation is a stability- and security-oriented reaction of a system to errors, unexpected events, or partial failures, in which the system maintains operation as good as possible.

The security protection goals are framed by generally accepted security objectives:

- Availability: The information and information systems are available within the planned thresholds.
- Integrity: The information is complete, up-to-date and is processed and transmitted as defined.
- Confidentiality: The information is only accessible to authorized persons/systems.
- Authenticity: The processing of information can be assigned to an instance or a person

Additionally, the general project objectives are structured in five objectives, each one contains an objective from the point of view of the railway operator and from the point of view of the customer.

The main objectives are:

- Cost
- Capacity
- Availability
- Safety
- Service Quality

Based on this, the following security protection goals are defined:

- Goal 1: No "safety" incident
- Goal 2: ISMS by ISO 27001, controls by pr TS 50701 (IEC 62443)
- Goal 3: Financial viability: security contributes to cost-effectiveness
- Goal 4: Prevent blackmail
- Goal 5: The maximum tolerable delay (train delay minutes per year)
- Goal 6: The maximum total failure tolerance
- Goal 7: The maximum partial operation tolerance

3.1 No "Safety" Incidents

In the area of railway infrastructure security is primarily dedicated towards securing the people from the system. Security must support this effort.

3.2 ISMS and Controls

Use processes and an ISMS (information security management system) according to ISO/IEC 27001 [13]. Use the controls of IEC 62443-3-3 [10]. Operate processes and an ISMS and strive for conformity with the standards ISO/IEC 27001 [13] and pr TS 50701 [9] (IEC 62443 [10]).

3.3 Financial Viability

Any security means shall be tested against the financial viability for the railway operator. It is counterproductive to define requirements, that are dropped in the procurement process caused by high cost.

3.4 Prevent Blackmail

Securing against cyber attacks of the attacker type "nation states" goes beyond the financial abilities of the railway operators. Therefore, this is not a goal. The current attack landscape shows highest activity and therefor probability for attacks driven by financial blackmailing.

3.5 Train Delay

The maximum tolerable delay caused by security incidents within railway operation needs to be defined and tracked (e.g. 100'000 minutes delay per year).

3.6 Maximal Total Failure Tolerance

The maximum tolerance for total failure caused by security incidents needs to be defined by each railway operator (e.g. one hour, within one hour a total service failure must be reinstated to a partial service level).

3.7 Maximal Partial Operation Tolerance

The maximum tolerance for partial operation caused by security incidents needs to be defined by each railway operator (e.g. four hours, within four hours a reduced service level must be reinstated to a normal service level).

4 Security Principles

Security is a collaborative continuous effort. All members including employees and partners are responsible for the protection of information systems and information in their area of responsibility, influence, and control.

The following principles are taken into consideration for OCORA. They shall support the achieving the ambitious security protection goals stated in chapter 3.The OCORA security principles represent a mind-set that underpins and influences the design and architecture of OCORA. They are the foundation for more detailed security related requirements that must be translated into functionalities by the suppliers.

The OCORA security principles have been derived from security standards and best practices to provide guidance for designing and architecting the complex system of an onboard CCS system. They are based on well-known security standards and best practices (i.e. ISO2700x [13] [14], NIST Cybersecurity Framework [11] [12], IEC 62443 [10]) as well as rail specific standards and best practices (i.e. prTS 50701 [9], EN 50126 [6]).

4.1 Secure by Design

Make security part of requirements, and not an afterthought. Protect a business application or information system against attacks by considering security requirements as part of its overall requirements.

- Experience has shown it is both costly and difficult to implement security measures after a system has been developed
- Avoid unnecessary development efforts by considering security requirements early on
- As security interferes with safety (e.g. timings, fail behavior) there must be a holistic approach

Implications:

- Understand the resulting security requirements in the engineering, design, implementation, and disposal of the system
- Security should treat the root cause of a problem, not its symptom

4.2 Defence in Depth

Avoid reliance on a single type of security control. Implementing security on multiple layers is better than relying on a single defence layer. If one security control fails or is bypassed, an additional layer can help preventing the attack.

- Identify and secure the weakest links first
- Use multiple security layers to increases the effort required for an attacker to compromise a system or application

Implications:

- Create a security architecture that documents the different layers of protection
- Balance defense in depth against simplicity and business needs
- Each subsequent security layer should not trust the previous layers
- Compartmentalize the system by defining security boundaries for information flows
- Prepare for the worst possible compromise scenario

4.3 Detectability (Logging & Monitoring)

It is impossible to manage something what you can not see nor measure. Exceptions, failures, and maybe outages will happen in complex systems and these cannot be predicted with accuracy. Effective and continuous monitoring of system states is the key to detect deviations, failures, and attacks easily and early and to act swiftly to combat cascading failures or risks.

Implication:

- Start the logging & monitoring design with the goal of being able to detect breaches and compromise.
- Design components and systems to be able to log exceptions, vital heath and security events in a standard format and way

Design systems for secure and centralized logging (protection of log data in motion and at rest).

4.4 Design for Security Automation

Complexity can be managed easier when security related processes are automated. Manual security tasks are inefficient, expensive, and prone to inconsistencies and human error. It is no longer possible to deploy, operate, and secure complex applications and infrastructures without automation. Security, agility, scalability, and control are a direct function of automation in today's complex and rapidly changing technology and threat environment.

Implications:

- Automation reduces complexity and ensures consistency
- Reduces the talent gap by freeing scarce expertise form mundane tasks
- Automated testing
- Requires discipline and design

4.5 Secure by Default

Set secure default options to limit inherent security vulnerabilities. System or application configurations should favour security over not being secure. The default setting for a security control should be to deny access to a resource and require a configuration to specifically grant access. When the system goes into an error or exception state, these states must favour security over not being secure.

Implication:

- Security should not require extensive configuration to work and should just work reliably where implemented
- Establish secure defaults when system starts or goes in error or exception states
- Providing least privilege or making only necessary services and features available
- Use encryption by default for both data at rest and in transit

4.6 Simplicity over Complexity

Complexity is the worst enemy of security. Complexity in systems leads to increased human confusion, errors, vulnerabilities, automation failures, and difficulty of recovering from an issue. Favour simple and consistent architectures, designs, and implementations. Avoid unnecessary complexity. The more complex the system is, the more likely it may possess exploitable flaws.

Implication:

- Simplicity should be a key objective in design of systems and security
- Do not repeat yourself
- Reduce the variety and types of hardware and software

- Design systems that use the least resources possible (in terms of hardware and software)
- Favor convention over configuration
- Do not implement unnecessary security mechanisms
- Complexity makes vulnerabilities harder for developers and testers to uncover. Each feature, function, and interaction are a potential threat vector
- Complexity makes vulnerabilities harder to fix

Notes:

- No over-simplifying
- Balance reduced complexity against diversity required to achieve resiliency and reduced single-point-of-failures

4.7 Assume Failure & Compromise

Complex distributed systems lead to unpredictability and cascading failures. Even when all the individual components of complex system are functioning properly, the interactions between those components can cause unpredictable outcomes and vulnerabilities. Rare or surprising combinations of events, vulnerabilities, and creative user interactions make such systems inherently chaotic. In such systems prediction, complete testing, and modelling of all states is not possible in such systems with manageable efforts. Therefore, we must assume and account for failures and compromise.

Implications:

- Our systems are too complex to anticipate all potential interactions or vulnerabilities
- Assume that critical parts of the infrastructure are already compromised when designing architectures, systems, and components
- Embrace principles of chaos engineering and testing - facilitate real and repeated tests to uncover systemic weaknesses
- Design system for automated testability
- Establish continued and comprehensive monitoring of vital parameters to determine system health and security

4.8 Fail Safe and Secure (Graceful Degradation)

Failures should lead to a safe and secure state. Risk does not hurt - the impact does. If a security control fails, it should maintain a state of deny access. Design security mechanisms so that a failure will follow the same execution path as disallowing the operation. Prevent unauthorized access in case of errors, failures, exceptions, system degradation, or compromise**.**

Implication:

- Design to minimize the impact of component or control failures or compromise
- Confidentiality and integrity assurance top availability assurance
- Security methods (like authorized, authenticated and validated) should all return false if there is an exception during processing
- Assume system failure & compromise in design decisions
- Ensure safe reaction on denial

Examples:

- Dead man’s switch is automatically operated if the human operator becomes incapacitated
- Traffic light controllers use a Conflict Monitor Unit to detect faults or conflicting signals and switch an intersection to an all flashing error signal, rather than displaying potentially dangerous conflicting signals.

4.9 Usability & Manageability

Balance of security and usability - make secure behaviour easy instead of complex. Make it easy to do the right thing, make it difficult to do the wrong thing, and make it almost impossible to do the catastrophic thing. Security controls should not obstruct users in performing their work and should not be difficult to manage. User interfaces must be easy to use, so that users routinely and automatically apply the mechanisms correctly. This relates to the paradigm of Least Astonishment in UI design and Simplicity Principles

Implications:

- A component or system should be designed to behave in a manner consistent with how users of that component are likely to expect it to behave
- Design security interfaces and functions for ease of use, so that users routinely and automatically apply the protection mechanisms correctly

Note:

- If security gets in the way, sensible, well-meaning, dedicated people develop hacks and workarounds that defeat the security

4.10 Open Design

The security of a mechanism should not depend on the secrecy of the details of its design or implementation. Assume outsiders and attackers will have access to source code (also for closed source software) and complete design and network topologies. Assume sensitive information regarding security measurements are leaked or sold. Encourage proactive reporting of security issues or vulnerabilities and act on such reports.

Implications:

- Never store secrets in code, documentation, or configurations
- Open security design promotes faster improvement cycles
- Security measurements should be open and transparent

Examples:

- Shannon’s Maxim: The enemy knows the system

4.11 Zero Trust

Assume everything to be insecure until a level of trust is established. The historic concept of trust that is based on a perimeter separating the inside from the outside does no longer hold in today’s rapidly changing environment. Assuming no trust is a security model that more effectively adapts to the complexity of the modern environment. It embraces the mobile workforce, and protects people, devices, apps, and data wherever they are located.

Implication:

- Trust is not granted until the user, system, or component can be authenticated and authorized first
- Verify anything and everything trying to connect to a system before granting access
- Workforce: Authenticate users and continuously monitor and govern their access and privileges
- Workloads: Enforce controls across the entire application stack, especially connections between containers or hypervisors in the public cloud
- Data: Secure and manage data, categorize, and develop data classification schema, and encrypt data at rest and in transit
- Supply Chain: Question and assess the integrity and security of suppliers and the delivered products, systems, and services

4.12 Least Privilege

Only grant the minimal set of permissions that are necessary for a given action - and no more. Systems and users should operate while invoking as few privileges as possible. Granting permissions beyond the scope of the necessary rights of an action can allow a user or system to obtain or change information in unwanted ways. This principle limits the damage that can result from an attack, accident, or error. It also reduces the number of potential interactions among privileged systems to the minimum for correct operation, so that unintentional, unwanted, or improper uses of privilege are less likely to occur.

Implication:

- Minimize the system elements to be trusted
- This principle restricts how privileges are granted and revoked and timed out

4.13 Design Security for Safety

An insecure system is not safe (also called "security4safety"). This ensures that security threats are prevented from exploiting vulnerabilities in control systems that would compromise the integrity of safety functions. In addition, security functions and safety functions are separated so that security updates do not require safety recertification (separation of concerns). The security design of a system with safety relevance (SIL1-4) shall include the following means by level:

Implication:

- "Secure by design"
- "Defense in depth"
- "Detectability"
- System theoretic process analysis (STPA) for security with the concern's "safety", "crisis mode" and "availability".
- Ensure "heterogenity4security" (where possible, reuse dissimilarity)
- Separate the concerns of security and safety where possible
- Ensure a patch process inside the safety realm is available
- Make sure the system "fails safe" even when compromised.

Considering a security compromise or breach is caught through means on the respective level:

- The lower the risk is, the lower the number of the upper means is.
- The "difficulty of means" (and most likely cost) is higher, the higher the number of the upper means is.

4.14 Design Security for Innovation

Security shall be designed with innovation and agile development in mind (also called "security4agility"). Security functions are relevant quality attributes for the system.

Implication:

- Security means shall be made in a way, that agile development and innovation is still possible.

4.15 Design Legacy System Inclusion

Means for the secure operation of old systems are provided. New Systems and components will be introduced step by step over a timespan of at least 20 years. In that time those systems must cooperate with old systems with less security. Even after a complete deployment of completely new systems there will be systems interacting without the respective security means.

Implication:

- It must be possible to integrate legacy systems (e.g. locomotives) into new components and systems.
- Security must provide measures to handle such exceptions in a more secure way ("broken security").

Methods and technics for the integration of legacy systems encompass hardening, monitoring, ensuring least privileges are offered and compensating controls (technical, procedural) to name a few.

4.16 Heterogeneity for Security

Ensure availability even when full stacks must be shut down for security reasons. The system and components must be able to function even after massive security compromises have been detected.

Implication:

- If a major security weakness (systematic failure) is detected in an application stack (e.g. in APS), it must be possible to shut down that stack completely and run the corresponding region on a dissimilar other stack until the weak stack is patched, tested and put back into operation.
- Availability and dissimilarity shall be guaranteed for several weeks on a reduced (n minus 1) stack.

5 Security Guidelines

The security implementation guidelines provide further details on how the security principles will be applied.

The following are the implementation security guidelines:

- Dynamic Security
- Demand and promote security to suppliers
- Industrial collaboration
- Supply chain security
- Support Standardisation
- Build on available efforts from the partner railways
- Advanced availability and robustness
- Accountable operation and SIEM capability

5.1 Dynamic Security

Dynamic security via integrated toolchains in the Cyber Defence Center (CDC) up to the patching of the systems. Information about assets, security analysis, threats, and the risk assessment (impact, easiness, detectability) shall be combined with a dynamic feed of new vulnerabilities. The result is a real-time risk dashboard that shows where security actions/measures are most important.

5.2 Demand and Promote

Dialogue procedures with suppliers shall be used to determine what is affordable to maintain. The work in the standardization organizations is also essential about security and industry partners (EULYNX, RCA, etc). In this context, the following principles must be considered:

- Nation wide: use generic applications
- Europe wide - European Union Agency for Railways (ERA): use generic standards
- Industry - internationally: use generic products

5.3 Industrial Collaboration

Furthermore, a regular exchange with the various stakeholders needs to be maintained. The OCORA cooperation partners work closely together, whereby resources from the OCORA members are directly involved in the daily work.

5.4 Supply Chain Security

It must be ensured that the complete supply chain (suppliers, contractors, vendors, and operators) is understood and supports all security efforts (like cybersecurity verification, cybersecurity validation and cybersecurity assurance).

5.5 Support Standardization

All railway operators shall support standardization projects of security in Europe (example: RCA; EULYNX Baseline 4; prTS 50701 [9]; FRMCS).

5.6 Build on available Efforts

Build on available efforts of the partner railways. Close coordination with cyber security departments from railway operators and other partners shall be maintained.

5.7 Advanced Availability and Robustness

The solution offers continuous operation and availability even if faced with partial failures (controlled graceful degradation). It has no single point of failure in elements that cause domino effects or have high leverage in terms of availability. Any loss of integrity or failure must be revealed instantaneously. The system will actively prevent triggering safety reactions.

5.8 Accountable Operation and SIEM Capability

The solution (-process) must provide real-time security related monitoring including all operational interventions such as trouble shooting, deployment, configuration management and change management. All participating elements have a properly authenticated identity. No anonymous components are accepted.

END OF DOCUMENT