Industrial Internet of Things
Volume G4: Security Framework

IIC:PUB:G4:V1.0:PB:20160919

USE OF INFORMATION—TERMS, CONDITIONS AND NOTICES

This is an Industrial Internet Consortium document (the “Document”) and is to be used in accordance with the terms, conditions and notices set forth below. This Document does not represent a commitment by any person to implement any portion or recommendation contained in it in any products or services. The information contained in this Document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) and its Industrial Internet Consortium (the “IIC”) a nonexclusive, irrevocable, royalty-free, paid up, worldwide license to copy and distribute this Document and to modify this Document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any

such copyright holder by reason of having copied, distributed or used such material set forth herein.

Subject to all of the terms and conditions below, the owners of the copyright in this Document hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense) to use, copy and distribute this Document (the "Permission"), provided that: (1) both the copyright notice above, and a copy of this Permission paragraph, appear on any copies of this Document made by you or by those acting on your behalf; (2) the use of the Document is only for informational purposes in connection with the IIC's mission, purposes and activities; (3) the Document is not copied or posted on any network computer, publicly performed or displayed, or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (4) no modifications are made to this Document.

This limited Permission is effective until terminated. You may terminate it at any time by ceasing all use of the Document and destroying all copies. The IIC may terminate it at any time by notice to you. This Permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, or at any time upon the IIC's express written request, you will destroy immediately any copies of this Document in your possession or control.

The Licenses and Permission relate only to copyrights and do not convey rights in any patents (see below).

PATENTS

Compliance with or adoption of any advice, guidance or recommendations contained in any IIC reports or other IIC documents may require use of an invention covered by patent rights. *OMG and the IIC are not responsible for identifying patents for which a license may be required* to comply with any IIC document or advice, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. IIC documents are informational and advisory only. Readers of this Document are responsible for protecting themselves against liability for infringement of patents and other intellectual property that may arise from following any IIC recommendations or advice. *OMG disclaims all responsibility for such infringement*.

GENERAL USE RESTRICTIONS

This Document contains content that is protected by copyright. Any unauthorized use of this Document may violate copyright laws, trademark laws and communications regulations and statutes. Except as provided by the above Licenses, no part of this work covered by copyright may be reproduced or used in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping or information storage and retrieval systems—without permission of the copyright owner(s).

DISCLAIMER OF WARRANTY

WHILE THIS DOCUMENT IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP, INC. (INCLUDING THE IIC) AND THE COPYRIGHT OWNERS LISTED ABOVE MAKE NO WARRANTY, REPRESENTATION OR

IIC ISSUE REPORTING

All IIC documents are subject to continuous review and improvement. As part of this process, we encourage readers to report any ambiguities, inconsistencies or inaccuracies they may find in this Document or other IIC materials by sending an email to *admin@iiconsortium.org*.

ACKNOWLEDGEMENTS

This document is a work product of the Industrial Internet Consortium Security Working Group, co-chaired by Sven Schrecker (Intel), Hamed Soroush (Real-Time Innovations) and Jesus Molina (Fujitsu), in collaboration with the Technology Working Group co-chaired by Shi-Wan Lin

(Thingswise), Bradford Miller (GE) and E. Eric Harper (ABB), and the Safety Task Group Qinqing (Christine) Zhang (JHU) and Andrew King (University of Pennsylvania).

EDITORS

Stephen Mellor (IIC), Marcellus Buchheit (Wibu-Systems), JP LeBlanc (Lynx Software Technologies), Sven Schrecker (Intel), Hamed Soroush (Real-Time Innovations), Jesus Molina (Fujitsu), Robert Martin (MITRE), Frederick Hirsch (Fujitsu), Kaveri Raman (AT&T), Jeffrey Caldwell (Belden), David Meltzer (Belden) and Jeff Lund (Belden).

AUTHORS

The following persons have written substantial portions of material content in this document: Sven Schrecker (Intel), Hamed Soroush (Real-Time Innovations), Jesus Molina (Fujitsu), JP LeBlanc (Lynx Software Technologies), Frederick Hirsch (Fujitsu), Marcellus Buchheit (Wibu-Systems), Andrew Ginter (Waterfall Security Solutions), Robert Martin (MITRE), Harsha Banavara (Schneider Electric), Shrinath Eswarahally (Infineon Technologies), Kaveri Raman (AT&T), Andrew King (University of Pennsylvania), Qinqing (Christine) Zhang (Johns Hopkins University), Peter MacKay (GE Wurldtech) and Brian Witten (Symantec).

CONTRIBUTORS

The following persons have contributed valuable ideas and feedback that significantly improve the content and quality of this document:

Brad Miller (GE), Michael Jochem (Bosch), Jeff Caldwell (Belden), Timothy Hahn (IBM), Anish Karmarkar (Oracle), David Welsh (Microsoft), David Meltzer (Belden), Jason Garbis (EMC), Kambiz Esmaily (Toshiba), Lancen LaChance (GlobalSign), Marc Blackmer (Cisco), Geoffrey Cooper (Intel), Mitch Tseng (Huawei), Omer Schneider (CyberX), Rajeev Shorey (TCS), Rob Lembree (Intel), Shi-wan Lin (Thingswise), Steve Hanna (Infineon Technologies), Ekaterina Rudina (AO Kaspersky Lab), Yoshiaki Adachi (Hitachi), Suresh K. Damodaran (MITRE), Tom Rutt (Fujitsu), Robert Swanson (Intel), David Wheeler (Intel), Arjmand Samuel (Microsoft) and Michael Mossbarger (ENT Technologies).

CONTENTS

Part I: Introduction

Part II: The Business Viewpoint

Part III: The Functional and Implementation Viewpoints

Annexes

FIGURES

TABLES

This document is the first version of the 'Industrial Internet of Things, Volume G4: Security Framework' (IISF). It initiates a process to create broad industry consensus on how to secure Industrial Internet of Things (IIoT) systems.

The IIoT is being shaped by many participants from the energy, healthcare, manufacturing, transportation and public sectors, each of which needs to consider security. To avoid security hazards, especially as systems from different sectors interoperate and exploitation attempts are made in the gaps between them, it is important and urgent to build early consensus among the participants on IIoT security.

This work builds on 'Industrial Internet of Things, Volume G1: Reference Architecture' (IIRA, [IIC-IIRA2016]) that lays out the most important architecture components, how they fit together and how they influence each other. Each of these components must be made secure, as must the key system characteristics that bind them together into a trustworthy system.

This document extends naturally from a chapter in the IIRA describing security concerns. It moves into security-specific territory to ensure security is a fundamental part of the architecture, not bolted onto it.

This document has several parts that do not mirror the IIRA document structure exactly. Part I examines key system characteristics, how they should be assured together to create a trustworthy system, and what makes IIoT systems different from traditional IT systems.

Part II reviews security assessment for organizations, architectures and technologies. It outlines how to evaluate attacks as part of a risk analysis and highlights the many factors that should be considered, ranging from the endpoints and communications to management systems and the supply chains of the elements comprising the system. Different roles are identified that should be considered in conjunction with the key characteristics, including, owner/operator, system integrator/builder and equipment vendor. Each role offers different risk management perspectives that affect the decisions regarding security and privacy.

Part III covers the functional and implementation viewpoint of the IIRA (and subsumes its usage viewpoint). It describes good practices for achieving confidentiality, integrity and availability, and considerations for trusting data when it is communicated and stored, as well as establishing trust in the code and overall execution environment. It also includes patterns for protecting against and limiting risks, including firewalls, separation of networks, separation of privilege, unidirectional gateways, identity management, cryptography, public key infrastructure and trusted execution environment.

The annexes cover topics that apply to more specific segments of the security domain. One covers numerous guidelines, standards and regulations relating to protection of industrial internet systems and discusses the role of standards and compliance in industrial internet Security. Another provides an example of a cybersecurity capability maturity model for evaluating the maturity of the security posture and associated processes within an organization. The last annex lists some security techniques and processes, their mapping to important security objectives, and their high-level requirements.

Part I: Introduction

An *Industrial Internet of Things* (IIoT) system connects and integrates industrial control systems with enterprise systems, business processes and analytics. An IIoT system enables significant advances in optimizing decision-making, operations and collaborations among a large number of increasingly autonomous control systems.

These systems differ from traditional industrial control systems by being connected extensively to other systems and people, increasing their diversity and scale. They also differ from traditional information technology (IT) systems in that they use sensors and actuators in an industrial environment. These are typically systems that interact with the physical world where uncontrolled change can lead to hazardous conditions. This potential risk increases the importance of safety, reliability, privacy and resiliency beyond the levels expected in many traditional IT environments. Such IIoT systems may also have data flows that include multiple intermediary organizations, requiring security approaches beyond simple approaches such as link encryption. Having long lifetimes, IIoT systems include legacy installations and are regulated because human health and safety is at risk. The cultures of operational and information technology worlds differ, leading to a need to integrate these cultures for IIoT systems. All of these differences have implications on how these systems need to be secured.

Part I examines key system characteristics, clarifying how they should each be assured and assured together to create a trustworthy system appropriate for IIoT systems, taking into account what makes these systems different.

1 OVERVIEW

This document is relevant to enhancements to existing implementations and new implementations. It provides guidance for improving organizational approaches, processes and the use of technologies for creating a trustworthy system.

Subsequent revisions of this document may consider additional details or topics as needed.

1.1 PURPOSE

The purpose of this document, 'Industrial Internet of Things, Volume G4: Security Framework' (IISF) is to identify, explain and position security-related architectures, designs and technologies, as well as identify procedures relevant to trustworthy Industrial Internet of Things (IIoT) systems. It describes their security characteristics, technologies and techniques that should be applied, methods for addressing security, and how to gain assurance that the appropriate mix of issues have been addressed to meet stakeholders' expectations.

This document is also a reference for the Industrial Internet Consortium's testbeds that already span verticals such as smart grid, transportation, industrial maintenance and others. The security evaluations of these testbeds will provide continuous feedback that will be used to update the information here in subsequent revisions of this document.

1.2 SCOPE

This work is an expansion of the discussion on security in 'Industrial Internet of Things, Volume G1: Reference Architecture' (IIRA, [IIC-IIRA2016]). The reader should be familiar with that document, as many of the terms and concepts used here are defined there.

This security framework identifies and explains how risks associated with security and privacy threats may be identified, evaluated and mitigated using technologies and processes. Privacy and other system characteristics are mentioned where it relates to specific security concerns within the document, but this document is not intended to be a tutorial on privacy, safety or other characteristics defined in the IIRA.

This document is informational in nature and not a normative technical specification. It does not contain specifications for conformance or compliance. Implementations may use a variety of mechanisms to address the concerns noted in the document.

1.3 AUDIENCE

The audience for this document includes owners, operators, system integrators, business-decision makers, architects and any stakeholder with interest in security and related key system characteristics. Business decision makers can use this document to guide the development of interoperable technologies and solutions related to security, balancing it with other stakeholder requirements. Owner, operators and system integrators can use it as a common starting point of system conception and design related to security.

1.4 TERMS AND DEFINITIONS

The 'IIC Vocabulary' [IIC-IIV2016] provides terminology and definitions for this document and other IIC documents. All acronyms are listed in Annex E and are hyperlinked in the text, marked with dotted underlines.

The document refers to multiple standards' development organizations most commonly known by their acronyms. These include International Standards Organization (ISO), Institute of Electrical and Electronic Engineers (IEEE), International Electrotechnical Commission (IEC), Internet Engineering Task Force (IETF) or National Institute for Standards and Technology (NIST).

1.5 CONVENTIONS

Given that the document is non-normative, all 'must', 'may' and 'should' statements are to be interpreted as English language and not as in RFC 2119 [IETF-RFC2119].

1.6 RELATIONSHIP WITH OTHER IIC DOCUMENTS

The 'Industrial Internet of Things, Volume G4: Security Framework' (IISF) is one of many framework documents that extend from the IIRA [IIC-IIRA2016]. It provides a cohesive view of security behavior across the IIRA reference architecture viewpoints.

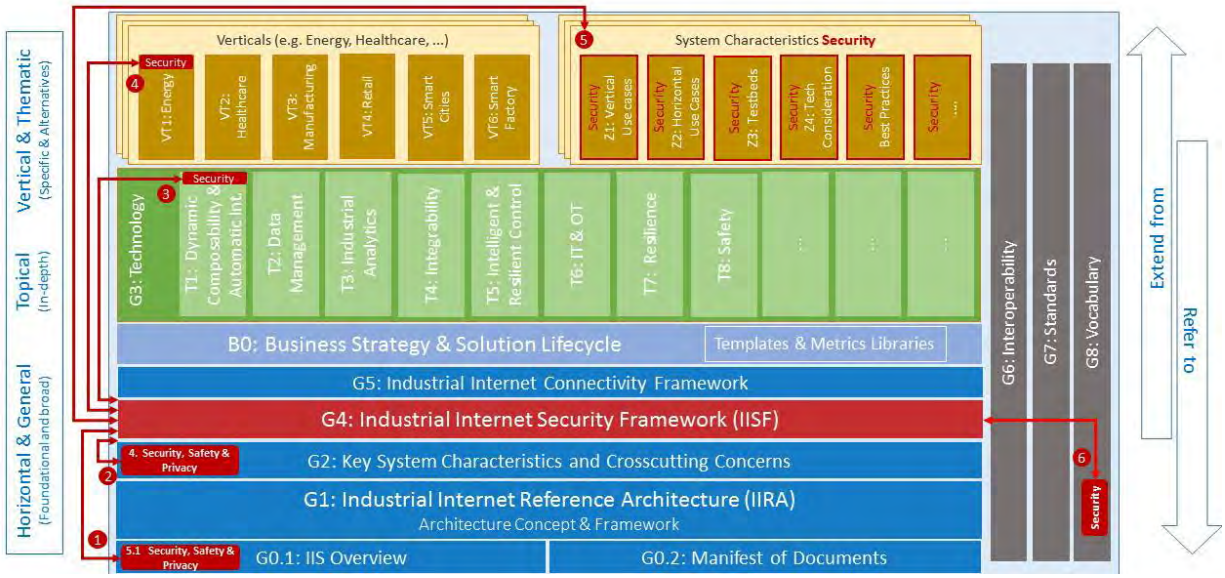

Figure 1-1: IIC Technical Publication Organization

As shown by ❶ in Figure 1-1 the IISF is part of a series of documents covering Security, Safety and Privacy issues. The IISF addresses security specifically as a key system characteristic and cross cutting concerns of an IIoT as described by ❷. This figure also shows how other documents extend from the IISF in covering security related issues in the respective areas.

Security design and implementation issues are covered in each of the technology specific documents ("T" series), such as the Connectivity framework, indicated by ❸. Specific security

implementation and design issues for each vertical target segment in the IIoT will be covered in a collection of documents ("V" series) capturing security-relevant topics as part of the use cases, testbeds, solutions and best practices for each of the addressed vertical markets as illustrated by ❹. More specific to security as a system characteristic across all vertical markets, security use cases, security solutions, and security best practices are covered as part of system-thematic specific documents as shown in ❺. Other key system characteristics such as safety and privacy will be addressed in topic-specific frameworks and will be covered in documents across all layers of the document stack in a similar fashion to security. Finally, all security related terms used in this document and their respective definitions are provided in a common 'Industrial Internet of Things, Volume G8: Vocabulary' document shown in ❻.

2 MOTIVATION

Industrial Internet of Things (IIoT) systems connect and integrate different types of control systems and sensors with enterprise systems, business processes, analytics and people. These systems differ from traditional industrial control systems by being connected extensively to other systems and people, increasing the diversity and scale of the systems.

Historically, security in trustworthy industrial systems relied on physical separation and network isolation of vulnerable components, and on the obscurity of the design and access rules for critical control systems. Security was, and still is, enforced through physical locks, alarm systems and in some cases armed guards. The potential for human error or misuse was primarily through direct access and concerns focused on disrupting the safety and reliability of the system, with those risks mitigated by good design, analysis and reviews, thorough testing and training. Designers and operators rarely considered that these systems might one day be exposed to a global network, remotely accessible by many, from legitimate users to rogue nation-states.

Over the past few decades, increasingly affordable computing power, ubiquitous connectivity and evolving data analytics techniques have opened the door to convergence of control systems, business systems and the internet. This convergence started small, initially being used for remote monitoring and management of systems, but quickly expanded to include mining and analyzing operations data for performance metrics to predict failures, optimize across fleets and perform remote software upgrades. This convergence has increased productivity, efficiency and performance of the existing operational processes and enabled the creation of new ways of leveraging operations data, thus delivering business value now and into the future.

But with these gains come risks. Systems that were originally designed to be isolated are now exposed to attacks of ever-increasing sophistication and the design assumptions of existing operational technology (OT) systems no longer apply. A successful attack on an IIoT system has the potential to be as serious as the worst industrial accidents to date (e.g. Chernobyl and Bhopal), resulting in damage to the environment, injury or loss of human life. There is also risk of secondary damage such as disclosure of sensitive data, interruption of operations and destruction of systems during such an attack. The results of attacks on IIoT systems may be widespread and comparable to large natural disasters, but stemming from malicious intent. This will result in damage to brand and reputation, material economic loss and potential damage to

critical infrastructure. With a geographically distributed IIoT system, care must be taken to ensure that disruption of an isolated system does not cascade to have global effects.

Organizations must take these risks seriously; they must use their expertise to make their IIoT systems trustworthy. The use of sensors and actuators in an industrial environment is not the typical Information Technology (IT) experience, nor are systems that span many organizations and organizational systems. IT and OT prioritize system characteristics differently. For example, resilience in IT is less important than in OT, and security is less important in OT than in IT, as illustrated in Figure 2-1. These characteristics interact with each other, and can conflict. In IIoT systems, these system characteristics must converge and be reconciled with each other into overall system *trustworthiness*.

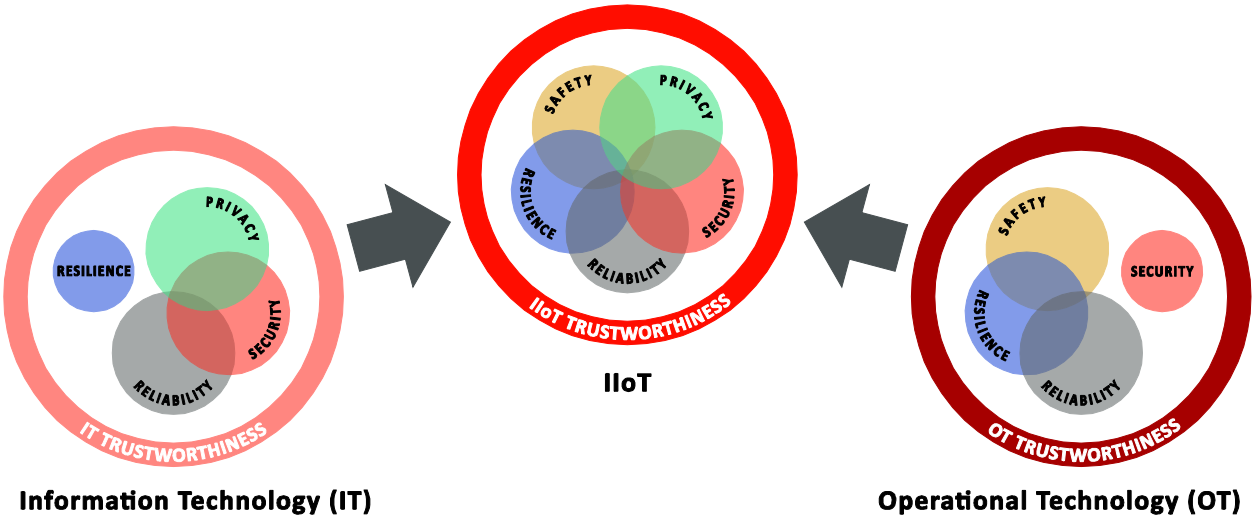

Figure 2-1: Convergence of IT and OT Trustworthiness

IIoT organizations must place increased importance on safety and resilience beyond the levels expected in many traditional IT environments. IIoT systems may also have data flows that include intermediaries and involve multiple organizations, requiring more sophisticated security approaches than, for example, link encryption. Unfortunately, IT departments rarely speak the same language as those concerned with control systems and OT. The two perceive risk differently, and they cannot be combined for positive gain without a balanced consideration of their differing motivations.

The highest priority of many OT systems is safety: do not cause injury or death, do not put public at risk and protect the environment from harm. The second and third priorities are often quality of production and meeting production targets, which depend on the reliability and resilience of the system. Reliability and resilience are required to prevent the interruption of society-critical processes such as the electric grid, and to avoid idling machinery that represents large investments in physical infrastructure. Security aspects are considered in OT, but given that most systems are not connected it is mostly physical security. (Some industries, such as healthcare, must protect patient data.) Security concepts such as user-based access control applies less often in OT systems than they do in IT.

On the other hand, security and privacy are important characteristics for most IT systems, together with reliability. Safety is rarely an issue, and resilience is reserved for specialized systems where business continuity is a motivating factor, for example for financial transactions.

This document offers a framework to balance the security-relevant considerations of the two different cultures, OT and IT. As each culture endeavors to create trustworthy systems that deal with their functional needs, environment, possible disruptions, system faults, human errors and attacks, the considerations need to be made explicit so that members of each can understand and appreciate the needs and motivations of the other.

3 KEY SYSTEM CHARACTERISTICS ENABLING TRUSTWORTHINESS

An Industrial Internet of Things (IIoT) system exhibits end-to-end characteristics that emerge as a result of the properties of its various components and the nature of their interactions. The five characteristics that most affect the trust decisions of an IIoT deployment are security, safety, reliability, resilience and privacy. These are referred to as *key system characteristics*. Others, for example, scalability, usability, maintainability, portability or composability may be important in general too but are not considered "key" in respect to trustworthiness. Each key system characteristic must be assured in its own way, but there are some common techniques.

3.1 ASSURANCE OF KEY SYSTEM CHARACTERISTICS

Assurance requires the collection and analysis of evidence that supports the design, construction, deployment and test of the system, and its activities in operation. The evidence must support the claim that the right mixture of innate system capabilities and compensating security controls to mitigate risks has been put in place.

Assurance includes risk analysis to identify hazards and prevent incidents or accidents. *Risk*, the effect of uncertainty on objectives, takes into consideration the likelihood of an event occurring along with the impact of that event if it were to occur. Rigorous product and system design, including design reviews and testing, intends to prevent faulty operations and improve system resilience to potential events identified in the risk analysis.

When making claims about what has been done to address specific attacks and weaknesses, public knowledge sources[1] should be used when possible so that discussion of these aspects can be grounded in common terminology and the same reference source(s).

Assurance cases structure the reasoning behind claimed security behavior, features or absence of vulnerability. They provide evidence about removal of weaknesses by means of protection mechanisms and security features, and provide arguments supporting claims that key system

[1] Example of public sources include [CWE], [CAPEC], [OWASP], [WASC], [ATT-CK] and [CVE]

characteristics have been met. Assurance cases demonstrate to stakeholders that their expectations for each key system characteristic have been met.[1]

An *abuse case* is a test that provides inappropriate inputs to determine how the system responds. Abuse cases are similar to use cases in that they make an interaction explicit; they differ in that the result of the interaction is harmful.[2] Abuse tests that fail are evidence to support the claims. With appropriate misuse and abuse testing included in the assurance case, the stakeholders can gain confidence that attackers' influence has been limited. These cases can be used both for requirements analysis and testing.

A *threat model* is a systematic approach to the definition of potentially hazardous events and malicious attacks to the system. It begins with identifying the most important ways in which system behavior may be compromised. The types of security violations are then elaborated into concrete threats, and they may be validated using abuse cases. This top-down approach reveals other threats so that comprehensive security measures can be developed during system design, implementation, configuration and maintenance.

3.2 SECURITY

Security is the condition of the system being protected from unintended or unauthorized access, change or destruction.

The secure behavior of a system is a continuum, not a Boolean state. No IIoT system can behave securely in every context so the specific contexts deemed relevant must be explicitly stated along with the secure behavior that the stakeholders expect.

Assurance of security is often assessed in terms of risk. Elements of security risk include a threat (someone or something that is attempting to do harm), the targeted asset (that has a value), a potential vulnerability or weakness of the asset that the threat will exploit, and countermeasures that attempt to reduce the likelihood and impact of any security incidents.

The elements that need be upheld to provide the security of information and system assets are confidentiality, integrity and availability, often referred to by the acronym CIA.

Confidentiality is the property that information is not made available or disclosed to unauthorized individuals, entities or processes. Breaches of confidentiality can occur by word of mouth, printing, copying, emailing, or through software vulnerabilities that allow attackers to read or exfiltrate data. *Data exfiltration* is the unauthorized transfer of data read through exploits at another location under the control of the attacker. This data may be used for blackmailing or other purposes. Confidentiality controls include access control and encryption technologies.

Integrity ensures that improper information modification or destruction is guarded against. Integrity controls include hashes, checksums, anti-virus functionality, whitelisting and code signing that ensure there have been no changes to the system, code and elements controlling

[1] See [NIST-7608]
[2] See [McDer1999]

the physical processes of the system. *Data integrity*, a subset of integrity, ensures that unauthorized parties cannot alter data and take control of the system without detection.

Availability is the property of on-demand, timely and reliable access to and use of information by an authorized user. The systems responsible for controlling the physical process should provide continuous control and oversight by human operators of the physical process. A human may need to intervene in the case of an attack, for example to shut the system down. Availability controls generally involve redundancy and engineering change control. Sometimes they include security activities that find and mitigate software vulnerabilities that create unreliable execution, visualization or resource consumption that negatively affect the systems.

In traditional operational technology (OT) systems, availability has been considered paramount, followed by integrity, with confidentiality generally being the last consideration, leading to the acronym AIC (also known as the *security triad*).

3.3 SAFETY

Safety is the condition of the system operating without causing unacceptable risk of physical injury or damage to the health of people, either directly or indirectly, as a result of damage to property or to the environment.

Assurance of safety endeavors to eliminate both systematic and probabilistic failures. Traditional OT safety-assessment techniques focus on physical items and processes, then combine empirically derived component failure probabilities into total system risk. Risk analysis to identify hazards intends to prevent faulty operations and improve system resilience to unexpected events.

However, a software component always behaves exactly as it is programmed; it is not possible to make useful statistical characterizations of software failures. If a software component has never misbehaved during testing, it may not have been exposed to a sequence of inputs that would have uncovered the defect. Test coverage does not necessarily correlate to failure rate. Approaches for managing probabilistic failures do not address threats because adversaries will be able to exploit security-related systematic failures reliably once those vulnerabilities have been discovered.

Traditional efforts for industrial software focused on functional correctness and did not assume that an adversary was involved. In today's connected systems, a remote attacker is able to exploit weaknesses[1] to drive the system into an unsafe state. This contrasts sharply with traditional IT security, where a security analysis of the threat and threat-actor skills and capabilities is used to determine the likelihood of weaknesses that can be exploited.

Many of the same tools, techniques and practices used to produce safety-critical software can also identify, remove and mitigate potential security weaknesses. Many safety regulations and

[1] For a public collection of weaknesses, see [CWE]

guidance documents[1] require that the software used in safety-critical systems is rigorously validated and verified using, for example, full branch-coverage testing or even formal methods to uncover security issues. Rigorous software development practices can help developers identify and eliminate potential safety issues and security vulnerabilities.

3.4 RELIABILITY

Reliability is the ability of a system or component to perform its required functions under stated conditions for a specified period of time.

Reliability and availability are related. Reliability is the fraction of actual availability over scheduled availability, as affected by scheduled maintenance, updates, repairs and backups. These reduce availability, but they do not reduce reliability if properly scheduled. Reliability reflects how much a company can count on a system working when it's scheduled and expected to be working.

Assurance of reliability requires detailed understanding of the operational environment, the system's composition and how it was engineered and pre-fielded to establish the likelihood of failure. The parameters, configuration settings and physical attributes are needed for each element. Verification activities that tested whether the planned values for these were implemented are also required. Uptime requirements and mean-time-to-failure are apportioned across the system and its components, and captured in an assurance case. Operation of the system can compare the actual reliability of the system against claimed reliability.

Adding a connection to the internet can invalidate some security assumptions from when the original design was made. In addition, it can introduce new and potentially more complex interactions with other systems. Approaches that manage purely probabilistic-related reliability failures will fail to address threats because adversaries will be able to exploit security-related systematic failures reliably once those vulnerabilities have been discovered. By considering which reliability aspects an attacker could influence and designing the system and its security to address those types of attacks, the reliability of the system can be improved.

3.5 RESILIENCE

Resilience is the emergent property of a system that behaves in a manner to avoid, absorb and manage dynamic adversarial conditions while completing the assigned missions, and reconstitute the operational capabilities after causalities.

Often, resilience is achieved by designing the system so that failures are compartmentalized. If a single function fails it should not cause other functions to fail, and there should be alternate ways of performing the failed function in the design that can be invoked automatically, immediately and reliably.

Assurance of resilience adds physical or logical redundancy for elements and interconnections and provides for transfers to the alternate elements and connections when needed. Testing

[1] For example, Positive Train Control (PTC), see [CFR-236]

should be performed for normal and abnormal scenarios and examined as to whether an attacker could purposely disrupt a combination of components.

Software must also be able to transfer over to alternate functionality, implementations, configurations, locations or network segments that may have different weaknesses so the same threats and hazards are not as disruptive to the replacement capabilities.[1]

3.6 PRIVACY

Privacy is the right of an individual or group to control or influence what information related to them may be collected, processed, and stored and by whom, and to whom that information may be disclosed.

Assurance of privacy depends on whether stakeholders expect, or are legally required, to have information protected or controlled from certain uses. It is important to stay up to date with regulations and standards, such as the new framework for transatlantic data flows called the EU-US Privacy Shield and the EU General Data Protection Regulation (GDPR)[2].

In the US, the Federal Trade Commission (FTC) maintains many guidelines that apply in commercial environments. Rules apply to firms in healthcare, finance, education, auto sales, direct marketing, entertainment and consumer credit. In each case, firms must abide by specific guidelines. For example, in healthcare environments HIPAA[3] rules must be followed when handling patient-related information.

Care needs to be taken to minimize the use of data and to address risks associated with establishing the identity of parties when those identities should not be revealed. Identity might be revealed through the examination of metadata associated with the party (fingerprinting) or the correlation of data about the party. Integrating IIoT systems might increase this risk. Security systems themselves might increase privacy risks by increasing the amount of data collected and associated with a party.

Privacy risks may increase as industrial systems are interconnected with other systems that contain sensitive data. For example, if a customer relationship management (CRM) system is integrated with a manufacturing system then information about the items produced for certain customers might be revealed through a security breach of either system. Additional risks may involve the inappropriate sharing and distribution of information by third parties, should they decide to share the sensitive data.

There are a number of frameworks that may apply, depending on regulation, but all may be useful in understanding privacy effects on business models. Examples are GAPP from AICPA, PPTF from OECD, FIPPS from FTC and 'Regulation 2016/679' from EU.[4]

[1] See [NIST-800-160]
[2] See [EU-GDPR]
[3] See [HHS-HIPAA]
[4] See [AICPA-GAPP], [OECD-PPTF], [FTC-FIPPS] and [EU-2016/679]

3.7 TRUSTWORTHY SYSTEMS

A main stakeholder goal for a system is that it be trustworthy in respect to the key system characteristics. The importance of each key system characteristic to a given deployment is unique to each system and achieving one can conflict with achieving another. Interactions between the key system characteristics must be understood based on drivers such as regulatory compliance, business process and industry norms, not in isolation.

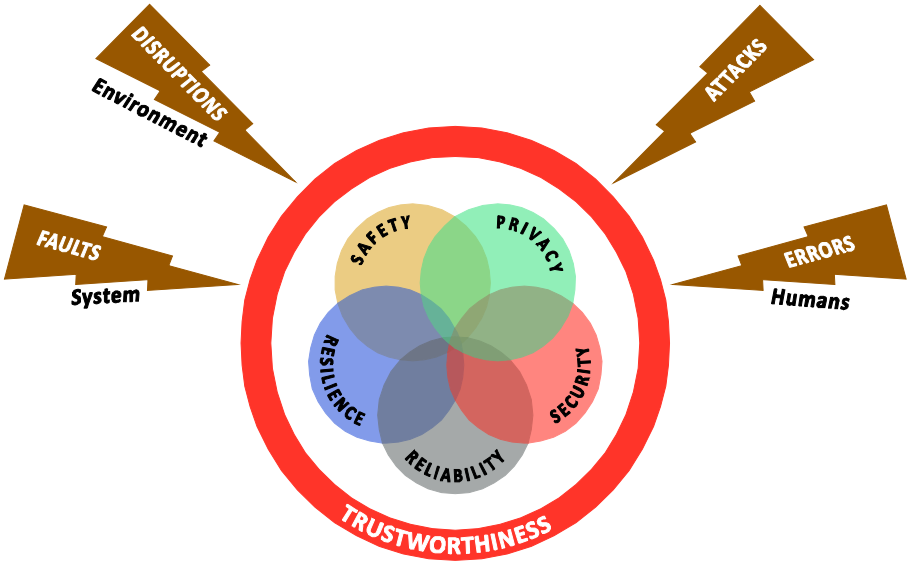

Figure 3-1: Trustworthiness of an IIoT System

Trustworthiness[1] is the degree of confidence one has that the system performs as expected in respect to *all* the key system characteristics in the face of environmental disruptions, human errors, system faults and attacks. The needs of IT and OT must both be met.

[1] Originally defined in [Schneider1998], again defined also in [NIST-CPS]

4 DISTINGUISHING ASPECTS OF SECURING THE IIOT

Traditionally, the security of Information Technology (IT) and Operational Technology (OT) systems has been evaluated independently, but an Industrial Internet of Things (IIoT) system is more than a simple merge of the two. Trustworthy IIoT systems require their security functions to be evaluated end-to-end across both IT and OT.

Integrating IT and OT security requires understanding the differences between them and their approaches to evaluating and protecting systems. Security, regulations and standards must evolve in both worlds and together to be effective. They can no longer focus narrowly.

4.1 CONVERGENCE OF INFORMATION TECHNOLOGY AND OPERATIONAL TECHNOLOGY

In the past, there has been a strong separation between IT and OT. IT covers computer and communication systems common across industries. Software applications are people-centric, and risks are often low. Real-time behavior is usually bounded by human interaction times, for example, how long someone will wait for information to be displayed.

OT, on the other hand, is a combination of hardware (initially) and software (more recently) that collects information and causes changes in the physical world through the direct monitoring and control systems. Control of physical systems, unlike IT systems, are task-specific, customized, automated and require less user interaction. In OT, real-time behavior can be essential for correctness, which may affect the type of security controls implemented.

Converging IT and OT involves a complex merge of their key system characteristics. Though many industrial systems are combining IT and OT to control devices by software, these systems are usually isolated on the OT side. Bringing these systems together modifies the security implementation both in IT and OT. For example, preserving information integrity stored in the cloud may affect OT system reliability and so becomes a matter of safety. If the control information stored in an IT system is modified without authorization due to incorrect security implementations, the OT system relying on these data may fail.

Convergence of IT and OT also brings different drivers and attitudes. Few IT specialists consider safety in their designs, while safety is not optional in OT. IT generally focuses on cost reduction once quality requirements of the system are met and may not have the resources to improve the safety quality of the system. More generally, key system characteristics and their assurance have different priorities in the two worlds that must be reconciled.

This convergence requires that the various functions that execute in the IIoT system always be considered together. It is for that reason that the 'Industrial Internet Reference Architecture' [IIC-IIRA2016] merged IT and OT functions into a set of functional domains (control, operation, information, application and business) that cover what needs to be done, rather than where it has been done in the past.

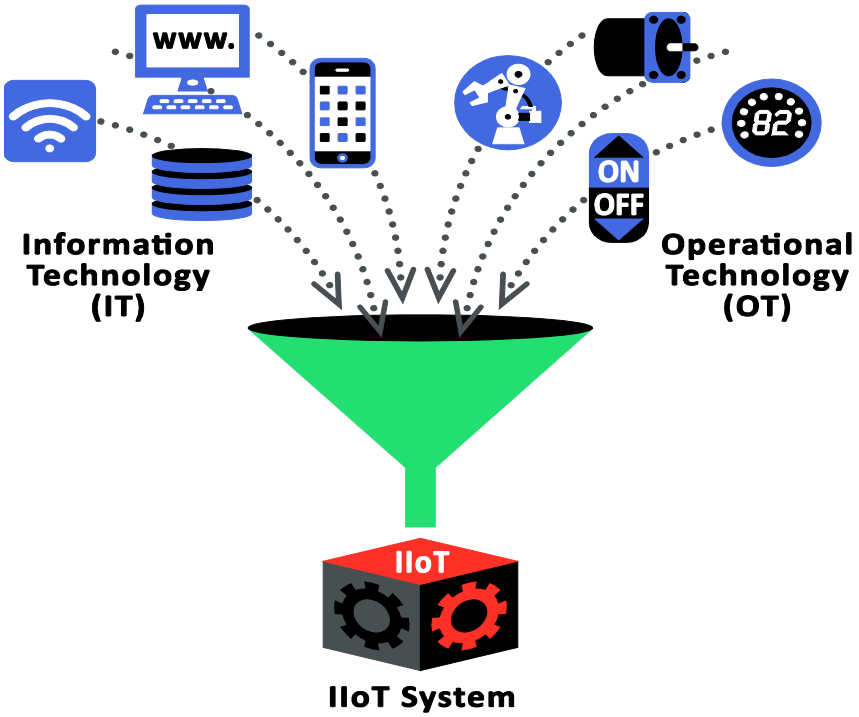

Figure 4-1: IT/OT Convergence

4.2 Security Evolution in IT and OT

Security to date has been mostly IT-centric. This view comes with some implicit assumptions about how risk is managed, that is, endpoints are adequately secured and communications between machines are protected. IT often assumes a client-server model, where clients and servers run multiple processes, and communicate using a well-known set of protocols such as IP, TCP or HTTP. Because of this homogeneity, security controls and monitoring assume a range of well-known attacks and attack models.

The evaluation of risk in IT systems depends on the probability of a successful attack and the damage that would be caused, but this damage usually involves money or reputation and rarely accounts for other outcomes such as safety threats. As a result, from business decisions to implementation, OT security is overlooked. Attack types that are common in OT, such as physical attacks, are not part of policy, and network elements do not account for industrial protocols.

But now consumers are adding new devices, such as light bulbs or televisions, to IT networks using protocols from industrial systems that have been repurposed to control home appliances. These 'internet of things' systems focus on usability rather than safety, so while the technology drivers are the same as for the *industrial* internet of things, the business drivers and key system characteristic requirements are different.

OT systems in the meantime have added more IT components, in particular consoles that run device-management software. Even control systems not connected to a network are subject to IT attacks, such as software viruses on removable media.

4.3 REGULATORY REQUIREMENTS AND STANDARDS IN IT AND OT

Given the risks, it is unsurprising that governments have put in place wide-ranging regulations and require their compliance. Regulatory and compliance rules mandate controlling access to financial systems, protecting credit card information, upholding privacy expectations and protecting critical infrastructure. Decisions on the implementation and operation of an IIoT system must account for these externally imposed business policies, including strict safety requirements.

Many IIoT systems are subject to external regulations that require compliance, and these compliance requirements may include IT and OT regulations as discussed in Annex A.

A wider view of regulations will be needed. Those from the OT environment will have to expand their view beyond safety to include a broad view of security for widely networked systems. Those with an IT background will have to consider safety regulations, as well as considering how IIoT systems relate to security regulations. In both cases, privacy regulations are of increasing importance as data is collected and shared for storage and analysis.

New legislation will likely impose additional types of audit, assurance and compliance requirements on both OT and IT to cover IIoT. For example, HIPAA[1] in healthcare focuses on protecting the IT side, such as patient data confidentiality, but fails to cover endpoint protections, including X-ray machines and insulin pumps, which now are connected to the network and can be target of attacks, or even used to pivot into restricted networks.

Compliance requirements are based on standards that are heavily fragmented into IT and OT. Annex A in the appendix describes a wide range of standards and regulations that may apply to IIoT deployments.

4.4 BROWNFIELD DEPLOYMENTS IN OT

The term *brownfield* describes an environment where new solutions and components must co-exist and interoperate with existing legacy solutions. The term is used in contrast to *greenfield*, where legacy systems are absent, removing such constraints.

OT systems are often deployed as brownfield due to the size and capital expense involved in building and retrofitting the industrial processes they encompass. Assets are often very long-lived, and reflect massive investments in operational, reliability and safety testing. It is therefore neither economically nor technically feasible to replace existing equipment and applications wholesale with newer alternatives in the short- or medium-term.

Most industrial installations contain equipment that by IT and security standards is “old” or “out of date.” Such equipment is at greater risk of attacks than equipment with the latest versions of security features and the latest security updates applied, deeply affecting security.

[1] See [HHS-HIPAA]

Many systems today still rely on physical security (locked doors and guards), isolation of OT networks and the obscurity of industrial protocols to compensate for a lack of cyber-security. But it doesn't work. For example, wired and wireless networking circumvents traditional physical controls such as doors and walls, because the network extends past physical boundaries.

From the attacker perspective, legacy OT systems are now a desirable target. Many industrial systems are routinely breached[1] due to out-of-date security protections. Eventually attackers will devise blueprints for monetizing OT breaches and the rate of attacks will increase dramatically. Attack payloads for complex OT end-points (such as nuclear centrifuges) have only been available to nation-state players, but that may also change.

Finally, traditional OT systems were designed to operate industrial processes safely and reliably without any communication to any external network. As IIoT systems incorporate OT components and subsystems created without security in mind, they may have unpredictable behaviors due to reuse or repurposing of their components. IIoT practitioners need to consider feature and function interactions carefully to address these concerns. Implementing security for existing brownfield OT environments should be as non-invasive as possible. Network perimeter protections such as firewalls, data diodes and routers, and passive network intrusion detection technologies that detect undesired activities, must be carefully deployed to enforce isolation between the OT control environment and networks external to the control systems.

4.5 CLOUD SYSTEMS IN THE IIOT

One of the benefits of the IIoT is the possibility of analysis and control of the OT infrastructure using external networked computing power. This practice of using remote servers to store, manage and process data, rather than a local server or computer, is called *cloud computing*. Organizations such as the Cloud Standards Council and the Cloud Security Alliance[2] offer ample guidance on the architecture and security of cloud computing. We focus here on the distinguishing aspects data-cloud systems need to account for in IIoT systems.

In a typical IIoT system, thousands of devices communicate with a cloud system, and possibly store data on them. Using shared third-party service providers creates a number of *trust boundaries* that can affect security and privacy. Information must be protected for security and privacy. Information flowing into control systems must be adequately secured to protect the safety and resilience of physical processes. For example, stolen credentials may allow attackers to control physical infrastructure remotely and facilitate attacks on many of the vendor's customers simultaneously. Moreover, attacks on other cloud customers or the platform may propagate, allowing attacks on the process owner.

4.6 IMPLICATIONS FOR SECURING THE IIOT

There is a need for an evolution in both business and implementation as it relates to security. From the business perspective, we look more closely at how risk is managed. Regulatory

[1] See [SANS-SSCS]

[2] See [CCSS-AIOT] and [CSA-IOT]

constraints apply to both OT and IT safety and security systems, and equipment that involves human or environmental safety must be certified at some cost in time and money. New attacks and threat models must be evaluated, and security programs should include all stakeholders. These stakeholders may have complex roles in securing IIoT systems, with different systems' boundaries implying different business models—and risk models. When single owner/operators controlled an isolated system, there was one boundary with clear security concerns. In IIoT systems, increased connectivity requires exposing more interfaces and that implies risk.

Most OT systems depend on infrastructure with lifetimes measured in decades, while IT systems can be upgraded frequently at little or no cost. In the upcoming years, these systems need to be integrated into an evolving landscape of endpoint, communication, monitoring and management systems that provide the required security. Safety-critical systems now are connected to the cloud for management and analysis of collected data.

Part II: The Business Viewpoint

Effective business decision-making is an important component of industrial security programs. Security risks, as well as the costs and benefits of different defensive postures, should be communicated to business decision makers, especially as they are often unfamiliar with the details of security risks and countermeasures.

IIoT system manufacturers, system integrators, owners and operators should establish and maintain a security program that provides governance, planning and sponsorship for the organization's security activities. These activities should align with the overall business objectives and risk strategy of the organization. Such a security program should keep policies, mechanisms and associated security processes up-to-date in response to changes in business priorities and resource availability, new risks and new protection goals.

Investment in IIoT systems and their operations must be protected against the risk of damage. This damage may include interruption or stoppage of operations, destruction of systems, and leaking sensitive business and personal data resulting in loss of intellectual property, harm to the business reputation, and loss of customers. But heightened security may lead to additional investment and greater times to deploy. It may affect user experience negatively. These additional costs must be justified to stakeholders by reference to the business risks they are taking and the costs saved by averting damages.

Industrial systems security engineering protects systems from errors, mischance and malice by consistent, comprehensive and well-defined operational procedures and protection policies. These policies must be informed by protection goals, risk strategy and business priorities with protection mechanisms to realize them with high-assurance.

An evaluation framework enables organizations to evaluate security capabilities consistently, communicate the capability levels meaningfully and prioritize security investments. (This framework is used internally and is different from a security audit.)

Managing risk is an important goal of a security (and privacy) program. This often consists of deriving an adversary model, then evolving a threat model and finally defining the security controls and capabilities to manage the risk taking into account the lifetime of the system. These models and decisions should consider the parties with different roles in the system (i.e., equipment vendor, system integrator or operator).

5 MANAGING RISK

Maintaining business value requires safeguarding the business investment in Industrial Internet of Things (IIoT) systems and protecting their operations from risk. *Risk*, the effect of uncertainty on objectives, takes into consideration the likelihood of an event occurring along with the impact of that event were it to occur. Elements of security risk that address the likelihood of an event occurring include threats and threat actors that may attempt to exploit vulnerabilities in the system unless countermeasures are deployed to mitigate the risk. Threats may be inadvertent (from hazards) or intentional (from attackers). Several elements of risk define the impact of an event, including the value of the asset (for example, the replacement cost of equipment or the revenue loss from equipment downtime), reputation damage, potential liability concerns, and physical and safety consequences of misoperating physical processes.

As it is not feasible to eliminate all risk from a system, we must manage risk so security investments are balanced against the effect of undesirable outcomes. This balancing must be grounded in a realistic assessment of the threats, the risks they pose and how they might prevent the system from fulfilling its intended functions. Costs must be evaluated and a rational selection of implementation choices made to deliver an acceptable return on investment.

It is possible to proceed with no security, and accept all the risk. It is also possible to spend exorbitant sums on security to the point that it no longer justifies the security gains. To manage risks, the organization should evaluate them, decide which parts of a security program in which to invest, deploy and periodically reevaluate both risks and the effectiveness of the program.

Security risk can be addressed in a variety of ways:

*Risk avoidanc*e seeks to eliminate the risk entirely to avoid all exposure. Often, complete risk avoidance can only be achieved by removal of the functionality causing the risk.

Risk mitigation implements compensating measures to reduce the impact of unavoidable threat. Mitigation is the most applicable strategy when risk avoidance cannot be achieved. It is implemented with a systematic approach to software security, audit and patch management.

Risk transferal transfers risk to a third-party. Most commonly this is in the form of insurance, where the risk is accepted by the third-party in return for payment. Transferring risk is a common technique for high-impact, low-frequency incidents that have unacceptably high mitigation costs. Risk transfer may also be achieved by passing the costs on to customers, or as an aspect of outsourcing.

Risk acceptance does not reduce the risk; it simply means one accepts it. This strategy is usually applied when the cost of the mitigation exceeds the cost of an adverse incident, should such an incident occur.

Residual risk is the risk that remains after all countermeasures have been implemented. When all known vulnerabilities are removed, there are still unknown ones. Risk may remain due to incorrect assumptions about system security or trusted personnel. Residual risk must be tracked to prioritize additional security operations, justify the security choices made and determine when

a balance has been struck in cost versus effectiveness of security controls. Applicable metrics help observe shortcomings continuously so as to create and apply corrective actions in a timely and efficient manner. In turn, the metrics may also change.

Effective business decision-making is an important component of industrial security programs. Security risks, as well as the costs and benefits of different defensive postures, should be communicated effectively to business decision makers, especially as they are frequently not familiar with the details of security risks or of countermeasures.

5.1 SECURITY PROGRAMS

Security programs encompass a range of technologies and activities essential to a comprehensive, robust security posture. The NIST '*Framework for Improving Critical Infrastructure Cybersecurity*' for example, has been adopted across many industrial sectors internationally.[1] It identifies five essential program activities:

- *Identify*: Develop the organizational understanding to manage security risk to systems, assets, data and capabilities.
- *Protect*: Develop and implement the appropriate safeguards to ensure delivery of critical infrastructure services.
- *Detect*: Develop and implement appropriate activities to identify the occurrence of a security event.
- *Respond*: Develop and implement the appropriate activities to take action regarding a detected security event.
- *Recover*: Develop and implement the appropriate activities to maintain plans for resilience and to restore any capabilities or services impaired due to a security event.

In this model, risk management is primarily a business process, while implementation is a technical and operational one. The implementation process provides asset, vulnerability and experience inputs to the risk management process, and the risk management process provides priorities, policy and budget decisions to the implementation process.

Risk is not static. The process to assess risk needs to be performed periodically. Changes in risk can come from:

- changes in the concept, value, or criticality of the system,
- changes in the physical composition of the system,
- changes in the threats to the system,
- adding assessment activities and addressing the findings from those assessments and
- adding new features or changing in existing capabilities.

System designers frequently have to choose between several options of technical, procedural and operational controls to address attacks. The decision process and metrics used in making the

[1] See [NIST-FICIC] and [NIST-FFAQ]

tradeoffs should be documented to support reexamination and independent verification. Decisions should be made based on future requirements across the lifetime of the system, not on the current threat landscape of today.

Deployment of security solutions needs to consider the legal ownership of the deployed IoT components and support systems. For example, the sensors or actuators may be deployed and owned by one legal entity, the IT components by another, while the data may be owned by yet another. In addition, a distributed deployment may be under multiple jurisdictions. Therefore, challenges posed by the ownership of equipment and data by multiple legal entities also need to be considered in assessing risks to the security of the deployed systems.

Deployment of security solutions must also consider operations, since security implementation that hinders them tend to be circumvented by operators, rendering the security measures ineffective. Security controls that frequently generate false-positive security alerts, require repeated user authentication or prevent the user from operating in the expected manner impede business productivity and should be discarded lest the installer be asked to remove the offending security controls. Security capabilities must be efficient, accurate and provide demonstrable value in contributing to defensive postures.

Other security programs may organize specific capabilities into different groups or elements based on traditional methodologies used in a specific sector or organization. For example, the North American Electric Reliability Corporation's Critical Infrastructure Protection standard (NERC-CIP standard[1]), which lays out the security program approach for electric providers in their jurisdiction, has organized the activities into eleven elements, some of which are unique to securing an electric system. Another example is European Union Agency for Network and Information Security (ENISA)'s Security Framework for Governmental Clouds[2] that offers four high-level phases with fourteen elements spread across them.

Several methodologies exist to assess security programs, the security posture of organizations and their process for secure development and maintenance of their products. Examples are provided in Annex A.3. IIC plans to utilize the Cybersecurity Capability Maturity Model (C2M2) model, discussed in Annex B, in IIC testbeds.

5.2 RISK ASSESSMENTS

Business risk is defined by IIC as "the effect of uncertainty on objectives" and information security risk as the "potential that a given threat will exploit vulnerabilities of an asset or group of assets and thereby cause harm to the organization."[3] Information security uses the term *threat* instead of the more general term *uncertainty*.

A risk assessment is the process by which risk, and specifically information security risk, is characterized. Approaches to information systems risk assessment are documented in many

[1] See [NERC-CIP]
[2] See [ENISA-SFGC]
[3] See [IIC-IIRA2016]

standards. IIoT risk assessments are unique in that they include physical consequences of errors and attacks as well as classic information systems risk.

The most commonly discussed threats come from malicious attackers who wish to disrupt a system, steal information or cause harm or fear, but even an adequately secured system must account for failures in the operating environment, such as extreme environmental or weather conditions. The term *threat,* then*,* should be interpreted broadly to include any influence or incident that would interfere with the normal, intended use of the underlying system.

While it is not practical to anticipate every possible threat, a strong security model that contemplates broad changes in the operating environment can mitigate the impact of many unplanned situations.

Identifying threats and consequences requires an understanding of the overall system and its implementation. The elements of the IIoT system exposed to possible attacks are called its *attack surface*. Growth in the number of technologies and increased complexity both increase the attack surface and vulnerabilities of the system, increasing risk.

Each of these elements may be vulnerable via an *attack vector*, a mechanism by which an attack can take place. Attack vectors include physical attacks, networks attacks, attacks against software, attacks on operators and attacks on the supply chains of the elements that comprise the system. Each industry has a specific set of attack vectors, as does each class of technology. The impact of each type of attack depends on the system's industry, design and business priorities.

Practitioners carrying out risk assessments should consider physical consequences of threats related to safety and the consequences of tampering with physical control equipment, as well as threats to analog and digital control systems.

The existence of some physical components may increase the attack surface by being more susceptible to tampering. For example, equipment exposed to the public has a greater attack surface than equipment behind security perimeters at dedicated industrial sites. Moreover, digital systems designed to prevent equipment damage or injury to workers will warrant increased attention in the risk assessment process, especially when safety risks are not mitigated with additional physical protections.

Physical safety systems may mitigate some attacks. Examples include over-pressure valves, flare stacks, berms and containment systems. Some of the work of assessing the effectiveness of these physical safety systems in light of attack scenarios may already have been carried out as part of the system's safety engineering assessment.

Modern attacks span a wide range of possible means and motives. Approaches to enumerate cyber threats and attack methods include lists of attack vectors such as OWASP, or the STRIDE threat identification approach and threat modeling.

5.2.1 OWASP IoT Attack Vectors

A non-exhaustive collection of potential attack vectors for IoT systems has been compiled from various sources by the OWASP[1] IoT project, which also compiles attack vectors for web applications. Each organization, depending on risk tolerance, needs to evaluate the list below to understand which vectors are applicable. Countermeasures and mitigations to address these attacks need to be confirmed by formal evaluations that may include static analysis, dynamic testing, fuzz testing and penetration testing.

The OWASP IoT Top Ten List[2] includes:

1. Insecure web interface
2. Insufficient authentication/authorization
3. Insecure network services
4. Lack of transport encryption
5. Privacy concerns
6. Insecure cloud interface
7. Insecure mobile interface
8. Insufficient security configurability
9. Insecure software/firmware
10. Poor physical security

This list is a good starting point for questions and analysis and to understand the relevant attack types, but it is not complete and consequences of attacks between IoT and IIoT differ.

5.2.2 STRIDE Threat Model

STRIDE[3], developed by Microsoft, models risks and evaluates threats for the IT environment. The STRIDE model has also been extended to incorporate IoT threats[4] that are applicable to IIoT systems. The STRIDE model comprises several elements:

An *adversary* (STRIDE's term for an attacker) is a malicious entity whose goal is to prevent an asset from working as designed to compromise the integrity, availability or confidentiality of a system or its data. Adversaries exploit vulnerabilities in assets. This process comprises a threat. The threat model describes the set of possible attacks on an asset. These threats are then classified based on severity, and the potential countermeasures can be evaluated.

Spoofing identity: This is a type of threat where a person or device is using another person's credentials such as login and password. A device can use a spoofed device ID.

Tampering with data: Altering the data related to a device or traversing the network.

[1] See [OWASP]
[2] See [OWASP-IOT]
[3] See [MS-STRIDE]
[4] See [MS-STRIDE-IOT]

Repudiation: Denial that a person or device was involved in a particular transaction or event. This refers to the ability (or lack thereof) to trace which person or device was responsible for an event.

Information disclosure: Exposure of information to individuals who are not supposed to have access to it. In the Industrial Internet, this could mean sensor data for a smart city in the hands of persons with intentions to launch an attack on the city.

Denial of service: This refers to making a particular service unavailable, often through resource consumption or unreliable execution.

Elevation of privilege: An unprivileged user gains sufficient access to compromise or destroy an entire system. In elevation of privilege threats, an attacker has penetrated all system defenses and become part of the trusted system itself, a dangerous situation indeed.

These last six items make up the acronym STRIDE.

5.3 COMMUNICATING RISK

Effective business decision-making is an important component of industrial security programs. The costs and benefits of different security risks and defensive postures should be clearly communicated to business decision makers, especially as they are often unfamiliar with the details of security risks and countermeasures.

There are three basic methods for communicating risk:

Quantitative risk assessment expresses the risk of an incident as the product of probability of that incident occurring and the cost of consequences of the incident. This approach works well for high-frequency, low-impact events. Systems with large numbers of devices, where the cost of compromise is comparatively low, and system compromise occurs frequently enough to produce statistically significant estimates of probability, are well-served by quantitative risk assessment. Quantitative risk assessment is much less effective for communicating the risk associated with low-frequency, high-impact events. For example, in IIoT system where the cost of compromise is high, and there is no way to make statistically significant predictions of the probability of such a disasters occurring in the future.

Qualitative risk assessment uses surrogates for cost or probability estimates, and expresses risk as a mathematical function of these qualitative surrogates. For example, the French ANSSI standards[1] calculate the importance of an industrial system by assigning a small integer rating to each of *consequences, likelihood, system complexity/functionality, connectivity, exposure and accessibility*. These rankings are combined arithmetically to produce a number between one and three describing the importance of an industrial control system. Minimum-security measures are then prescribed for each class of control system. Other qualitative systems assign and calculate qualitative metrics for individual risks and kinds of incidents.

[1] See [ANSSI-CMKM]

Systematic approaches develop models of possible consequences and possible attacks. *Fault-tree analysis* is a method for understanding how lower-level events might combine into significant undesirable outcomes. *Attack-tree analysis* is a method for understanding how attack vectors exposed in individual IIoT components might be combined to bring about a specific compromise.

Risks due to low-frequency, high-impact events can be difficult for business decision makers to evaluate. Qualitative risk scores are used widely to evaluate such risks, but qualitative scores can be difficult to relate to security budgets, return-on-investment and the risk appetite of an organization.

Some security teams communicate such risks by selecting and communicating a representative set of attack scenarios with significant adverse consequences that are not defeated with a high degree of confidence, given the organization's current security posture. Business decision makers often find specific, representative attack scenarios easier to understand and evaluate than abstract qualitative scores. Specific representative attack scenarios allow them to select those they believe should be better addressed by the security program, and to compare the cost of upgrading security systems to the cost of the consequences of those specific scenarios.

5.4 ONGOING BUSINESS ATTENTION

Updates to security-related technologies are often overlooked as organizations focus on desired functionality. Ongoing attention to the key system characteristics of the system as they are used in operations must be adequately planned for, resourced and managed.

Because attacks change over time, security should be subject to periodic review. The rate of change in the techniques, maturity and focus of attacks varies across various types of technologies and the business sectors and verticals they support. The maximum reasonable interval for these reviews should be selected during the system design process based on the business model appropriate for the kind of system under construction. Periodic reassessments and changes may be needed to address issues found during those reviews. More frequent reviews and updates of security countermeasures may be required based on the emergence of new threats or regulatory changes on top of the operational updates and product revisions driven by vendor's release of software fixes and updates.

The periodic security reviews should follow the same process as that used in the original conceptualization, design, creation and deployment activities for the system. The original lists of significant threats to the operations, usability, safety and other business needs should be revalidated and updated if necessary. Existing countermeasures should be revalidated against current industry best practices.

With an accurate record of the original design decisions, made to meet the claimed capabilities and the evidence used to support the architecture, design and technology choices made, this review will take minimal effort.

5.5 METRICS AND KEY PERFORMANCE INDICATORS

Business decision makers should monitor reports on the security of their IIoT systems from the moment the systems are conceived, through their design and creation, and throughout their operation. This should be at the same depth as they monitor other characteristics such as performance, throughput, cost and efficiency. The correct measures and metrics inform decision makers, operators and other stakeholders. The interests and needs of key stakeholders, legal responsibilities from laws, regulations and contracts, as well as norms of behavior in the industrial sectors of the system, should all be taken into consideration in establishing appropriate metrics and baselines (metrics define quantitative results against a baseline and measurements describe an absolute observation). All of these considerations should be reviewed periodically for possible adjustment.

Some of the metrics and measures will be common across verticals; others will be unique. As an example of the former, most industries track security metrics such as the number of detected attack attempts, and the breakdown of those attempts, as well as characterizing successful attacks, incidents, close calls, policy violations and anomalies that have merited investigation. For the latter, in the utility and energy industry, it is important to collect metrics on remote terminal units (RTUs) and sensor outages. The function of those metrics is to identify an outage in an RTU quickly, visualize it on a display and set up a process to investigate whether the outage was malicious or an accident.

Clear and accurate representations (dashboards and other visualizations) of security metrics, including data sources, communications and system capabilities, as well as key performance identifiers allow operational and business personnel to make improved business decisions. Security then becomes a valuable part of the operational process, and its value can be quantified in terms of the costs saved by averting wrong decisions.

Security metrics can set up a continuous feedback loop to identify areas of risk, increase accountability, improve security effectiveness, demonstrate compliance with laws and regulations and provide quantifiable inputs for effective decision making. Such metrics help identify security problems early and assist in faster and more efficient management and governance. Key performance indicators selected for each application also improve the quality of service as issues such as the number of times a capability is disrupted can be identified early, and corrective or compensating measures taken. Dashboards and other visualizations displaying security metrics collected through continuous feedback loops are desirable, but not essential to conduct periodic risk assessments.

5.6 MANAGEMENT CONSIDERATIONS

Managing risk balances the threats against the IIoT system with the security responses that counteract those threats and the risk they represent. Risk management involves *ongoing action* for making the appropriate decisions based on the security evidence from metrics and key performance indicators (KPIs) as well as monitoring data to prioritize security tasks. Building out a feedback loop to identify security issues attest that those issues have been correctly addressed is highly recommended.

Managing IIoT security involves *coordinated action* within the organization, and focus on *rapid response* to ensure timely execution of security tasks.

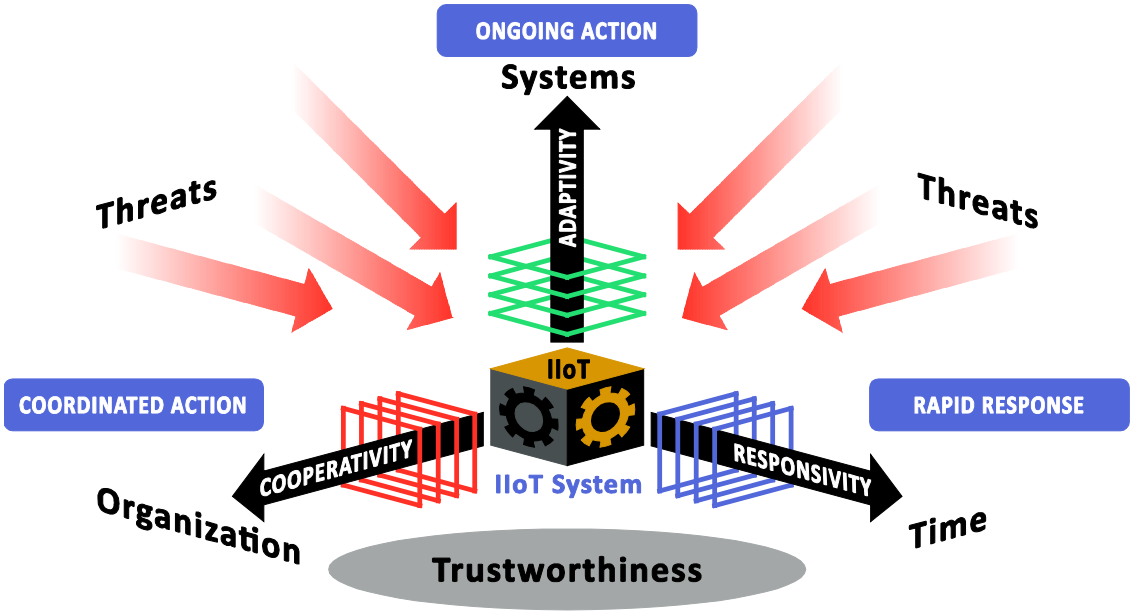

Figure 5-1 Trustworthiness Management Considerations

There is considerable complexity in orchestrating security responses, and the problem space quickly becomes a multidimensional challenge. Security measures should be able to adapt as needed to continually changing threats and system configurations (adaptivity), provide responses that will minimize the impact on the IIoT system if a security threat does materialize (responsivity), and enable different organizations work together to ensure the early identification of security threats (cooperativity).[1]

Security must be adaptable to maintain effectiveness over time. Security management must adapt to the changes in the environment, new threats to which the system is exposed, and new vulnerabilities that are found. Threats should be dealt with before they materialize.

To achieve this, management, engineering, operations and human resources need to cooperate continually. In addition to choosing the improvements to make based on systematically identifying the new threats and other changes to which the system is exposed and assessing the risks they present, they must establish improvement practices for cyberspace, physical space, and operational management, and formulate implementation plans based on these improvement practices. An important factor in this work is risk assessment.

Formulation of a rapid response when an incident occurs, based on an assessment of the scope of the security threat, is important to protect systems and minimize damage.

[1] See [HIT-ISA-65-5] and [IEC-FOTF]

6 PERMEATION OF TRUST IN THE IIOT SYSTEM LIFECYCLE

A typical Industrial Internet of Things (IIoT) system is a complex assembly of system elements. The trustworthiness of the system depends on trust in all of these elements, how they are integrated and how they interact with each other. *Permeation of trust* is the hierarchical flow of trust within a system from its overall usage to all its components.

Each IIoT system has a unique permeation of trust. Each element has *actors* (designers, developers, manufacturers, operators etc.) that execute the various roles in the creation, integration and usage of the hardware and software of an IIoT system. These roles cut across multiple organizations, each with its own interests.

Permeation of trust cuts across the complete system lifecycle, not only operation. It depends on the integrity of the *chain of custody* of each element of the system and its data. Everything from supply chain, commissioning, provisioning, regular usage and end-of-life decommissioning must be carefully monitored to ensure the initial trustworthiness is preserved throughout.

6.1 SYSTEM LIFECYCLE

Figure 6-1 shows the permeation of trust from an industrial operator, such as a hospital or a nuclear power plant, throughout the hardware and software that makes up the system. This trust should be explicitly described, verified, controlled and supervised and not be based only on the reputation of the vendor, without validation that the trust is warranted.

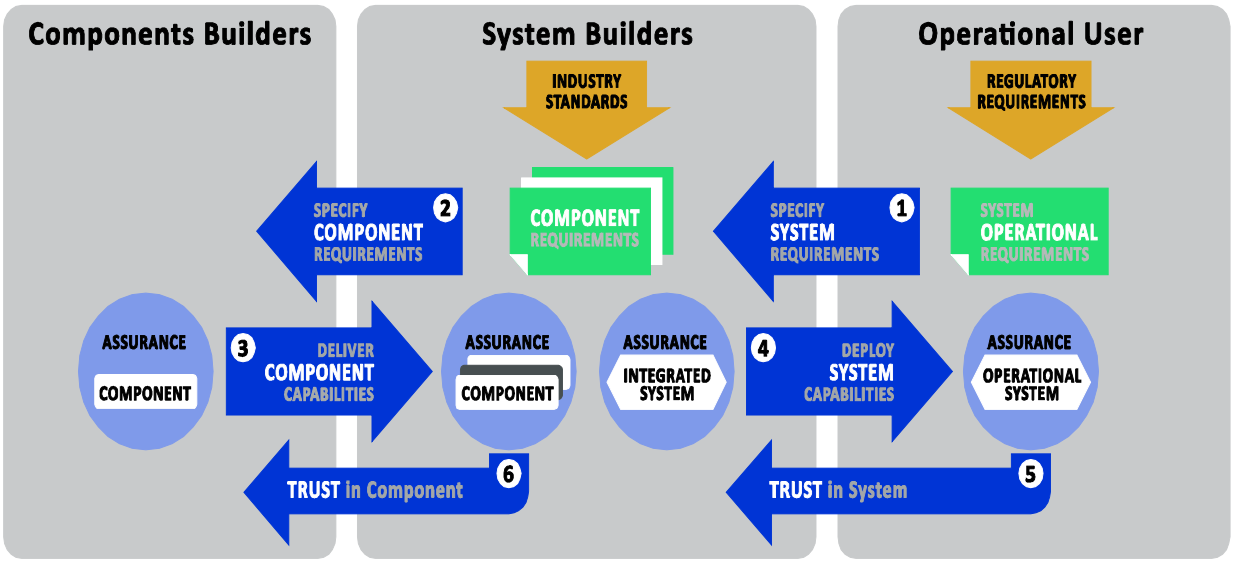

Figure 6-1: Permeation of Trust

The trust lifecycle starts with the specification of requirements that result in the delivery of capabilities. The assurance that these capabilities meet the stated requirements becomes the basis of trust in the system.

System owners and operators instigate the building of trust by specifying trust-related requirements as part of the operational system requirements. These requirements are then issued to the system builders as part of the system specification. System builders in turn break them down into specific trust requirements for each of the components of the system. Component builders respond to these requirements by delivering components that meet the specified requirements.

Compliance of the delivered component capabilities to their specifications is a part of assurance performed by the vendors prior to delivery, by system builders on receipt and probably by (potentially independent) third-party agencies. System builders are responsible for integrating all the assured components and assuring that together, they meet the specified requirements for the integrated system. The delivered system capabilities are verified and assured in the operational context by the owner/operator, or an independent third party.

Once operational assurance is achieved, trust is initiated in the system and permeates down from the owner/operator to the component builders via the system builders.

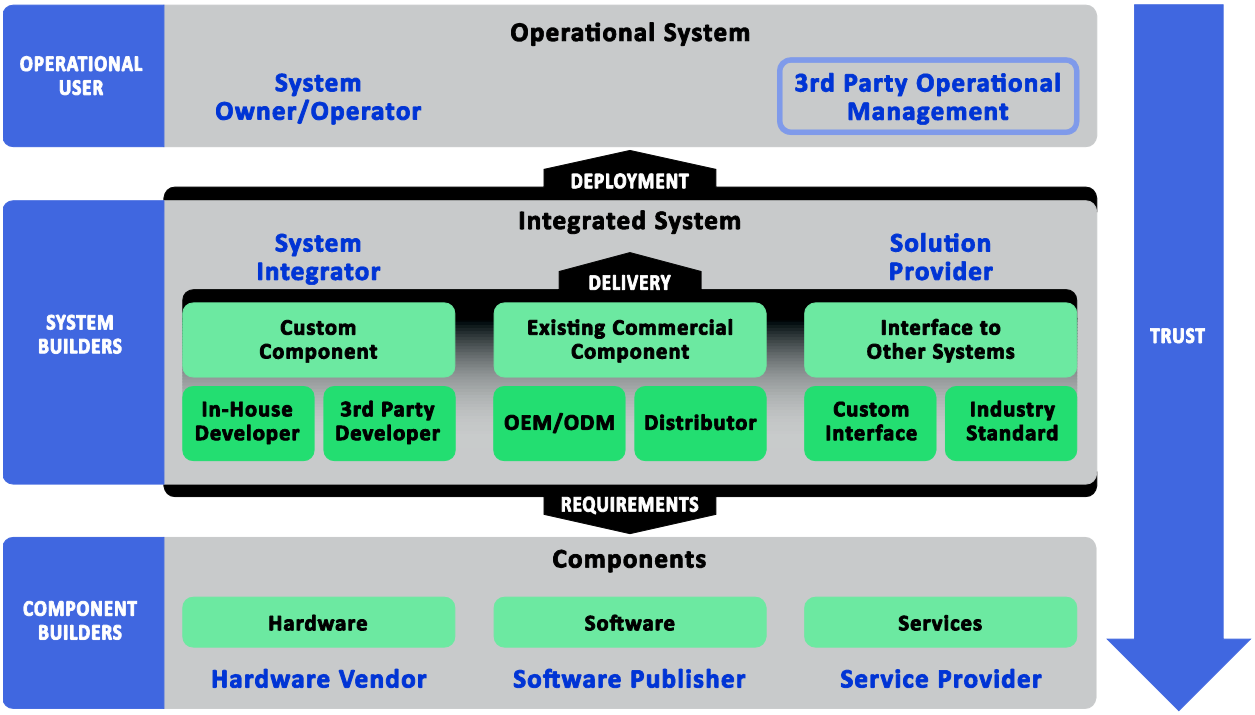

Figure 6-2: Trust Relationship between Actors

Trust flows down from the owner/operator to all parts of the system but trust must be built from the bottom up. Figure 6-2 also shows the case where the owner/operator may assign the overall operation management of the system to a third party. Either way, the owner/operator is responsible for ensuring the delivered system continues to deliver its business purpose while meeting operational requirements and maintaining stated levels of trustworthiness.

To establish the trust relationship, the receiving actor delivers requirements, so defining the expected capabilities that will be delivered by the delivering actor below. This process may face numerous challenges:

- Requirements may be difficult to elicit and hard to describe for exceptional situations.
- Requirements may change during the design phase or during the lifecycle of the system.
- Background knowledge and technical terminology may be different; in international relationships the communication language can lead to confusion.
- Small mistakes in defining the requirements for subcomponents can escalate during the permeation of trust to large problems for the whole IIoT system.

Standards, provided by organizations like ISO, IEEE, UL, IEC and government organizations simplify the communication. Requirements and capabilities are clearly documented, so reference to one or more well-described standards can define the system. Over time, the standards should define the expectations on implementing trust sufficiently in the system.

There are explicit and implicit requirements. *Explicit requirements* define features that are needed by the system, such as when a water pump stops due to achieving a certain water pressure. *Implicit requirements* are characteristics of the system, such as quality, composability, manageability, security, resiliency. An example for an explicit requirement is a low price; an example of an implicit requirement is high trustworthiness. A practical example for an explicit requirement in a modern air conditioning system is that the temperature can be controlled via the internet; an implicit requirement is that internet access cannot be abused by hackers. Receiving actors focus on explicit requirements and tend to neglect implicit requirements that can lead to a compromise of trust. Most standards focus on implicit requirements; another reason to use standards while defining the requirements.

Since the delivering actor provides the capabilities, the trust that the capabilities entirely fulfill the requirements is based on assurance. The receiving actor should assure the trustworthiness of the system by carefully studying the documentation describing the capabilities, and then validate them through thorough practical tests in an environment similar to the intended environment for actual operation. This process can be expedited by consulting an independent third party, for example a qualified test laboratory, to verify the assurance within the design.

Unfortunately, receiving actors frequently neglect such systematic confirmation of the capabilities because of limited time or resources. Trust is then based on the existing relationship to the delivering actors or even the history and “good name” of the deliverer only.

Since the trustworthiness of the system depends directly on the implementation of the trust mechanisms at each step in the process, the trustworthiness is the composition of the key system characteristics implemented in each component *and the assurance thereof*.

6.2 Roles in the Permeation of Trust

Permeation of trust can be structured in terms of the role a specific actor has, in one of three different layers that are shown as rows in Figure 6-2:

- *Component builders* are hardware vendors, software publishers and service publishers who provide specific capabilities as a standardized product or service.
- *System builders* are system integrators and solution providers who integrate or adapt these built components in usage-specific individual solutions or service capabilities.
- The *operational user* is the *system owner/operator* that uses the components, solutions or services for their intended purposes.

Once again, hardware, software and service components are built upon other components, so trust permeates from a base component up to higher-level components.

System builders are responsible for integrating components from multiple sources properly. The components may be delivered through many delivery mechanisms: custom development, *commercial off the shelf* (COTS) integration or integration of another system. Each of these approaches has their respective processes for assuring trust. For some types of equipment, such as medical, aeronautics, and railroad, well-founded and defendable assurance is addressed by assurance cases and supporting evidence.[1]

Trust in custom development environments relies on in-house or third-party developers to build components that comply with specified requirements. COTS integration requires verification for compliance of existing products with trust requirements. If the COTS components are not capable of delivering on those requirements, then system integrators may encapsulate or isolate the COTS components in environments capable of delivering the required level of trust. Integration of other systems depends on defining clear interface specifications or interface standards-coupled *service level agreements* (SLAs) that meet the specified trust requirements.

In each of these system-building approaches, system builders will need to integrate hardware, software and services components. The component builders must show that their respective components meet the specified trust requirements. When these components are an aggregation of other components, the builder of the main component is responsible for assuring that all the components and their integration meet the specified trust requirements.

The IIoT system owner/operator must trust that each prior step in the process has been implemented correctly to support the trust assumptions in the layers above him.

Each layer of the trust model depends on the one below it: Each actor builds a *trust relationship* with the actor below, following the schema of Figure 6-2. Trust is achieved in the operational system when assurance that the operational requirements of the system have been met. This trust then permeates back down through all levels of actors, which created, integrated or supplied components or sub-systems of the operational system.

The trustworthiness of the operational system produced by the manufacturers and vendors is transferred to the trustworthiness of the capabilities the system builders provide. These capabilities again are based on the trustworthiness in the integrated technical components.

[1] See [AAMI-TIR2014] and [NASA-CR2015]

There are cases where the boundary between the roles is less distinct. Members of one role may take on characteristics of an adjacent role. For example, some manufacturers may wish to maintain control over and manage the devices they produce. Device management, security management and predictive maintenance are example use cases where the manufacturer may wish to play the role of the system builder, specifically the third party operational management provider or the service publisher, in addition to the manufacturer role.

Similarly, some equipment owners and operators may wish to purchase directly from the manufacturer and integrate the equipment directly into their environment. In that case, the owner/operator is acting as the in-house system integrator, potentially even developing their own solutions in-house.

6.3 TRUST AT COMPONENT BUILDER ROLES

Manufacturers and vendors develop technical components to sell as standard. They can be adapted for specific usage, but this is the responsibility of the system builder. The deliverer of the component is responsible for delivering the capabilities that fulfill the anticipated and implicit requirements over the lifecycle of the component. The receiver of the component is responsible for assuring its trustworthiness at the next level of the trust hierarchy.

Trust must permeate down through all the components and their subcomponents, as shown in Figure 6-3. Component builders must ensure that trust requirements are applied to each of the subcomponents and their integration.

Hardware component builders must provide trust requirements and assure their compliance down the chain through the decomposition of all the subcomponents. For example, the *original equipment manufacturer* (OEM) delivering a controller is responsible for ensuring the trust of all the components ranging from the microprocessor, memory, peripherals, power supply and enclosure.

Some of these components might be delivered as integrated hardware and software subcomponents. For an example, a device may be delivered with a board that integrates the application processor, memory module, graphics process and integrate *unified extensible firmware interface* (UEFI) firmware.[1] Once again, the component builder responsible for the aggregated components is responsible for assuring compliance with the trust requirements.

[1] See [UEFI]

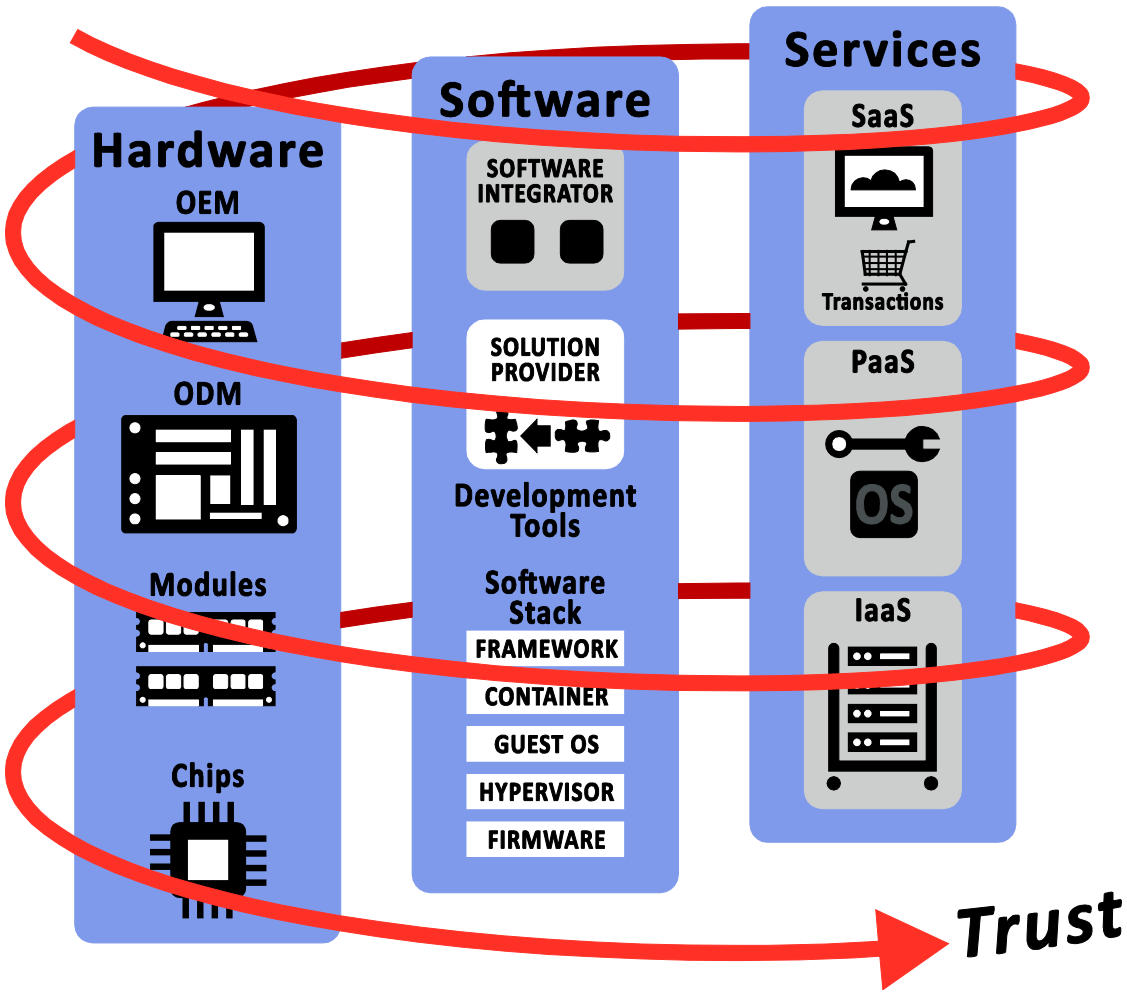

Figure 6-3: Trust Relationship between Component Builders

These components may be delivered in the form of a service integrating and exposing both hardware and software components. The trust in service components is assured by the fulfillments of the requirements of the SLAs by these components and their subcomponents. For *infrastructure as a service* (IaaS) such subcomponents may include hardware and low-level software components such as firmware and hypervisors. *Platform as a service* (PaaS) usually includes as subcomponents operating systems, system components such as databases and application frameworks. Finally, *software as a service* (SaaS) may have other software subcomponents running on a third-party platform. In all three of these service offerings, the main component builder is responsible for the permeation of trust through all the subcomponents of the service.

Vendors and manufacturers seek to implement incremental value-adds to products already in the market, and so maintain the return on investment on the research and development required to implement trust. However, if the manufacturer and vendor do not implement appropriate trust mechanisms, it is difficult for the system builders and equipment owner/operators to implement those mechanisms later on. The trust must be designed in from the beginning.

The trustworthiness of a technical component is not just defined as the sum of the trustworthiness of its subcomponents. It is the responsibility of the component developer to assure that the subcomponents are working correctly together with their specified capabilities. Weakness of a single subcomponent may lead to the loss of trust into the whole system. For example, one incorrectly selected hardware component with a smaller temperature range than specified for the system may lead to a complete system failure as soon the system temperature exceeds that component's temperature range. Or a single software component with limited security behavior may compromise the security of other software components and finally the entire system.

In operational technology (OT), safety certification requires the fulfillment of national and international standards and national law, which generally requires rigorous tests, typically confirmed by authorized independent test laboratories.

In information technology (IT), it is less common to implement rigorous safety compliance tests. However, it is becoming more common for components designed for the consumer market to be applied to industrial purposes, but their resilience may not be up to industrial standards. Moreover, the lifespan of products for consumer markets are usually much shorter than required in industrial usage. In either case, any shortcomings in the IT element's trustworthiness may have unacceptable negative effect on the OT process. Industrial-grade products are available, but they must be explicitly sought out.

When software publishers include software subcomponents, a patch may not be available because the publisher of the subcomponent no longer supports it. Even if the source code is available it may be difficult to understand and limited access to the required elements in the code-build environment may inhibit fixing any bugs.

Many software products have *application programming interfaces* (APIs) that other software products depend on. Software publishers and SaaS publishers must keep such interfaces consistent or at least backwards compatible during the lifespan of *all* IIoT systems that use them.

Many SaaS services are IT-based and human-interaction oriented. Small and frequent changes in user interfaces are easily accepted by must human users, but making such changes in remote API can diminish the trust in the SaaS publisher.

Replacing hardware components or updating software components during the lifetime of a system involves the risk of non-authentic copies, including illegal chips from gray markets or malicious modification of software during the update delivery process. The former can be addressed by adding unique serial numbers, registered with specific production dates, while integrity protection works well in keeping software updates authentic.

6.4 TRUST AT SYSTEM BUILDER ROLES

A component builder can stretch the cost of development and rigorous testing of sold components over time. A system builder, on the other hand, delivers an operation-specific system that must be cost-effective with the first design. As a result, it is common for the system

builder to create a framework that allows various technical components to be integrated into a platform to be resold at scale across a number of equipment owner/operators.

Because the design from a system builder is more customized than that of a technical component builder, it may be possible to address trust issues in specific components by applying mitigating controls. This does not eliminate the risk if the weakness was to occur during operation, but it reduces the likelihood. For example, a software component that is no longer updated by the software publisher (and without an alternative vendor) may contain a well-known security weakness from a network attack vector. This component may require network-based countermeasures such as firewalls, strict access controls, network intrusion detection or behavioral anomaly detection to ensure the component is not compromised.

While a technical component builder should never deliver an untested product to a customer, the system builder should perform external tests and certifications so uncovered weaknesses can be addressed with design modifications. The system builder is in the position to address trust issues that may have been delivered by technical component builders.

System builders have similar challenges as component builders. They have to assure that their built system fulfills the expectations during the whole lifespan of the system. Initially they are paid only for the design, installation and successful setup and sometimes to assure the continuation of the running system, but a system builder must be able to deliver functionality across the expected lifespan of the system. This includes not only replacing failed components but also keeping and maintaining the knowledge about the built system over this lifespan.

In many cases, an owner/operator buys technical components and just uses them. The usage of combined components is still a role of the system builder, but it has been merged with the role of owner/operator and named *in-house developer* in Figure 6-2. In larger companies, such in-house system building is frequently delegated to a department that has only one “customer” (the operator/owner). But if this department is dissolved, required maintenance cannot be performed and the system risks instability, uselessness and danger.

6.5 TRUST AT THE OPERATIONAL USER ROLES

The *operational user* is the starting point of the permeation of trust. The owner/operator of the operational system must assure at regular intervals (see section 5.4):

- that the system meets the stakeholder needs,
- all threats to the deployed system are assessed,
- the risk to the deployed system is quantified and approved and
- updates, countermeasures and mitigating controls are implemented to manage this risk during the whole lifecycle.

Sometimes owner/operators take their role as owners of the system to a level that was never intended by the system builder, for example by disassembling a system and reusing its components somewhere else. The owner/operator should always be aware that the trust in a

system is limited to the delivered capabilities that depend on the specified requirements. This realization would quickly exclude such an abuse of the system.

All capabilities of the IIoT system are finally delivered here and all trust in the system begins here. The system owner/operator carries the risk of the operational process. Any failures in the system trustworthiness, due to poor security, safety, reliability, resilience or privacy, will directly affect the owner/operator's business.

Failing in these assurances can threaten the existence of an owner/operator. History shows that most of the damage, lost revenue, litigation payments and responsibility for serious injury or death were assigned to the owner/operator because its trust into the delivery was too high and the requirements were not well-enough specified to hold the deliverer responsible.

Part III: The Functional and Implementation Viewpoints

An implementation of an Industrial Internet of Things (IIoT) system must provide end-to-end security from the edge to the cloud. This includes hardening of endpoint devices, protecting communications, managing and controlling policies and updates, and using analytics and remote access to manage and monitor the entire security process.

Ideally, security and real-time situational awareness should span Information Technology (IT) and Operational Technology (OT) subsystems seamlessly without interfering with any operational business processes. Security must be built into the design and risks should be evaluated early, rather than trying to bolt-on security as an afterthought.

But greenfield deployments using the most current and secure technologies are not always feasible. Since the average lifespan of an industrial system is currently 19 years[1], security technology must often be wrapped around an existing set of legacy systems that are difficult to change. In both greenfield and brownfield deployments, all affected parties—manufacturers, systems integrators and equipment owner/operators—must be engaged to create a more secure and reliable IIoT system.

As there is no single "best way" to implement security and achieve adequately secure behavior, technological building blocks should support a defense-in-depth strategy that maps logical defensive levels to security tools and techniques. Due to the highly segregated nature of industrial systems, security implementation needs to be applied in multiple contexts. Multiple sub-networks and differing functional zones may have different operating technologies and security requirements. Security tools and techniques built for IT environments may not always be well suited for OT environments.

IIoT systems may have constrained system resources that need to meet various requirements such as system safety and real-time execution. These factors may not allow implementing all security measures and controls to their fullest extent (as required by defense-in-depth strategy). The security program implementation considerations should take into account all the required functional and non-functional aspects of the system behavior, including their relative priorities.

IIoT system security should rely on automation as much as possible, but people must be able to interact with the security implementation to monitor status, review analytics, make decisions when needed and plan modifications and improvements. Usable management and control systems may contribute to security by reducing operator errors.

[1] See [NIST-800-82]

7 IISF FUNCTIONAL VIEWPOINT

7.1 SECURITY BUILDING BLOCKS

The functional viewpoint of the security framework comprises six interacting building blocks, as shown in Figure 7-1. They are organized into three layers. The top layer comprises the four core security functions: endpoint protection, communications and connectivity protection, security monitoring and analysis, and security configuration management. These four functions are supported by a data protection layer and a system-wide security model and policy layer. These three layers comprise the functional viewpoint of the industrial internet security framework.

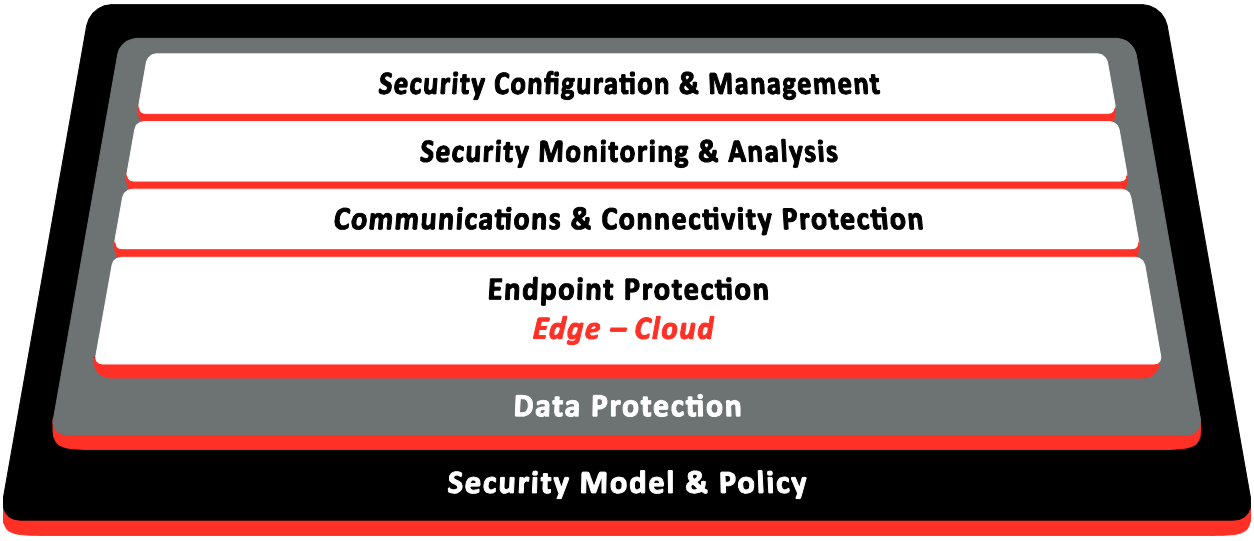

Figure 7-1: Security Framework Functional Building Blocks

Endpoint protection implements defensive capabilities on devices at the edge and in the cloud. Primary concerns include physical security functions, cyber security techniques and an authoritative identity. Endpoint protection alone is insufficient, as the endpoints must communicate with each other, and communications may be a source of vulnerability.

Communications and connectivity protection uses the authoritative identity capability from endpoint protection to implement authentication and authorization of the traffic. Cryptographic techniques for integrity and confidentiality as well as information flow control techniques protect the communications and connectivity.

Once endpoints are protected and communications secured, the system state must be preserved throughout the operational lifecycle by *security monitoring and analysis* and controlled *security configuration management* for all components of the system.

These first four building blocks are supported by a common *data protection* function that extends from data-at-rest in the endpoints to data-in-motion in the communications. It also encompasses all the data gathered as part of monitoring and analysis function and all the system configuration and management data.

Security *model and policy* governs how security is implemented and the policies that ensure confidentiality, integrity and availability of the system throughout its lifecycle. It orchestrates how all the functional elements work together to deliver cohesive end-to-end security.

We describe each of these functional building blocks and their interactions in this chapter. The specific implementation details, including security mechanisms and techniques, are covered in the implementation viewpoint, chapters 8 to 11.

7.2 IIoT System, IIRA Functional Viewpoint and IISF Functional Viewpoint

The functional viewpoint in 'Industrial Internet Reference Architecture' (IIRA, [IIC-IIRA2016]) describes the functional building blocks in the architecture of an IIoT system, how they interact and how they interface with the outside world. They include actuators and sensors and are connected to the physical assets at the edge, and optional business domain elements in the cloud (shown in blue as the middle layer of Figure 7-2).

These functional building blocks are implemented end-to-end from the edge to the cloud and supported by connectivity spanning both OT and IT (shown in purple).[1] As shown by the top layer with red contour, the IISF building blocks address security end-to-end across all the functional domains described in the IIRA.

[1] Although cloud deployment is called out, IIC does not endorse any specific deployment model and actual deployment of applications can occur anywhere in an IIoT architecture.

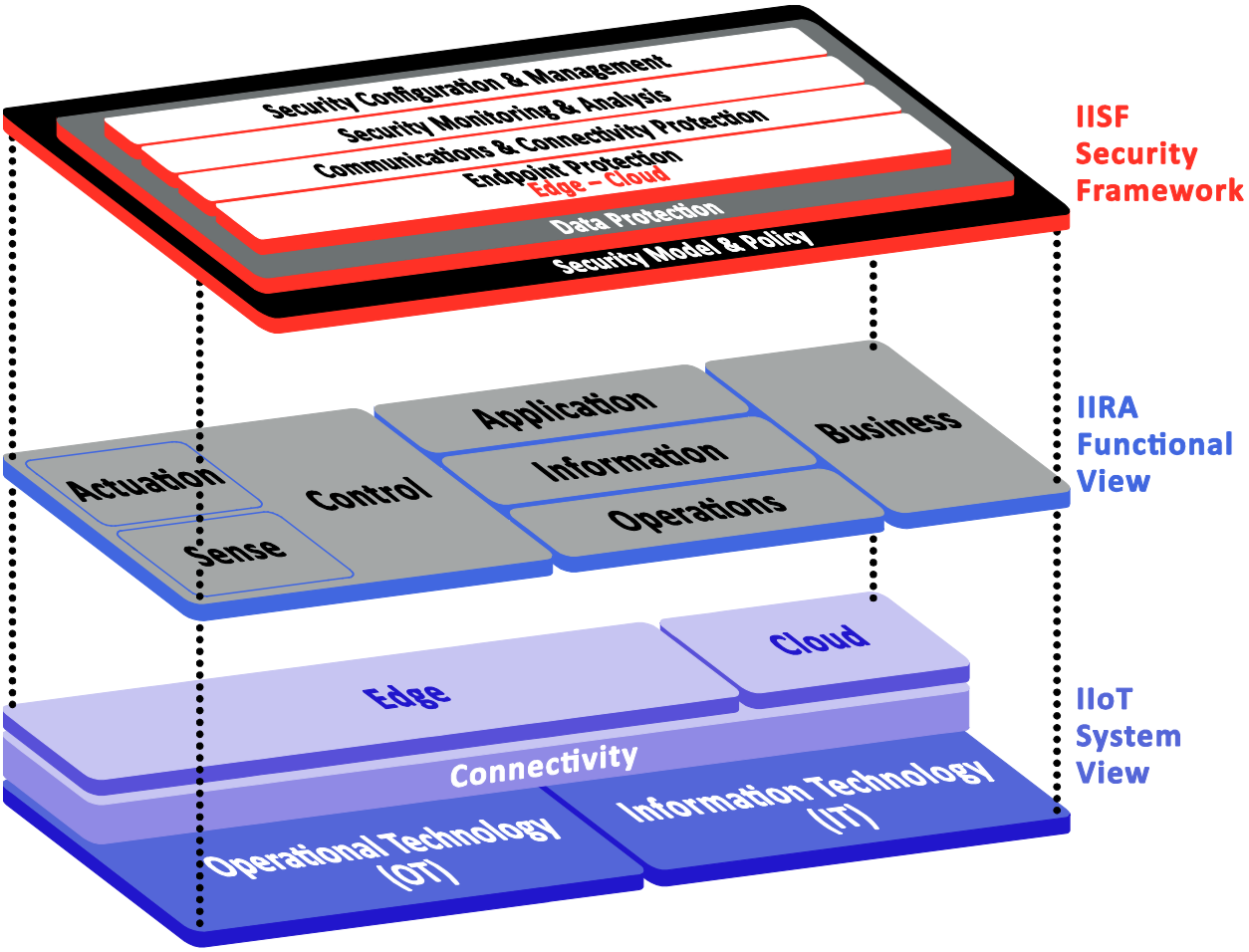

Figure 7-2: Alignment of IISF, IIRA Functional and IIoT System Views

7.3 ENDPOINT PROTECTION

Endpoints are any element of an IIoT system that has both computation and communications capabilities and exposes functional capabilities. These may be edge devices, communications infrastructure, cloud servers or anything in between. Each endpoint has different requirements and hardware constraints that affect the level of protection that can be achieved. Security mechanisms and techniques should be applied to the endpoints depending on their specific function and security requirements.

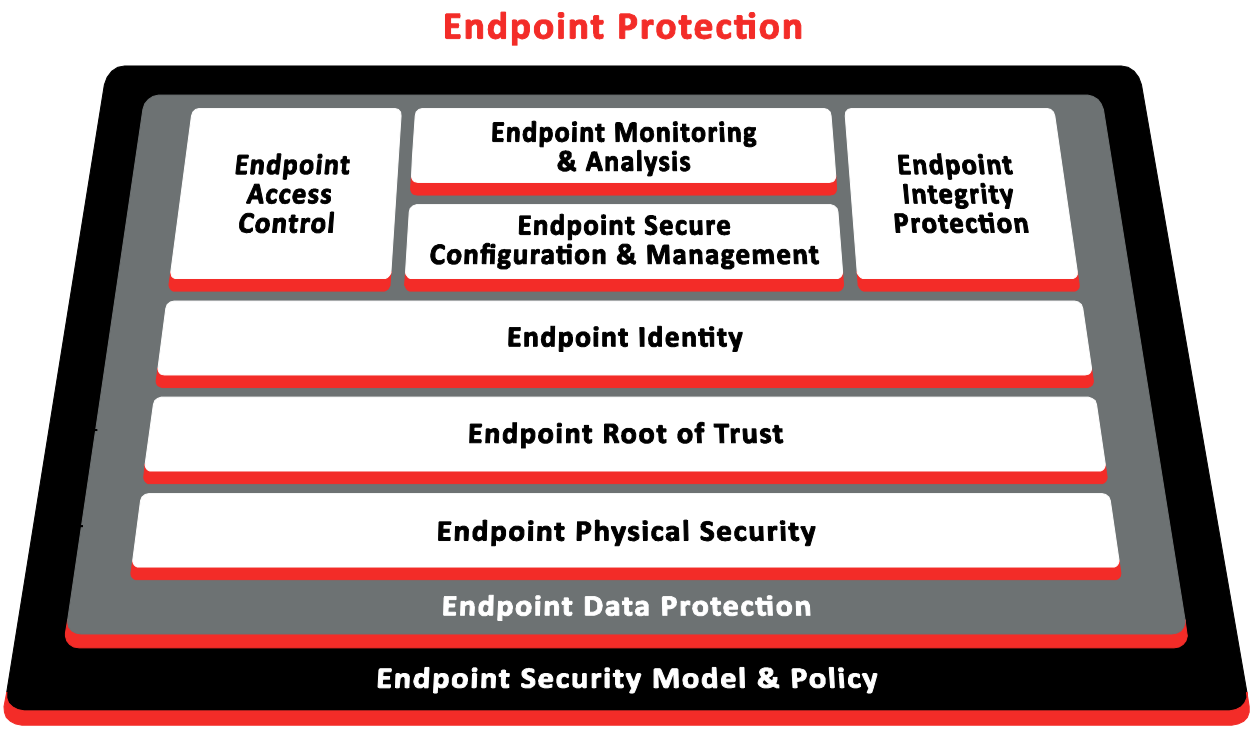

Figure 7-3: Functional Breakdown for Endpoint Protection

Endpoint Protection assures the availability, confidentiality and integrity of the functionality performed by the endpoint.

Endpoint security should consider at least these security functions:

Endpoint Physical Security provides physical protection of the endpoint with anti-tampering and theft prevention mechanisms to prevent uncontrolled changes or removal of the endpoint.

Endpoint Root of Trust provides a foundation to secure other functions at the endpoint, from the hardware to applications including firmware, virtualization layer, operating system, execution environment and application. It also provides confidence on the endpoint identity.

Endpoint Identity is based on the inherent properties of an endpoint that distinguishes it from other endpoints. Identity needs to be supported with evidence or testimonials that confirm the claim of identity, referred as credentials.

Endpoint Integrity Protection ensures the endpoint is in the configuration required to perform its functions predictably.

Endpoint Access Control ensures that proper identification, authentication and authorization is performed prior to granting any resources or services.

Endpoint Secure Configuration and management controls updates of security policy and configuration at the endpoint, including upgrades and patches of known vulnerabilities.

Endpoint Monitoring and Analysis includes integrity checking, detecting malicious usage patterns, denial of service activities, enforcement of security policies and analytics that track security performance indicators.

Endpoint Data Protection provides controls to preserve the integrity, confidentiality and availability of its data.

Endpoint Security Model and Policy governs the implementation of security functions on the endpoint.

Endpoint protection relies on Endpoint Physical Security and establishing the Endpoint Root of Trust. The root of trust determines the confidence in the system and its identity, and ensures integrity and access control to its resources. Once established, the endpoint state must be maintained and tracked in accordance with the system model and policy.

Endpoint Monitoring & Analysis is responsible for ensuring the prevention, detection and recovery from any activity deviant from policy, while Endpoint Configuration & Management ensures that all changes made to the endpoints are performed in a controlled and managed manner.

Endpoint Data Protection is responsible for protecting access and preventing tampering with data-at-rest and data-in-use on the endpoint through encryption, isolation and access control. Data protection spans all data on the endpoint, including configuration, monitoring, and operational data.

The overall security of endpoint is defined in the security policy and enforced through the security model for all controls.

7.4 COMMUNICATIONS AND CONNECTIVITY PROTECTION

Protection of communications and connectivity provides physical security of the endpoint connectivity to the network, protecting Information Flow in the Network, and Cryptographic Protection of communications between endpoints. These two functions, in the diagram below, are supported by areas that traverse the four building blocks at the top layer: Network Configuration & Management, Network Monitoring & Analysis, Communicating Endpoint Protection, and Physical Security of Connections.

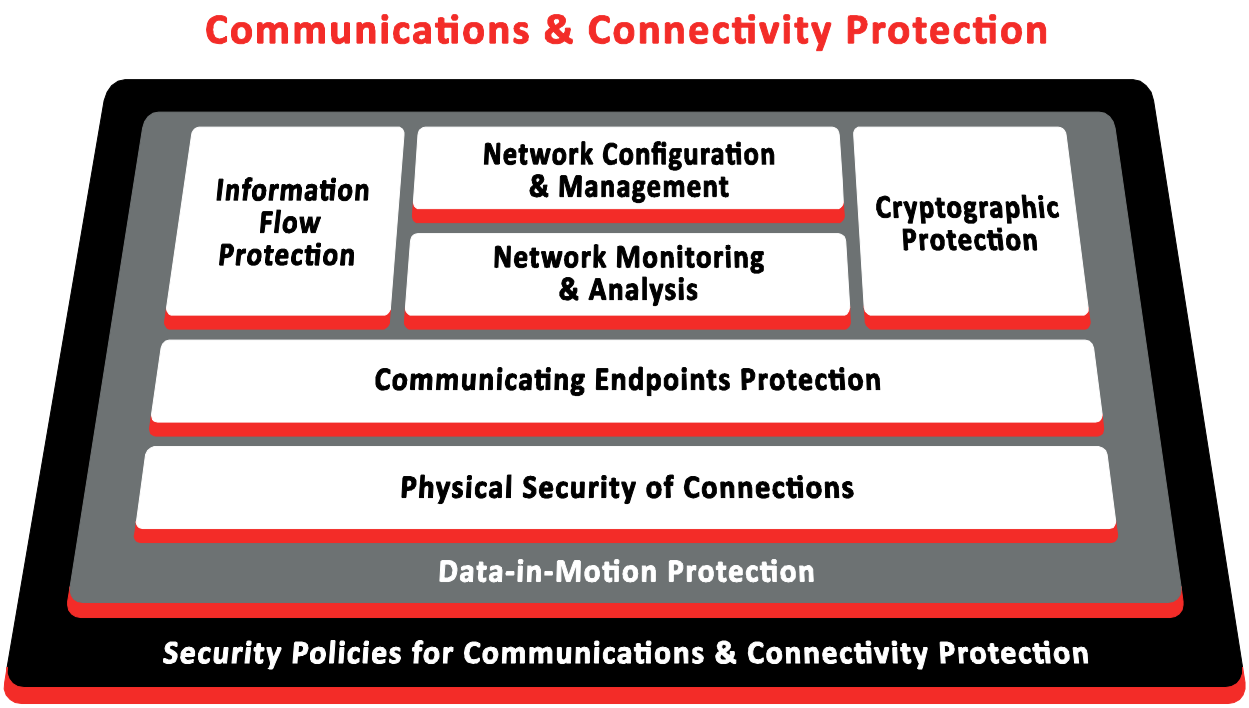

Figure 7-4: Functional Breakdown for Communications and Connectivity Protection

Communication and connectivity security should consider the following functions for protecting and controlling data-in-motion.

Physical Security of Connections ensures that the physical connectivity layer (cables, radios) to the network is protected.

Communicating Endpoints Protection provides some of the functional security building blocks, such as cryptographic keys, to secure communication between endpoints.

Cryptographic Protection uses cryptographic technologies to protect authenticity of communicating parties and integrity and confidentiality of exchanged data and metadata.

Information Flow Protection ensures that only permitted kinds of messages and content reach sensitive systems and networks by isolating network flows using network segmentation and perimeter protection technologies.

Network Configuration and Management controls updates to all network elements and provides enforcement of security policy and configuration for the communications, including network segmentation, cryptographically protected communications settings, and configuration of gateways and firewalls.

Network Monitoring and Analysis collects network data for analysis and includes intrusion detection, network access control, deep packet inspection and network log analysis.

Data-in-Motion Protection provides controls to preserve the integrity, confidentiality and availability of its data.

Security Policies for Communications and Connectivity Protection govern the implementation of security functions on the communications.

The functions listed above are interdependent and interact with each other to deliver security capabilities. For example, to establish secure communication, the communicating endpoints themselves must be secure and apply Cryptographic Protection.

The policy for Data-In-Motion Protection across all of the functions ensures the confidentiality, integrity, and availability of all data travelling between two endpoints. In addition, the Security Policies for Communications and Connectivity Protection defines how elements in the network are allowed to communicate with each other. Both policies must be comprehensive, consistent with each other and account for other key system characteristics including safety, privacy, reliability and resilience to protect availability, integrity and confidentiality of communications.

7.5 Security Monitoring and Analysis

Security monitoring and analysis is responsible for capturing data on the overall state of the system from the endpoints and connectivity traffic then analyzing it to detect possible security violations or potential system threats. Once detected, a broad range of actions derived for the system security policy should be executed. This Monitor-Analyze-Act cycle may complete in real-time or execute later to identify usage patterns and detect potential attack scenarios.

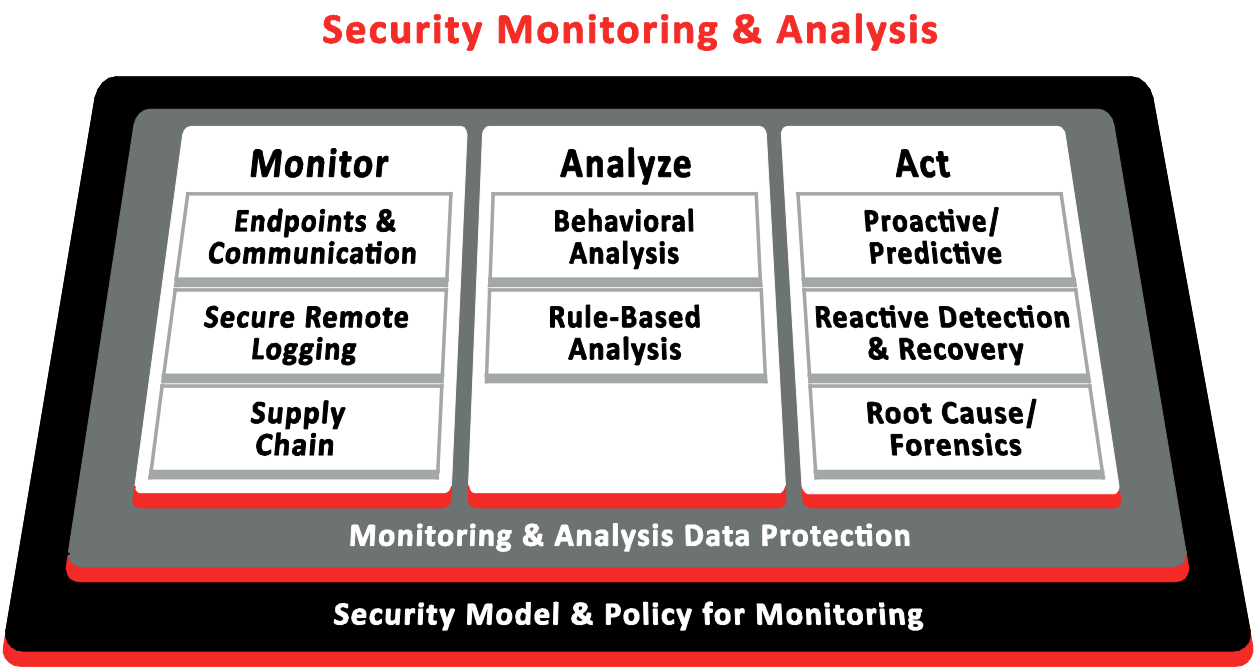

Figure 7-5: Functional Breakdown for Security Monitoring and Analysis

Security monitoring and analysis falls into three top-level functions:

Monitor. As determined by the security model and policy, monitoring captures and aggregates data from each of the sources in the system:

- *Endpoints & Communications*: Monitoring data is gathered by a local agent running on each of the endpoints and communications in the system obtaining information on the implementation of security controls in accordance with the system security policy.
- *Secure Remote Logging*: The sending and receiving of log messages using secure communications.
- *Supply Chain*: Collecting data from all components builders and integrators in the supply chain to assure that security requirements are met.

Analyze. Analysis uses looks for events (for example, violation of security thresholds) and trends that may uncover certain system security vulnerabilities or threats. This phase stores and saves the information for audit or other mining purposes. There are two types of analysis:

- *Behavioral Analysis* observes the usage patterns in the system and learns what is appropriate behavior for the system.
- *Rule-Based Analysis* monitors for violations of predefined policy rules that define events that should never occur in the system.

Act. Having analyzed events and trends, action must be taken. There are three types:

- *Proactive/Predictive* attempts to mitigate threats before the attack begins by observing leading indicators of an imminent attack.
- *Reactive detection & Recovery* provides manual and automated responses to attacks in progress and tries to mitigate them to recover and return to normal runtime state.
- *Root Cause/Forensics* analysis and forensics investigates the underlying vulnerabilities and exploits after the attack.

Monitoring is supported by the other functions in this layer. Monitoring requires protection for the collecting agents at the endpoint, and that the communication between the monitoring and analysis agent, if required, is also protected. Monitoring encrypted channels may not be possible, so monitoring of data-in-motion requires coordination with the policy defining the level of protection of communication between endpoints.

The data collected is protected according to the monitoring and analysis data policy. This policy may be more restrictive than policies for other data types, as it contains aggregated and sensitive information about the system. The security model and policy determines the data captured describing the overall state of the system that is input to the analysis phase.

7.6 Security Configuration And Management

Security Configuration & Management is responsible for the control of changes to both the operational functionality of the system (including reliability and safety behavior) and the security controls ensuring its protection. For example, security configuration and management provides stability to the system by ensuring that all changes to the system are performed in a secure, controlled and trusted way.

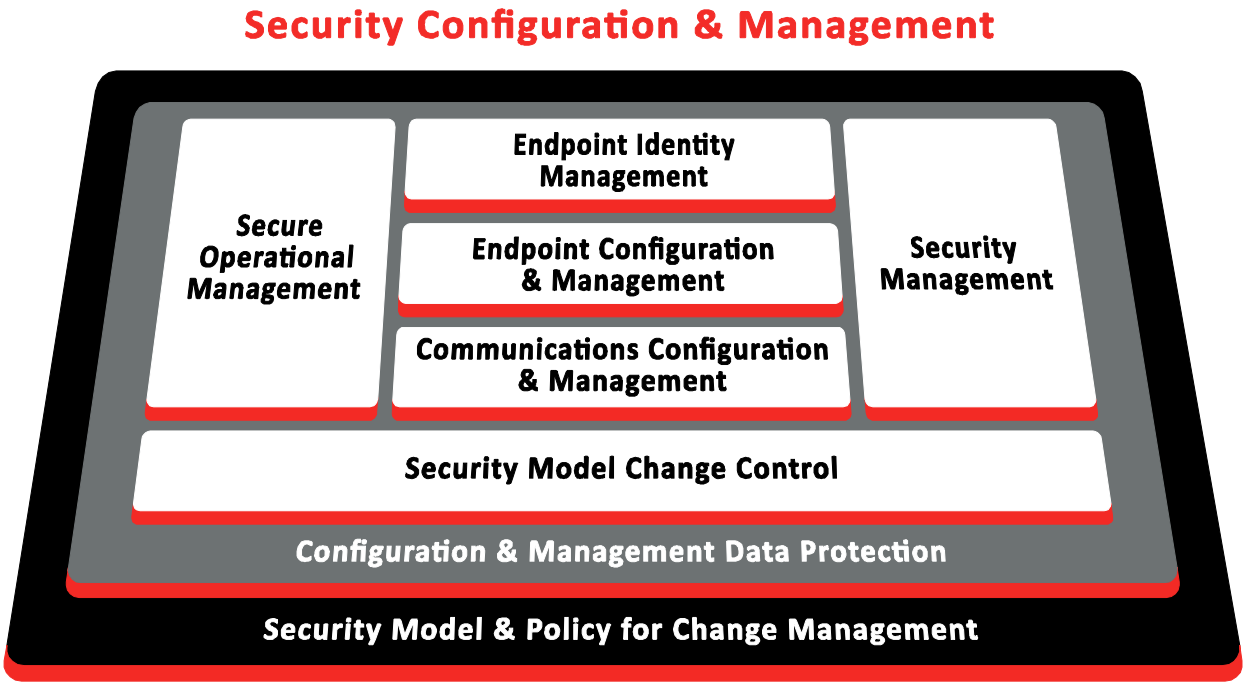

Figure 7-6: Functional Breakdown for Security Configuration and Management

Security configuration management includes following functions:

- *Secure Operational Management* is responsible for managing the secure and controlled changes to all aspects of the operational system, except for security controls for which it is performed separately by Security Management.
- *Security Management* is responsible for ensuring and executing the secure and controlled changes to the security policy and functions throughout the system. It should remain separate from Secure Operational Management.
- *Endpoint Identity Management* generates, updates and revokes machine (and user) principals and cryptographic materials (keys, certificates, etc.) used in the identification of the endpoint.
- *Endpoint Configuration & Management* is responsible for configuring and managing secure and controlled changes to the endpoint including both endpoint operational and security function. This function may be performed by a local agent on the endpoint or through a shared trusted central facility.
- *Communications Configuration & Management* configures and manages the security controls specifically for communications and the network.
- *Security Model Change Control* is the process by which changes to the security model and security policy are managed in the configuration and management process.
- *Configuration & Management Data Protection* is the function that is responsible for protecting all data (at rest, in use and in motion) related to the configuration & management of the system.

- *Security Model & Policy for Change Management* is the process that governs security configuration management functions.

To change the configuration on security controls, the security model should be transformed into actionable settings in the security policy, including the identification and configuration for the endpoints and their connectivity. The level of granularity for configuration and management of the system varies depending on the systems and trust requirements capture in system security model and policy.

7.7 DATA PROTECTION

Data is pervasive throughout the IIoT system. Each set of data has a different lifecycle, time of relevancy and potential risk associated with its compromise. The threat may result from its modification, interception or duplication. The effects of attacks on data vary from immediate change in system behavior to more subtle negative behavior in the future.

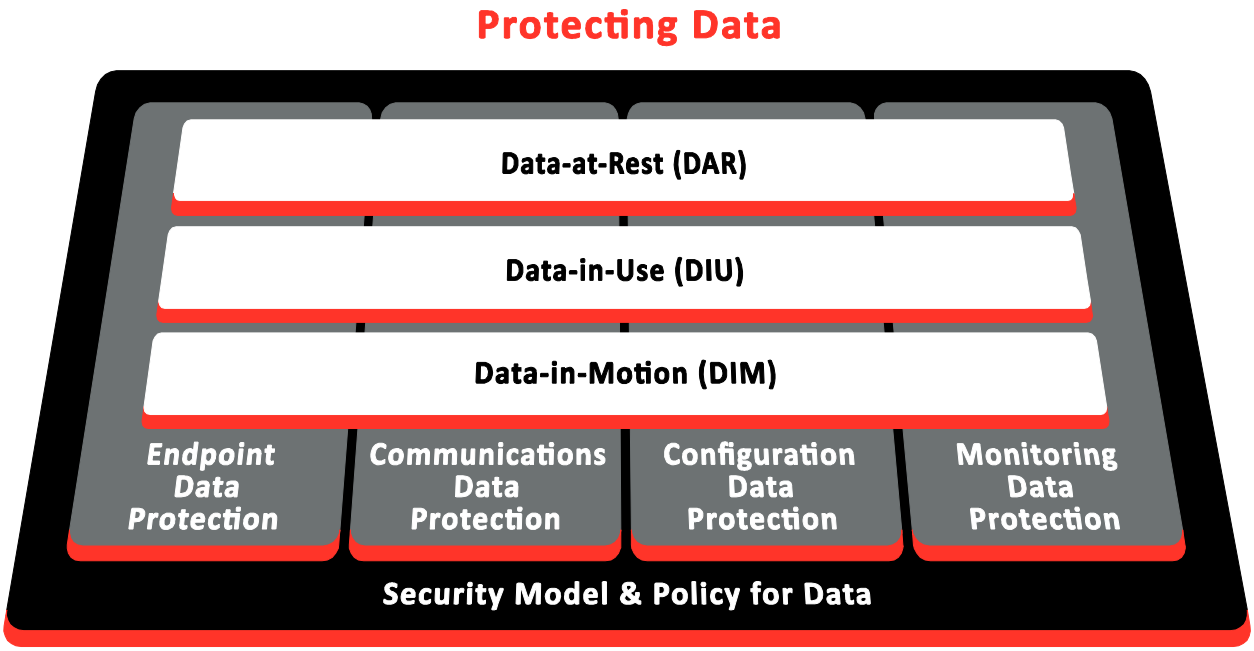

Figure 7-7: Functional Breakdown for Data Protection

Figure 7-7 shows a functional breakdown of the building blocks for data protection in the security framework. Different types of data to protect include:

- *Endpoint Data Protection* refers to operational and security related data used, stored or moved through the endpoint.
- *Communications Data Protection* addresses all data pertaining to the operations of the network communications and connectivity and the data transmitted across these connections between endpoints.
- *Configuration Data Protection* is all the data relating to the security or operational configuration of the system including all endpoints and connections.

- *Monitoring Data Protection* refers to all data generated by the system in response to tracking current state and changes of key system parameters, indicators and activities relevant in ensuring system trustworthiness.

The data protection strategies for each type of data fall into three categories:

- *Data-at-Rest* (DAR) is data in persistent storage, for example, on a long-term network-attached cloud storage drive, on a local USB drive, or in a *solid state disk* (SSD) on an edge device.
- *Data-in-Use* (DIU) is data placed in non-persistent storage such as *random access memory* (RAM) and CPU caches and registers.
- *Data-in-Motion* (DIM) is data moving between two or multiple connected endpoints.

Data, whether in-motion, in-use, or at-rest, must be protected against unauthorized access and uncontrolled changes by applying functions such as confidentiality controls, integrity controls, access control, isolation and replication. The level of protection should be commensurate with the impact of data loss or falsification, and the retention period should be defined.

7.8 SECURITY MODEL AND POLICY

The *Security Model & Policy* covers regulatory, organizational and machine levels of security (see Figure 7-8). The *Security Policy* describes the security objectives of the system, and the *Security Model* is a formal representation of security policies enforced in the system. Various security models may be applicable in a system, and the scope of these models may address different security functions or security domains within it. Security Model & Policy encompasses all security aspects of the system including how to protect endpoints, communications and data. It also defines what is to be monitored, analyzed and recovered and who and how changes may be made to all aspects of the system.

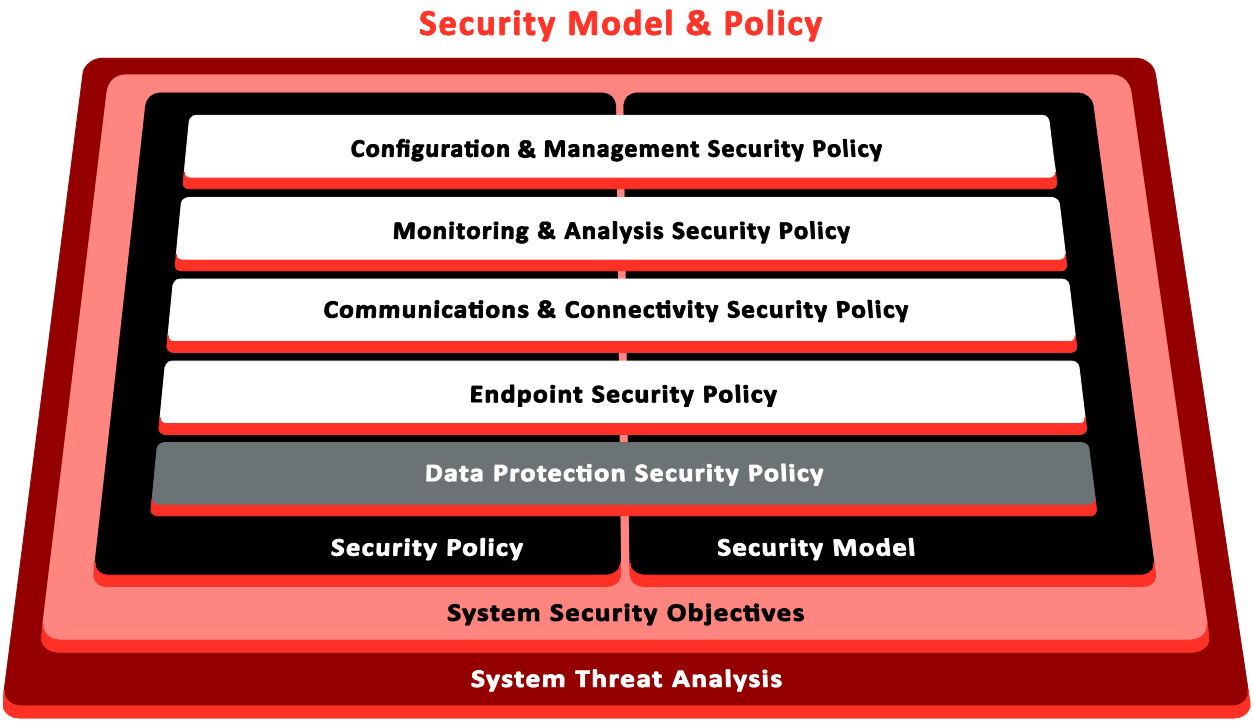

Figure 7-8: Functional Breakdown for Security Model and Policy

The key functions provided in security model and policy are as follows:

- *System Threat Analysis* function is responsible for performing the iterative and continuous process of identifying the threat capabilities, the threat's possible access to the system and assessing the systems vulnerability to attack.
- *System Security Objectives* building block is responsible for setting the security objectives of the system in terms of the confidentiality, integrity, availability and accountability requirements for the system. These objectives serve to guide in the creation of the specific security policy for the IIoT system.
- *Security Policy* for the system is the living document that defines the processes, rules, security measures and controls to be enforced throughout the IIoT system.
- *Security Model* is the function that provides formal representation for specifying and enforcing the security policies for the IIoT system.
- *Data Protection Security Policy* is the building block responsible for defining security policies pertaining to protecting the availability, integrity and confidentiality of data in all forms in the IIoT system.
- *Endpoint Security Policy* is the function responsible for defining and communicating the security policy for all endpoints in the IIoT system and ensuring that it is executed in a secure and controlled fashion through the respective Endpoint Secure Configuration & Management function.
- *Communications & Connectivity Security Policy* is responsible for defining and communicating the security policy for all communications and connectivity in the IIoT

system and ensuring that it is enforced system-wide via the Network Configuration & Management function.

- *Monitoring & Analysis Security Policy* is the function block responsible for defining and communication the security policy for all monitoring and analysis activities in the IIoT system and making sure that it is enforced system-wide via the Monitoring & Analysis function.
- *Configuration & Management Security Policy* is responsible for setting and communicating the security policy for the processes and controls associated with configuration change and management in the IIoT system. The Security Management capability is responsible for ensuring that this policy gets communicated to all the endpoints and communications capabilities of the system.

The Security Policy includes policies for the system and sub-policies for the endpoint protection, communications and connectivity protection, security monitoring and analysis, security configuration and management and data protection (see individual sections 7.3 to 7.7). The system threat analysis enables the creation of the security objectives for the system, derived from regulations and standards. From these objectives, the applicable security policies are selected based on the industry vertical, customer base, geographic location and other considerations. The security policy describes the overall business-risk considerations and defines the guidelines for securing the day-to-day proper functioning of the system. This policy is then transformed into a security model, and determines and drives requirements to the functionality of the building blocks of the security framework. For example, each machine-level security policy specifically covers the security policies associated with the endpoint and the devices it may be connected to or in control of.

7.9 FROM FUNCTIONAL TO IMPLEMENTATION VIEWPOINT

The functional viewpoint presented the six key building blocks for IIoT security. These functions serve as guidance for implementing security end-to-end across IIoT systems in the context of trustworthiness. A set of security design principles should guide the capabilities and techniques employed in the implementation viewpoint of a specific implementation.

As per Saltzer and Schroeder,[1] implementers should consider eight design principles prior to implementation of security capabilities for their IIoT system:

- *Principle of economy of mechanism*: keep the design as simple and small as possible.
- *Principle of fail-safe defaults*: base access decisions on permission rather than exclusion.
- *Principle of complete mediation*: every access to every object must be checked for authority.

[1] See [Saltzer1974]

- *Principle of open design*: a design should not be secret. The mechanisms should not depend on the ignorance of potential attackers, but rather on the possession of specific, more easily protected, keys or passwords.
- *Principle of separation of privilege*: where feasible, a protection mechanism that requires two keys to unlock it is more robust and flexible than one that allows access to the presenter of only a single key.
- *Principle of least privilege*: every program and every user of the system should operate using the least set of privileges necessary to complete the job.
- *Principle of least common mechanism*: minimize the amount of mechanism common to more than one user and depended on by all users.
- *Principle of psychological acceptability*: it is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly.

A broad number of capabilities and techniques may be applied to implementing each of the functional building blocks. Annex C provides an overview of these mechanisms and their respective applicability to each of the functional building blocks.

8 PROTECTING ENDPOINTS

Endpoints cover the entire spectrum of IIoT edge devices including simple sensors, Programmable Logic Controllers (PLC) and massive cloud servers with significant computing capabilities. An endpoint may be part of a control network, a concentrator between multiple communications streams, or routing traffic between other endpoints inside of the cloud infrastructure. The endpoints may be on dedicated hardware or shared or virtualized hardware. Endpoint security should consider at least these security functions as described in section 7.3.

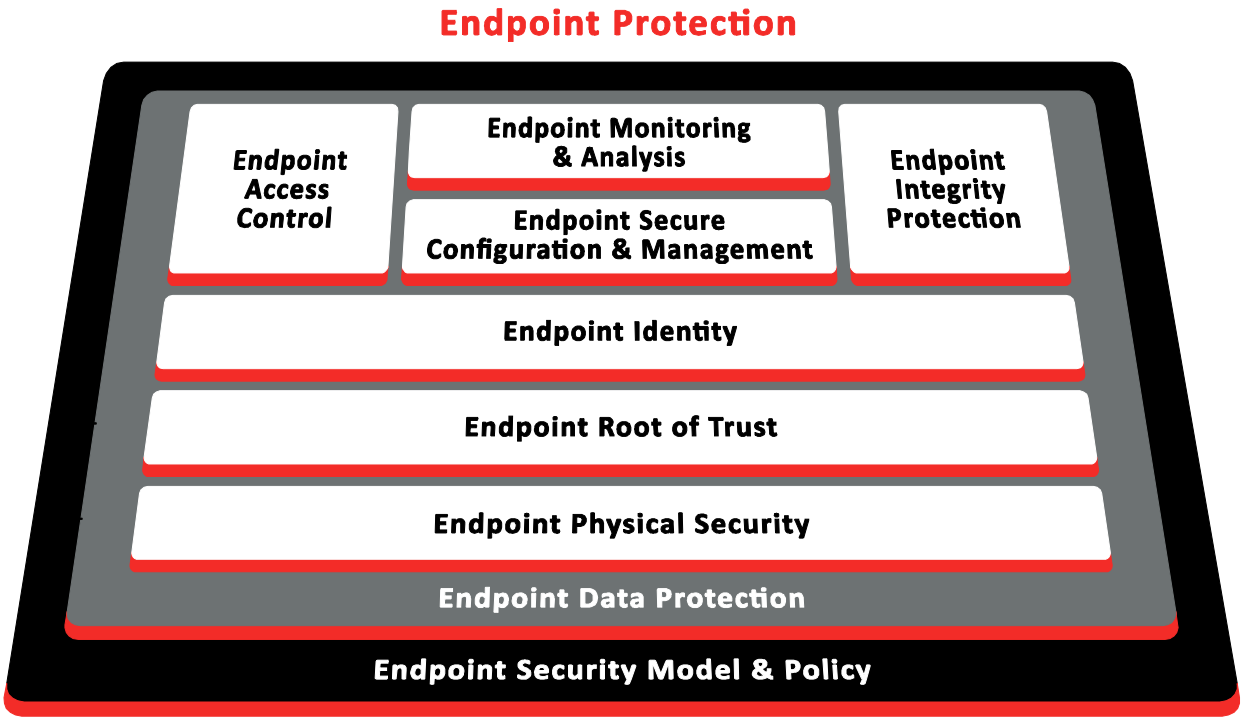

Figure 8-1: Functional Breakdown for Endpoint Protection

The checkmarks in Table 8-1 show the implementations for security functions that mitigate the vulnerabilities and threats to the endpoint.

Security Objectives			Functions and Techniques	
Availability	Integrity	Confidentiality	Description	Section
✓	✓	✓	Endpoint Physical Security	8.3
	✓		Establish Roots of Trust	8.4
✓	✓	✓	Endpoint Identity	8.5
✓	✓	✓	Endpoint Access Control	8.6
	✓		Endpoint Integrity Protection	8.7
✓	✓	✓	Data Protection	8.8
✓	✓	✓	Endpoint Monitoring & Analysis	8.9
✓	✓	✓	Endpoint Configuration & Management	8.10
	✓	✓	Cryptography Techniques for Endpoints	8.11
✓	✓	✓	Isolation Techniques for Endpoints	8.12

Table 8-1: Endpoint Objectives, Functions and Techniques (Chapter 8 Outline)

Along with the building blocks for endpoints, two techniques that apply to all building blocks, isolation and cryptography, are described in this chapter.

Cryptography Techniques is a discipline that embodies principles, means, and mechanisms for the transformation of data to hide its information content, prevent its undetected modification and prevent its unauthorized use.

Concealment of resources sometimes uses *Isolation Techniques* (see section 8.12) to provide visibility only to those that have authorization.

8.1 Security Threats and Vulnerabilities on Endpoints

Endpoints have many potential vulnerabilities susceptible to malicious or unintentional errors. Figure 8-2 shows a broad range of solutions stacks ranging from a bare metal application (left side) to a guest OS running in a virtual machine on a hypervisor (right side) that isolates applications in their respective containers. Each configuration has strengths and weaknesses that must be evaluated for each application. For example, bare metal applications generally have fewer security controls implemented, but run on more resource-constrained hardware. On the other hand, a hypervisor-based security solution requires more processing power, but can dedicate an entire virtualized instance to security.

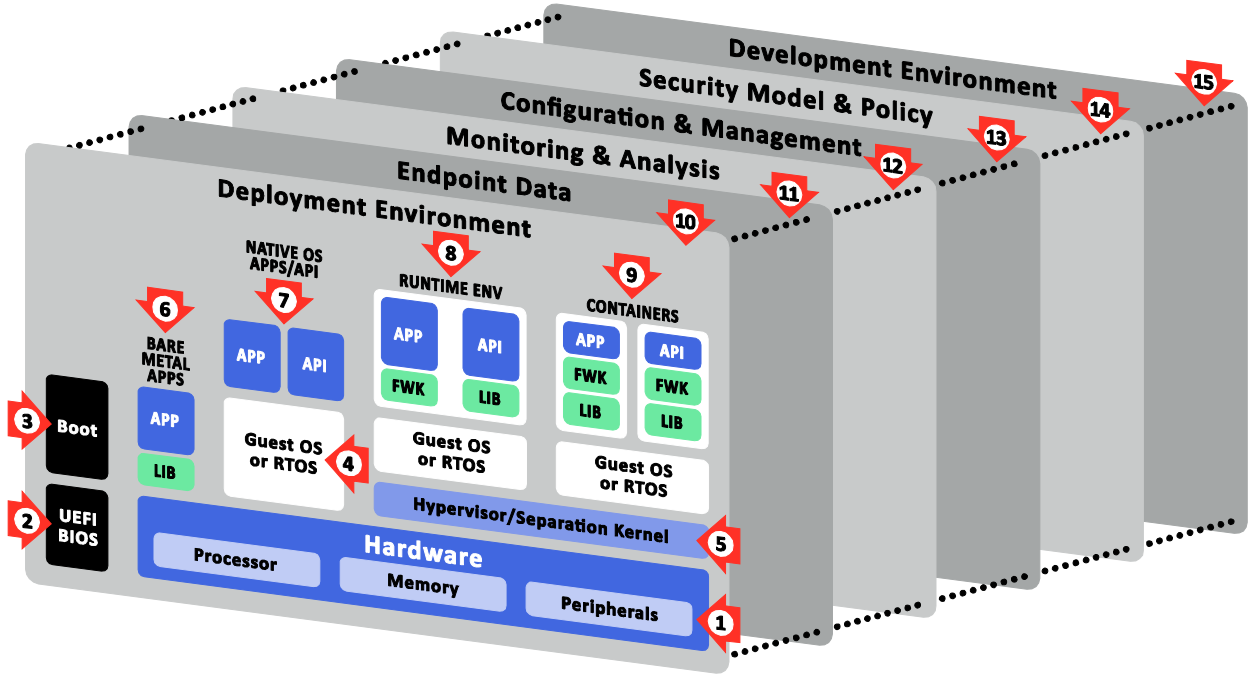

Figure 8-2: Threat and Vulnerabilities to IIoT Endpoints

As shown in Figure 8.2, a broad range of threat and vulnerabilities exist in different facets of the endpoints in each of the following areas:

- Changes in hardware components and configuration, ①: Hardware integrity must be assured throughout the endpoint lifecycle to deter uncontrolled changes to the hardware components. A potential vulnerability of the hardware is the usurpation of some part of the hardware resources. The endpoint must be able to protect itself against unauthorized access and the monopolizing of key resources such as memory, processing cycles and privileged processing modes.
- Intercepts or overrides of the system boot process, ②+③: The endpoint boot process can be altered by modifying the firmware interface between the hardware platform firmware and the operating system such as the *unified extensible firmware interface* (UEFI) or *basic Input/output system* (BIOS)[1]. Changes to the bootloader are another threat as changes could compromise the integrity of the endpoint by starting unauthorized or insecure versions of the operating system. Attacks at this level could also affect the normal or secure boot process of the endpoint, the recognition of all the hardware resources and the establishment of a solid root of trust for securing other components.
- Compromises to the Guest OS, Hypervisors and Separation Kernels, ④+⑤: These software layers control allocation of hardware resources to applications. Attacks to these layers can alter the behavior of the system, allow information flows to bypass security controls and enable attackers to gain privileged access to endpoint hardware and software resources. Once access is gained to this layer, attackers will have opportunity to affect the entire software stack and further alter security controls built in to this level.
- Illicit changes to Application Software or exposed Application Programming Interface (API), ⑥+⑦+⑧+⑨: Endpoint applications are often the target for malware or an attacker seeking to infiltrate and compromise the endpoint. Execution of malicious applications or overriding of application APIs can adversely impact the trustworthiness of the endpoint. Exposed APIs should also be protected against denial of service attack where continuous access from unauthorized users could limit the responsiveness and access to the exposed functionality.
- Vulnerabilities of the Deployment Process, ⑩: Errors and potential malicious code may also infiltrate the endpoint as part of the deployment process, for example, incorrect or malicious installation scripts, intercepted communications, or unauthorized replacement of a package on the update server. Reduction of possible endpoint configurations in large-scale endpoint deployments will be important in reducing complexity and vulnerabilities in the deployment process.
- Unwanted changes to Endpoint Data, ⑪: Data throughout the endpoint from low-level firmware all the way up the software stack represents a key area of vulnerability. These vulnerabilities include unauthorized access to mission-critical or private data. Attackers may adversely affect the behavior of the system by injecting false data. Denial-of-service

[1] see [UEFI] and [BIOS]

attacks on data access may impede timely and accurate execution of the endpoint functionality resulting in costly outcomes.

- Breach of the Monitoring & Analysis system, ⑫: An attacker could gain visibility on the functions of the monitored system. For example, an attacker could modify monitoring data to make it appear as if a particular event did not occur. Modification of the security logs and monitoring data may result in undetected vulnerabilities or compromised states. As a result, attackers would benefit from a coverage gap, compromising endpoint hardware and software or destroying evidence of their activities after an attack.
- Vulnerabilities in Configuration & Management, ⑬: Vulnerability of the Configuration & Management system may result from improper access control to the configuration management system, insertion of unauthorized changes in the system or corruption of update payloads. Updates to the endpoints should be planned and managed so as to limit the number of different operational configurations and reduce fragmentation of the fleet.
- Uncontrolled changes to Security Policy and Model, ⑭: Modification of the security policy and derived security models represent a serious threat to the system and its endpoints. Equally, weakness in the security policy is an area for exploitation by potential attackers.
- Vulnerabilities in the Development Environment, ⑮: The introduction of weaknesses during the software development lifecycle can leave the IIoT systems susceptible to attack. These weaknesses may be introduced during architecting, designing, or writing of the code. Use of vulnerable or malicious libraries or untrusted development frameworks may lead to their inclusion in the resulting code running in the IIoT system.

After consideration of all the above-mentioned potential threats to the endpoint, a sound and thorough assurance process is required to ensure that the resulting system is trustworthy. Gaining assurance about the software integrity of the endpoint includes gathering evidence across all of the development and operational lifecycle. This effort should determine whether potential weaknesses, like those in 'Common Weakness Enumeration' (CWE)[1] have been avoided, removed or remediated, and then tagging that baseline and using it to verify that correct software is loaded at boot. The ISO/IEC 19770[2] specification on Software Tagging may be useful for tagging software at the source. This provides assurance that those packages come from authenticated and authorized sources.

8.2 Architectural Considerations for Protecting Endpoints

Implementing security on endpoints depends upon their computational and communication capabilities. On the edge, endpoints may be resource-constrained devices with less computing power and with static configurations. In the cloud, endpoints may be servers with extraordinary computational capability and dynamic configurations.

[1] See [CWE]
[2] See [ISO-19770]

Endpoint security architecture should be modular, scalable and non-intrusive to the OT processes. Common building blocks and consistent interfaces across different endpoints ease integration and enhance end-to-end security. Consistent API-level capabilities across all endpoints (e.g. the edge, the communications and the cloud) promote a clear integration framework. Security isolation techniques separate capability and services while limiting their exposure and possible threat vectors.

Many deployments are spread across numerous legal entities where data ownership rights and implementation choices may lead to liability concerns. These concerns may lead to integration inconsistencies that can complicate even the most straightforward architectural choices.

8.2.1 ENDPOINT SECURITY LIFECYCLE

An IIoT security model begins with the security capabilities of the endpoints, as implemented by the vendors. Their choices have long-lasting effects on the security potential of the endpoint. Hardware is difficult to change after manufacture, and software too depends directly upon the vendors' willingness and ability to test for security flaws properly.

Once the vendors have delivered the endpoint to the market, the system integrator inherits the burden of integrating the products securely. Ideally, the system integrator designs a framework for end-to-end security across the system. In practice, inconsistencies between the vendors' security controls and the quality of their implementation often require additional effort.

A security maturity model[1] enables evaluation based on implementation mechanisms and architectural design, development and maintenance processes. Both the system integrator and the owner/operator can evaluate the maturity of the security posture holistically, rather than depending on penetration testing after the security has been implemented. This allows for future security needs to be shared with manufacturers, and both a roadmap of security capabilities, as well as the results of the periodic (hopefully frequent) testing of the security, to define clearly how the security improves over time to react to the threat environment.

8.2.2 HARDWARE VERSUS SOFTWARE

Implementing security in hardware as opposed to software offers some specific advantages and disadvantages that must be considered for IIoT. Specialized tamper-resistant hardware provides a greater level of trust, particularly for cryptographic keys and operations. However, this comes at a cost, either monetary or in terms of management and update complexity. Software security has been dominant in IT and enterprise settings, but those solutions may not translate well to OT-based environments. Software security solutions generally have a lesser level of trust, but have better infrastructure for management and updates.

Battery life is a concern for many resource-constrained devices. In most cases, hardware-assisted security dramatically extends the useful battery life compared to software.

[1] See [ENER-C2M2]

Often, hardware implementations are not upgradable, so the performance and battery life increases may come at the cost of a rigid and static implementation of the security functionality. If a vulnerability in the algorithm is found, it is more difficult to make the needed changes to the device. Architects must weigh these conflicting requirements when determining the balance between hardware- and software-enabled solutions. Field-programmable gate array (FPGA) chips provide both accelerated hardware benefits as well as reprogramability.

Hardware security modules (HSM) offer hardened and isolated hardware components for security operations. Common functions include strong tamper resistance, cryptographic key storage and lifecycle management, such as key generation and strong authentication. An HSM may also be leveraged for providing security during the upgrade process. Other applications of an HSM include secured remote communication establishment to a remote device and execution of firmware image flashing using cryptographic keys.

A common implementation of an HSM is the Trusted Platform Module (TPM)[1]. The TPM is sometimes difficult to qualify because it is simultaneously a standard, an implementation, and in some cases a discrete hardware chip on the endpoint. The standard describes a hardware container that performs crypto operations separate from the CPU. This container is generally used for key generation, key storage, signing and sealing of data and similar operations. The implementation takes place in a separate discrete hardware chip, or in a dedicated hardware container that may be co-located on the same physical die as the CPU, but in an isolated region.

Often working in conjunction with an HSM is another element, which may be hardware- or software-based: the Trusted Execution Environment (TEE). The TEE is an isolated area on the device platform providing security functionality for integrity and confidentiality. The TEE offers a higher level of security by separating the security functionality from the operational functionality on the main CPU. Common security functions include isolated execution of security operations, integrity of code loaded and data stored, and confidentiality for data stored in the TEE. It protects data-at-rest and data-in-use within the TEE. A software-based TEE could be a virtual gateway running on a hypervisor, isolating the security functionality from the operational applications running in a separate virtual instance. Other examples of software TEE include Docker containers and Trusty TEE for Android OS[2]. Examples of hardware TEEs include the GlobalPlatform TEE, Intel Converged Security and Manageability Engine (CSME), and ARM TrustZone[3]. There are also hybrid hardware-backed software-defined TEE implementations such as Intel Software Guard Extensions (SGX).[4]

8.2.3 BROWNFIELD ENDPOINT CONSIDERATIONS

In brownfield deployments, the endpoints are deployed for long periods of time, sometimes for decades, but they should be upgraded to safe levels. The primary consideration is not to disrupt

[1] See [TCG-TPM]
[2] See [Docker] and [Andr-Trusty]
[3] See [GloP-TEE], [Intel-AMT], [Ruan2014] and [ARM-TrustZ]
[4] See [Intel-SGX]

the existing business process with added security controls or false-positive security events. Security controls should be loosely coupled to the industrial processes to minimize the interdependencies between them.

The most common technique for implementing security quickly and effectively is to deploy a security gateway that provides security capabilities to the devices behind it. Common functionality includes:

Storing and managing identity on the gateway isolates the identities so they can be maintained for each device behind the gateway. This may limit the number of devices that a single gateway may manage.

Mutual authentication on behalf of devices behind the gateway with devices in front of the gateway makes it appear that the brownfield device is capable of maintaining identity and performing mutual authentication, even though the gateway is performing these tasks.

Authorizing network traffic to filter traffic down to only those flows that are explicitly allowed between the two devices. This is a network whitelist of allowed communications; all others should be logged and potentially blocked.

Confidentiality and integrity controls can encrypt the data for confidentiality or to sign the data for integrity purposes.

Using a gateway is generally quicker and cheaper to implement than modifying the devices in the environment. Gateways can be deployed relatively quickly to provide a consistent level of security across all of the devices, and to manage the devices uniformly. Gateways can also eliminate vendor-specific management inconsistencies between devices. This makes security independent of the make, model and manufacturer of the device. Gateways provide network-level security, but not the edge-device integrity and security that would provide fine-level control and visibility. Gateways are an initial step to achieve a quick increase in security to a consistent level. Later, device-level security capabilities such as runtime integrity controls can be added.

8.3 ENDPOINT PHYSICAL SECURITY

Endpoints are deployed in a broad range of environments with different security requirements for protecting assets against theft, tampering, vandalism, or adverse effect from environmental conditions. This protection may be integral to the endpoint (e.g. detection of changes to hardware configuration) or provided as part of an enclosure encapsulating the endpoint (e.g. protective rack enclosure for the device).

Physical access techniques are widely used in industrial systems to prevent unauthorized users from physical contact with endpoints and communication devices. Examples include physical perimeter security measures, such as doors and walls where access to unauthorized parties is prevented with access control techniques (locks, biometrics, RFID cards) and monitored by surveillance of the assets to be protected. Standards such as NIST SP 800-53 ‘Physical and

Environmental Protection' (PE)[1] provide information on methods for physical protection, access control and monitoring.

Some endpoints, such as smart meters and environmental sensors, must reside outside physical perimeter security. Physical enclosures may provide tamper evidence that exposes modification events as well as indicating the severity of tampering. Such enclosures can deter unauthorized casual tampering and protect system components from adverse weather conditions and other hazards that may cause unexpected failures. The enclosures should provide stable operating conditions by delivering controlled power source, stable temperature, protection from dust and other environment substances that could adversely affect the endpoints determinism. Physical access to endpoints that provide ports for peripherals, such as USB, should be controlled to prevent unauthorized attachment of peripherals.

Depending on the threat model, the endpoint should implement tamper-resistant hardware components or other secure storage to prevent key extraction. The level of protection from hardware attacks by a device can be accredited using certifications[2]. Endpoints may have physical tamper protection features built-in that are capable of detecting and reporting any change to the physical hardware including its sub-components. Essential endpoint parts may be tagged with unique identification numbers preventing their use outside the configured context. Hardware protection mechanisms should be able to detect the substitution of any component with less capable or malicious replacements.

In highly controlled and regulated environment, the physical security status of the endpoint should be monitored and controlled automatically as part of the endpoint monitoring and configuration management functions. This kind of physical security should be able to detect and report any unauthorized access or modifications to the physical configuration or integration of the hardware. These endpoints could expose an interface allowing higher-level system physical security services to monitor or receive notifications pertaining to the security status of the endpoint easily.

8.4 ESTABLISH ROOTS OF TRUST

The *roots of trust* (RoT), or trust roots, consisting of hardware, software, people and organizational processes, establish confidence in the system. An endpoint without a correctly implemented RoT will lack the ability to establish confidence that it will behave as intended.

The root of trust on a device determines the level of confidence in the authenticity of the credentials belonging to that particular device. The root of trust should be able to generate, manage and store at least one identity.

The strength of the RoT determines the level of trust attainable by the device. The level of security provided by the RoT depends on how it is implemented. The RoT should be simple and well protected against compromise to ensure its integrity. Ideally the RoT should be implemented

[1] See [NIST-800-53]
[2] See [FIPS-140-2]

in hardware, referred to as a *hardware root of trust* (HRoT). A HRoT is a stronger security control than a software- or firmware-based RoT. Hardware technologies such as TPM and HSM, discussed in section 8.2.2, provide efficient platforms for implementing RoT.

For many systems, the RoT is provided by the Unified Extended Firmware Interface (UEFI), which replaces the BIOS. It measures the integrity of the firmware stored in flash memory, ensuring it cannot be modified without authorization (having the proper keys), thus forming the RoT.

There are different types of roots of trust as explained in Trusted Computing Group TPM Specification[1].

The *attestation* process is the issuance of a statement based on a decision that fulfillment of specified requirements has been demonstrated. The trust root must be *attestable*, meaning it has a mechanism to share its integrity and the level of security it is providing to other trusted systems securely.[2]

8.5 ENDPOINT IDENTITY

Endpoint identity is a building block that enables a broad range of security controls that depend on proper handling of identity. For example, identity is the basis for trust in asset management, authentication, authorization, and remote maintenance.

An *entity* is an item with a recognizably distinct existence. For example, a device is an entity. But some devices comprise multiple endpoints, each of which is an entity, and each endpoint comprises multiple components, each of which is also an entity. *Identity* is an inherent property of an entity that distinguishes it from all other entities. An identity must exist in a namespace to allow it to be referred to without ambiguity. A *credential* is evidence that supports a claim of identity. An example of an identity is an entity identifier that is unique within a particular namespace; the credential would be the key.

An endpoint may have a single identity, or multiple identities, used for different applications. Credentials are used to verify the identity of the endpoint. There are several levels of trust that may apply to an endpoint, depending on the threat model of the particular IIoT system. Each level of trust determines the minimum security capabilities of the credentials, including credential uniqueness, credential storage, and credential usage (e.g. for authentication, authorization etc.). Digital certificates, RFID, passwords, biometrics and QR codes are all examples of credentials, but vary greatly in their level of trust.

One common example of a credential is a cryptographic certificate (e.g., X.509 digital certificate), which is a cryptographically signed structure that binds public keys to an identifier for the entity (i.e., a distinguished name). Certificates may be generated and signed by a certificate authority (CA), for better level of trust, but can be self-signed for localized self-assertion of trust requirements.

[1] See [TCG-Spec]
[2] See [TCG-TPM-Spec]

The level of trust attributed to a credential depends on its uniqueness and strength. An IP address, a MAC address and a QR code are all credentials, and they are unique, but they are not strong, as they can be falsified to impersonate another endpoint. A cryptographic certificate is both unique (with appropriate randomness) and strong (depending on key type and length). However, if the private key associated with the certificate is not stored and processed in protected storage and memory, the certificate can still be compromised.

Several standards exist that provide guidance on choosing the right level of protection for endpoint identity: ISO/IEC 29115, IEC 62443 and ISO/IEC 24760-1.[1]

In ISO/IEC 29115, four levels of authentication (LOA) are described in the list below:

- Low: Weak credential with no crypto (IP address, MAC address, etc.), or insecure authentication protocol
- Medium: multi-factor authentication, and secure authentication protocol with secrets being protected (no crypto), and controls to prevent attacks on stored credentials
- High: multi-factor authentication, and cryptographically protected authentication protocol, and any RoT (e.g. software keystore, or OS-enforced access control on a file system)
- Very High: all methods described in High plus the addition of tamper-resistant HRoT (including credential storage and cryptographic operations inside the HRoT), and cryptographically protecting privacy-sensitive data in the authentication protocols.

The descriptions above lists the levels including their mapping to a notion of trust levels from lowest to highest in strength.

In IEC 62443 four security levels (SL1-4) of protection are described for seven foundational requirements (FR), one of which is 'Identification and Authentication Control'. These four levels of security pertain to the security of the system in general as a measure of confidence that the system is free of vulnerabilities. For the 'Identification and Authentication Control' FR, technical security requirements are defined for identifying all entities (human, software processes and device). The security requirements (SR) for the selected required SL enable asset owners to assess the capability required to protect credentials.

If no threat exists against the endpoint, cleartext credential, such as identification numbers may be used. In some rare instances, it may not be required for all endpoints to support identity, but the risks should be well understood and documented. ISO/IEC 24760-1 defines three levels of trust for identities: identity, unique identity and secure identity. Industrie 4.0 provides information[2] on what a secure identity technology consists of, and in the case of digital identity a secure identity is a certificate protected by an HRoT such as a TPM.

[1] See [ISO-29115], [IEC-62443-11], [IEC-62443-21], [IEC-62443-23], [IEC-62443-24], [IEC-62443-3], [IEC-62443-31], [IEC-62443-33] and [ISO-24760-1]

[2] See [Ind4.0-SecId]

8.6 ENDPOINT ACCESS CONTROL

Endpoint access control depends on two related concepts: authentication and authorization.

Authentication is the provision of assurance that a claimed characteristic of an entity is correct. *Authorization* is the granting of rights, including granting access based on access rights. Authorization depends on verification of the mapping of identity of the entity compared to the rights and privileges on services and resources. Therefore, authorization is dependent upon authentication.

An entity comes in two forms: human and non-person entity (NPE). Both types of entities must provide credentials to assert their identity.[1] Credentials may be used for various purposes: authentication, identification and authorization. The secret parts of the credential required for authentication for both humans and NPE must be protected.

8.6.1 ENDPOINT AUTHENTICATION

The process of establishing trust through endpoint authentication, or identity assertion of the remote endpoint, has several steps. First, an attestation must be made that the credentials are of the proper level of strength, and that they are in the possession of the appropriate entity. Then, the actual value of data in the credential is evaluated for correctness. Finally, validity of the credential must be tested to ensure that the credential is not suspended, revoked or expired.

All successful authentication attempts do not result in the same level of trust in the identity of the remote endpoint. There are different levels of entity identity assurance based on what type of credential is applied to that authentication, how the credential is stored, and what actual authentication technique is implemented.

Strong cryptographic credentials are recommended for most endpoints. In addition, credentials should be stored in the strongest storage available, ideally in trusted hardware.

Mutual authentication is preferred over one-way authentication implementations wherever possible to prevent impersonation of the unauthenticated endpoint. Multi-factor authentication is recommended where possible for critical endpoints.

Application of more secure protocols that establish confidence in the remote endpoint identity wherever possible is recommended. Furthermore, implementation of proper authentication schemes that demonstrate possession and/or ownership of a credential while limiting exposure of the credential material should be part of the process for creating connections between endpoints. For example, implementing mutual authentication via Kerberos [MIT-Kerb] prior to establishing a Transport Layer Security (TLS) [IETF-RFC5246] tunnel is a common technique that avoids transmitting passwords over the network.

As part of the communication authentication process, the level of trust in the credential should be evaluated. Verification of the strength of the cryptographic algorithm used, capabilities of the

[1] See [ISO-29115]

hardware at the endpoint microcontroller, and evidence of the storage of the credential material, etc. may be required to evaluate the level of trust to grant to a successful authentication transaction.

8.6.2 ENDPOINT COMMUNICATION AUTHORIZATION

All communications between endpoints must be not only authenticated, but also authorized. Every connection attempt in an IIoT environment should be evaluated to determine whether it fits the endpoint or communication policy. Any such violation must generate an event notification, and may result in a block of the network connection attempt.

Authorizing a connection attempt involves asserting that the port, protocol, application, library and process is allowed via policy. Authorization may be enforced either on the endpoint or on the network. On the endpoint, much more information is available to determine the nature of the communication allowing for a more informed authorization decision.

8.7 ENDPOINT INTEGRITY PROTECTION

Measuring the device boot process enables the validation of its integrity, so we may assert that a device has powered up in a known good state. Given that devices may not be rebooted for long periods of time in OT environments, both static and dynamic integrity assurance of the runtime should also be implemented. Identity material must be properly secured in the trust roots to maintain its integrity and avoid identity spoofing, and data integrity must be monitored and maintained to establish trust in the data, including both data-at-rest and data-in-motion.

8.7.1 BOOT PROCESS INTEGRITY

The boot process initializes the main hardware components, and starts the operating system. Trust must be established in the boot environment before any trust in any other software or executable program can be claimed. So the booted environment must be verified and determined to be in an uncompromised state.

Measuring the boot-process enables the detection of manipulation of the host OS and software, so that malicious changes in the behavior of the devices can be detected. It enables boot-time detection of *rootkits*, *viruses* and *worms.*

The terms *trusted boot* and *measured boot* both refer to the process by which every entity in the booting sequence measures the next entity in the execution chain before executing it. It creates a chain of trust during the boot sequence whereby each element is measured and then executed (if in appropriate state) throughout the boot process. The measurements can be remotely attested and later used to evaluate trust on the endpoint.

Some boot-process-protection technologies interrupt the boot process if an improper component is detected. The term authenticated, *verified* or *secure boot* refers to technologies that interrupt and halt the booting process if the device is not in the desired state [BDI-CRTM].

A verified boot process is a type of trusted boot where the boot firmware and software is signed, but not measured; in this type of boot protection, the system will halt if the verification of a boot

component fails. However, if an attacker defeats the root of trust in a verified boot process, there is no way to determine that the system has been compromised.[1]

With measured boot, it is possible to boot a maliciously corrupted system, however, this can be detected through attestation. Halting the process with verified boot is a terminal state for the device; with measured boot, on the other hand, the system completes the boot process even with failed attestation, and is still in the position to be fixed.

The boot process protection provided by host processors can be enhanced with hardware. Depending on the host system and threat analysis, a choice can be made between several methods to involve the security hardware. Vendors of device CPUs promote the support of boot-process protection inside their processors. For PC-based systems, the *unified extensible firmware interface* (UEFI) specification[2] specifies a boot process whereby the validity of system firmware is checked along with features to block unauthorized writes to the flash. It also defines implementation guidelines and preliminary evaluation methods.

A significant problem with boot-process protection is the management of the measurements (integrity metrics). If the endpoint needs to be updated in the field, the integrity metrics approved for software and firmware in its boot process also need to be updated in a secure way. This management mechanism adds effort during the development and the lifetime of the device. The implementation of boot process protection on a device is not overly complex, but when this is extended to the whole system, implementation may require a significant effort. Normally, such complexity is managed by external management services, and is an integral part of supporting remote attestation protocols.

8.7.2 RUNTIME INTEGRITY

After the boot-process integrity has been attested to, the OS is running and applications can execute. Runtime integrity controls monitor, and ideally, enforce the integrity of the endpoint beyond the boot process.

Blacklist controls seek to identify files that contain malicious code elements, commonly known as *malware*. Blacklist controls define *signatures* that identify code elements as undesirable. It is challenging to provide a thorough blacklist of malicious indicators and keep the file size small enough on a resource-constrained endpoint. Moreover, new threats are constantly being discovered, so these definitions must be updated. The ever-present risk of an unknown vulnerability (a “zero-day vulnerability”) being exploited without being detected explains why blacklist technologies are commonly associated with traditional anti-virus products, and so more closely aligned with more loosely controlled IT operations rather than a safety-critical, and tightly regulated OT environment.

The obverse of blacklist integrity protection, *whitelist integrity protection*, seeks to identify only those files that are deemed “good.” Signing an executable and validating the signature prior to

[1] Physical attacks, such as booting from a USB drive, are common attacks to overcome verified boot.
[2] See [UEFI]

execution is one approach to creating some sort of cryptographic identifier that authoritatively confirms that the file has not been altered from its intended form. The whitelisting of files also protects against runtime integrity compromised by insertion of previous version of the file or incompatible version of files, such as known library files to be mistakenly or intentionally inserted in the system. In practice, many vendors avoid this technique because of the complexity in signing all the files during software development and release cycles. Alternatively, file hashing provides a separate ledger of hashes for allowed files. If a particular executable is not on the whitelist ledger, or the hash of the executable does not match the hash in the ledger, then its execution is blocked. All modifications to the ledger must be controlled and also be equally protected against tampering.

Memory-region protection controls memory-access rights, thus creating a TEE that prevents unauthorized access. Protection can be implemented in hardware, software, the OS, the separation kernel or the firmware. It is common to assign the memory regions during the boot process. This is especially effective in small, simple, resource-constrained devices.

Dynamic integrity controls include such applications as host intrusion detection (HID) or host intrusion protection (HIP) or runtime process integrity attestation controls. HIP monitor and analyze an endpoint, as well as the network traffic, looking for anomalous activity or known signatures that trigger alarms. HIPs may also monitor application access to protected resources, protected RAM, and privileged directories on the file system.

While there is no definitive best way to implement device integrity solutions, as much runtime integrity should be implemented as is possible within the constraints of the device.

8.8 ENDPOINT DATA PROTECTION

Securing data in endpoints involves *data-at-rest* (DAR) and *data-in-use* (DIU). The protection strategy for *data-in-motion* (DIM) differs at the edge, the cloud, and in the communications. Cryptography enforces data confidentiality and ensures integrity of the data. It may be used on all the data, only the sensitive portions or the entire storage medium. In practice, multiple data protection techniques may be applied simultaneously, providing protection from different types of attacks.

8.8.1 DATA CONFIDENTIALITY

Data confidentiality refers to ensuring that information is not disclosed to unauthorized parties. To implement this, cryptography renders data unintelligible to unauthorized entities that do not have the proper key for decryption of the data. The algorithm must be designed and implemented to ensure that no unauthorized party can determine the keys associated with the encryption or derive the plaintext. Data confidentiality is often mandated by regulations, in particular when privacy of the records is important or the record contains *personally identifiable information* (PII).

Some fields in a record may contain sensitive data that requires confidentiality while other fields need to be processed by an application. In this case, *data tokenization* can replace sensitive fields

or the value can be modified so confidentiality and privacy of those fields is preserved (Figure 8-3).

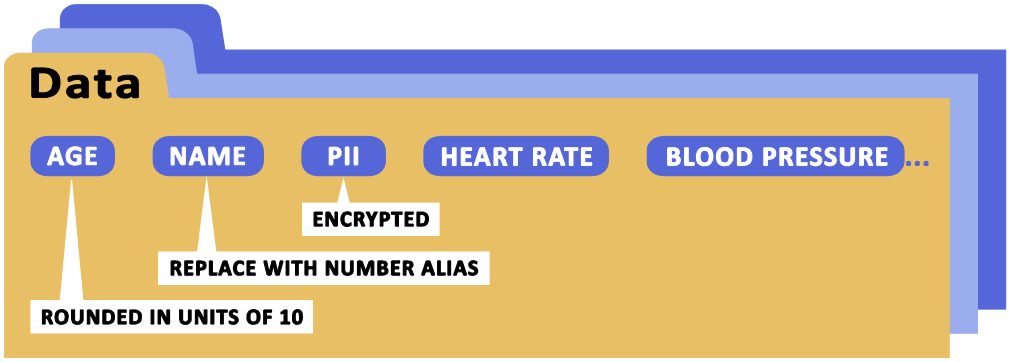

Figure 8-3: Example of Tokenization in a Medical Record

Data loss prevention (DLP) is commonly used to manage data confidentiality. DLP controls the usage of data, such as documents, records, emails, or any other sensitive data, in order to detect and prevent data breaches. DLP can either be endpoint-based or network-based. *Endpoint-based DLP* controls attempts to access or move data internally or externally of the endpoint. Internally, endpoint DLP controls and prevents data access across a physical device bus such as a hard drive, USB drive, or printer. Externally, endpoint DLP controls and prevents communications, including data before it passes over a network adapter. Network-based DLP relies solely on identifying confidential or sensitive information as it is being communicated between endpoints. Both attempt to identify violations of data use policy, but have different implementations.

8.8.2 DATA INTEGRITY

Data integrity assures that data alteration is detected. Traditional OT data integrity techniques (e.g. a CRC checksum) increase reliability and resilience of a system but are not effective against some malicious alterations due to their lack of cryptographic strength. Newer techniques such as digital signatures provide greater trust in the integrity measurements.

In general, data stored on the endpoint consists of two types: executable data (e.g. binary code and interpreted scripts), and non-executable data (e.g. raw data, configuration files, log files).

Non-executable data is operated on by executable data (code). The integrity of executable data is protected by runtime integrity techniques as explained in section 8.7.2.

The integrity of the non-executable data, the data-in-use, must be monitored while the data is being operated on. The DIU integrity is enforced by:

- proper coding techniques (such as using appropriate programming languages, implementing buffer-overflow protection, and strict checking of correct input parameters to prevent against injection attacks) and
- runtime integrity techniques that monitor memory access to detect and protect against memory attacks.

A common data integrity technique to detect alteration is the *digital signature*. The digital signature uses a secret or a private key to generate cryptographic signatures that record what the actual data was at the time of signing. This enables anyone to validate the integrity of the signed data at any point in the future, but requires more runtime processing effort to implement the cryptographic functions. Ideally, the signing key is kept in protected storage such as an HRoT, and the signing operation is performed in a TEE such as a TPM.

Applying digital signatures provides stronger integrity than hashing. In addition, since any party can validate the data, common security operations, such as software and firmware updates, can validate the integrity of the update prior to applying it. Also, configuration files and log files on the endpoint can be verified to ensure their integrity at any point in the future.

8.9 Endpoint Monitoring and Analysis

Monitoring mechanisms should also be protected. Endpoint monitoring concerns itself with detection of possible tampering with or compromise of devices, which would result in incorrect reporting of events. Monitoring of the endpoint security status may be performed internally on the endpoint or may be performed externally to the endpoint. Monitoring of least-capable edge devices will most likely be executed from another endpoint in the operational domain.

8.10 Endpoint Configuration and Management

The endpoint must provide secure and controlled changes to the endpoint components, though in some rare cases no security is desired. All updates and changes should be signed, their payload encrypted and actions logged for subsequent auditing and recovery of the endpoint. These services should be provided non-intrusively to the operational functionality and have a separate logical connectivity to system-level configuration management and control.

8.11 Cryptography Techniques for Endpoint Protection

Cryptography is the discipline that embodies principles, means, and mechanisms for the transformation of data in order to hide its information content, prevent its undetected modification and prevent its unauthorized use. Cryptography is used to perform a number of security operations at the endpoint. Providing an in-depth description of cryptographic techniques and algorithms is out of scope of this document. The following information clarifies some concepts and is included for completeness.

Endpoints must always use standard cryptographic algorithms. These algorithms should be implemented utilizing safe-coding practices, and whenever possible, with libraries that are updated and maintained regularly. Creating cryptographic algorithms without a public evaluation should be avoided.

In addition, keys must be random, not predictable, and of sufficient length to preclude brute force or exhaustive searches of the available key space. Two classes of *random-number generators* (RNG) are commonly used: deterministic and non-deterministic. Deterministic RNG (also called Pseudorandom Number Generators) use a secret starting value, called a *seed*, to initialize the generation algorithm, while non-deterministic RNG depend on some unpredictable

physical source that cannot be easily controlled. RNG should be provided by the hardware of the device, but this can be challenging in resource-constrained devices.

The length of time that a specific key is valid for use by legitimate entities is called the *cryptoperiod* [NIST-KEYM]. Cryptoperiods limit the exposure if a single key is compromised. When a key is compromised, a key revocation process must be in place to notify that the keying material is invalid before its cryptoperiod has expired. Unfortunately, the process associated with changing keys may be complex, so a *key management system* that automates the various steps in the key management is recommended.

Not all endpoints require cryptographic controls. In some cases, data may be publicly available, not requiring any confidentiality controls. In other cases, redundant sensors may be reporting the same measurement; so tampering with any of the sensors' data could be detected, removing the need for cryptographic integrity controls. Other surrogates, such as gateways, may be performing cryptographic operations on behalf of the endpoint.

Embedded designers may offload some, or all, cryptographic operations in computing resource-constrained devices to secure *microcontroller units* (MCU). The most common motivation is a desire to keep cryptocredentials in a secured environment along with increased performance and reduced burden on main processor. A need for secure random number generation can also be a factor. More secure MCU have high-quality random generator modules, cryptographic engines built with countermeasures to address physical attacks or a strong, unique public/private key-pair injected at manufacturing time.

Training and organizational maturity are required to deploy security correctly. For example, well-established cryptographic algorithms with appropriate key sizes and key management are required.

8.12 Isolation Techniques for Endpoint Protection

Isolation refers to the technique used to shield a component of a system from unwanted effects where an element of the endpoint cannot be affected by other elements of the endpoint, thus shielding its functionality from failures and malicious activity.

There are several isolation models. Each is described in turn.

8.12.1 Process isolation

The process isolation model relies on the operating system to isolate business or operational components from the security components at the process level (see Figure 8-4, left). Hierarchical protection domains protect functions and data from inadvertent or malicious failure acting as a gate to protect more privileged layers from less privileged layers.

Process isolation is the predominant security deployment model in the industry today. However, compromising any component within the operating system, including applications and libraries, breaks the integrity of the device and may form a foothold for further attacks.

Examples of process isolation include security agents, software libraries that perform security operations, a software key store and any directory and file access control lists that depend on OS enforcement of the security.

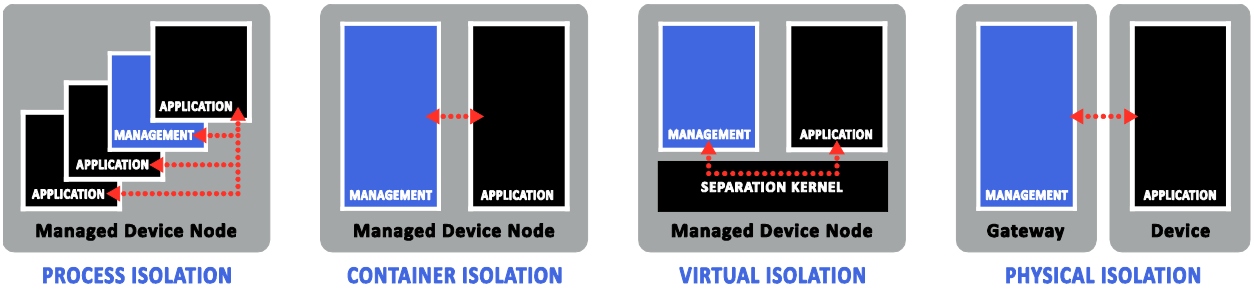

Figure 8-4: Endpoint and Container Isolation Techniques

8.12.2 CONTAINER ISOLATION

The container isolation model implements either hardware-or software-enforced boundaries (see Figure 8-4). Software containers rely on the OS to enforce the resource isolation boundaries; hardware containers use a physically different compute element on the same platform. Hybrid containers combine both approaches.

Examples of software containers include:

- Operating system-managed containers such as Android (Trusty TEE) or Linux Containers such as LXC and Docker. [1]
- Secure memory mapping that provides appropriate entry/exit locations for security to be implemented down to very small sensor-type devices.
- Network interface controllers that embed policy and enforcement directly on the hardware of the network interface so that only a predefined set of source/destination, port and protocol combinations from the security policy can communicate to/from the endpoint. All other communication attempts result in failure.

Hardware containers separate the security implementation by enabling a separate compute engine, either on the same chip or on the same board, or on a daughter board in the same physical entity. This creates a security coprocessor that implements some level of security functionality that is separate from the main processor's compute engine. Common examples of hardware containers include:

- TPM: The TPM (see section 8.2.2) is a trusted execution environment (hardware root of trust) that provides secure storage of credentials, and protected execution of cryptographic operations. It is isolated from the main CPU, and implemented either as a discrete chip, a security coprocessor (see below), or in firmware.

[1] See [Andr-Trusty], [LinuxC-LXC] and [Docker]

- Security coprocessor: Building an off-CPU security presence in a trusted execution environment (including a hardware root of trust) on a separate chip, enables a number of security capabilities to be implemented including all of the TPM-type operations, but also additional integrity controls, security for communications, event monitoring, security analytics and other security-related operations. The key to this approach is having the security elements deployed on a physically separate chip.

8.12.3 VIRTUAL ISOLATION

The virtual isolation model—sometimes referred to as hypervisor isolation—uses a hypervisor to implement isolation between each virtual instance running on the device. As a result, one of the instances running on the hypervisor can be a security instance that acts as a TEE on the device. The virtual instance TEE may store confidential information, such as identity material, and may implement security controls such as mutual authentication, connection authorization, cryptographic functions, firewalling, deep packet inspection, integrity controls and remote boot attestation functions. The device boot process often measures the hypervisor for integrity, and the hypervisor then measures each virtual instance before starting it, thereby extending the chain of trust into the virtual TEE such that the integrity can be assured immediately after boot. After boot, runtime integrity controls must ensure that the virtual TEE integrity remains intact.

One of the advantages of the virtual TEE comes in the form of consolidation of multiple platforms on the same physical hardware. This follows the cloud model for consolidating a number of physical servers onto a single hypervisor to benefit from the economies of scale. This enables, for example, combining Programmable Logic Controller (PLC) logic and a Windows Human Machine Interface (HMI) on the same physical device in an IIoT environment.

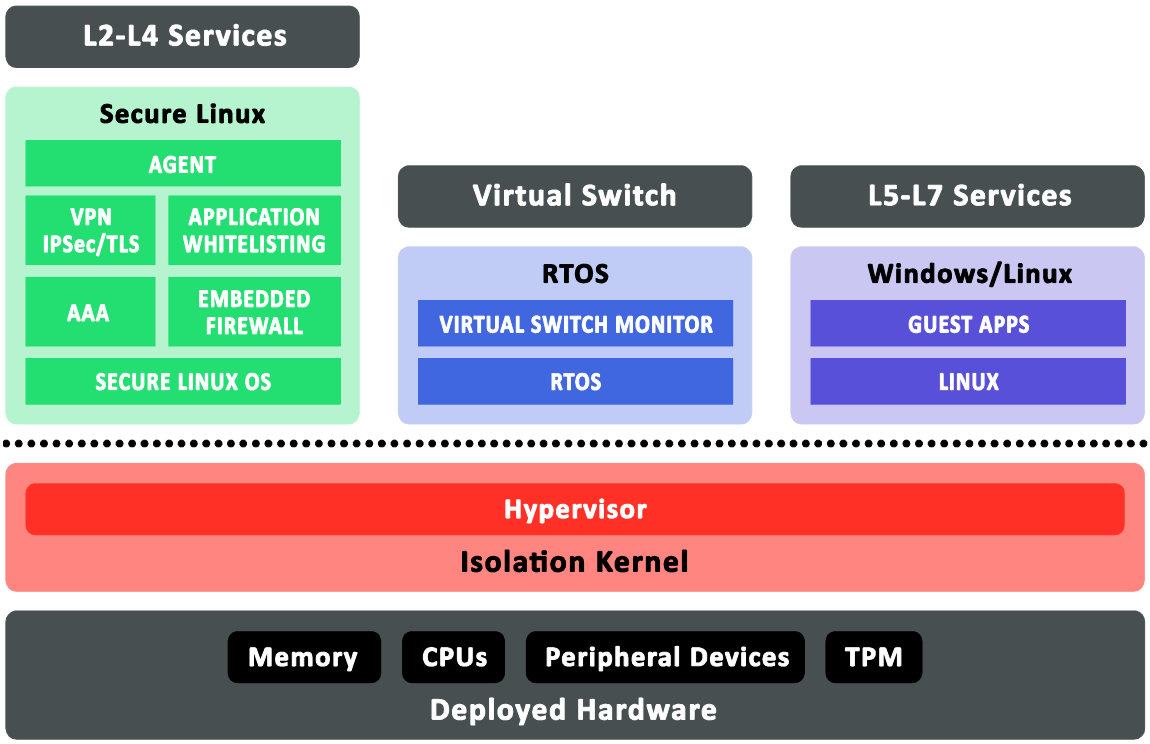

Figure 8-5: Virtual Isolation

Virtual isolation enables the same economies of scale that have driven the growth in cloud adoption. On the edge, virtualization enables OT components to function without change in their existing operating system, while allowing security functions to run independently in its own OS. As the security OS is on the same physical device as the OT operating system, it can provide many controls such as embedded identity, secure boot attestation and communication interceptor pattern, all below the OT operating environment.

Virtual isolation augments brownfield software deployments with security capabilities below the OS. The security does not reside in the guest OS, but rather in a dedicated security OS, acting as a TEE, that implements many of the security operations on behalf of the guest OS. This is analogous to deploying a gateway inside the device, rather than in front of it. The advantage for brownfield deployments is that it doesn't require changes to guest source code and that the application itself is oblivious to the existence of the security OS protecting it.

Separation kernels are a specific form of virtual isolation. They provide strong isolation that covers all the resources provided by the underlying hardware platform (processor time, memory and I/O devices). In addition to isolating components from each other, they also enable communication control between components and devices according to a security policy.

In contrast with monolithic hypervisor kernels, separation kernels do not implement many services commonly associated with operating systems, such as device drivers, file systems and network stacks. Separation kernels exist to provide separation between components and enable controlled communication among them. By intentionally limiting the functionality of the kernel to isolation and simple IPC primitives, separation kernels have greatly reduced attack surface and implementation complexity.

8.12.4 PHYSICAL ISOLATION

Physical isolation moves the security to a completely separate device. A separate device such as a gateway provides the security. This is discussed in section 8.2.3.

8.13 RESOURCE-CONSTRAINED DEVICE CONSIDERATIONS

Resource-constrained devices have the same security requirements as more powerful devices, including run-time protection, boot-time protection, communication authentication, configuration management and contribution to larger analytics systems.

Resource-constrained devices must be able to perform crypto operations. Newer devices are capable of performing crypto operations using hardware accelerators, co-processors, and embedded accelerators. These are often integrated through *system on chip* (SoC) designs, where a single integrated circuit integrates not only the CPU, but also the network controller and other features. *Field programmable gate arrays*, which may also have a CPU co-processor, are another popular SoC solution for accelerating crypto operations since the algorithms can be updated in the future. All of these techniques greatly increase device performance and battery life. It is now possible to build endpoints that combine embedded acceleration with new algorithms to provide the best compromise between upgradability, performance and security.

Device manufacturers now implement embedded cryptographic capabilities into crypto-accelerators that occupy only a small portion of the real estate on a single chip, through SoC designs. Some of these crypto-accelerators are built to hardware security module standards.

Although such chips and algorithms make it easier to build security into new devices, many manufacturers are saddled with decades-old devices that do not have such capabilities. For these systems, either the manufacturer must update the firmware to support new software supporting efficient software crypto operations and protocols, such as *ID-based encryption*[1], or they must be implemented in gateway devices.

Unfortunately, many industrial protocols do not yet support adequate authentication, but insecure protocols can be tunneled over Transport Layer Security (TLS) and other lower-layer protocols to provide needed security properties such as authentication. Alternatively, individual commands, messages and datagrams are sometimes authenticated at a higher *data object* layer without trusting, and without needing to trust, the underlying protocol.

If the device is able to perform state-of-the-art cryptographic operations, then it can verify the integrity, authenticity, pedigree and authorization of specific firmware to run. Furthermore, it can authenticate connection requests.

[1] See [Fuji-MAT]

RSA is one of the most widely used asymmetric cryptographic algorithms. Other algorithms such as those based on elliptic curves[1] can provide similar cryptographic strength as RSA, but with smaller key sizes, offering benefits such as lower space and processing requirements[2]. For example, a 283-bit ECC key is equivalent to a RSA 3072 bit key[3]. This means elliptic curve cryptography (ECC) algorithms may be more suitable for resource-constrained endpoints. Many parameters must be considered in the choice of elliptic curve algorithms as described in [IETF-RFC6090].

Configuration management can be done securely, and the device can safely contribute security telemetry to broader analytics systems in ways that the device's telemetry can be authenticated. Run-time security can be provided either in-device, or in a trusted gateway.

Implementing hardware acceleration in a *field-programmable gate array* (FPGA) enables algorithm agility, which allows changing algorithms in the future due to security considerations. custom *application-specific integrated circuits* (ASIC) cannot be changed, which is an important consideration for long-lived devices.

Other constraints include wireless limitations, battery consumption, intermittent availability of communications and constraints on maintenance windows, making updates less frequent. This forces run-time security to be based on whitelists instead of blacklists, and increases dependency on third-party security. Support for updates as small as 40K bytes, in contrast to gigabyte-sized images, makes it possible to update with orders of magnitude less bandwidth and battery consumption compared to monolithic updates. Other impacts of unreliable communications include careful consideration of key management and key revocation strategies, sometimes declining Certificate Revocation Lists (CRL) in favor of Online Certificate Status Protocol (OCSP) or OCSP stapling, Short Lived Certificates (SLC), or evergreen certificates depending on the specific constraints of the specific system.

[1] See [IETF-RFC7027]
[2] See [Sym-ECC]
[3] See [IETF-RFC5480]

9 PROTECTING COMMUNICATIONS AND CONNECTIVITY

The communications and connectivity function in Industrial Internet of Things systems supports exchange of information among endpoints. It provides interoperable communications to facilitate component integration. The level of protection required depends on the threats to such information exchange. This information can be sensor updates, telemetry data, commands, alarms, events, logs, status changes or configuration updates.

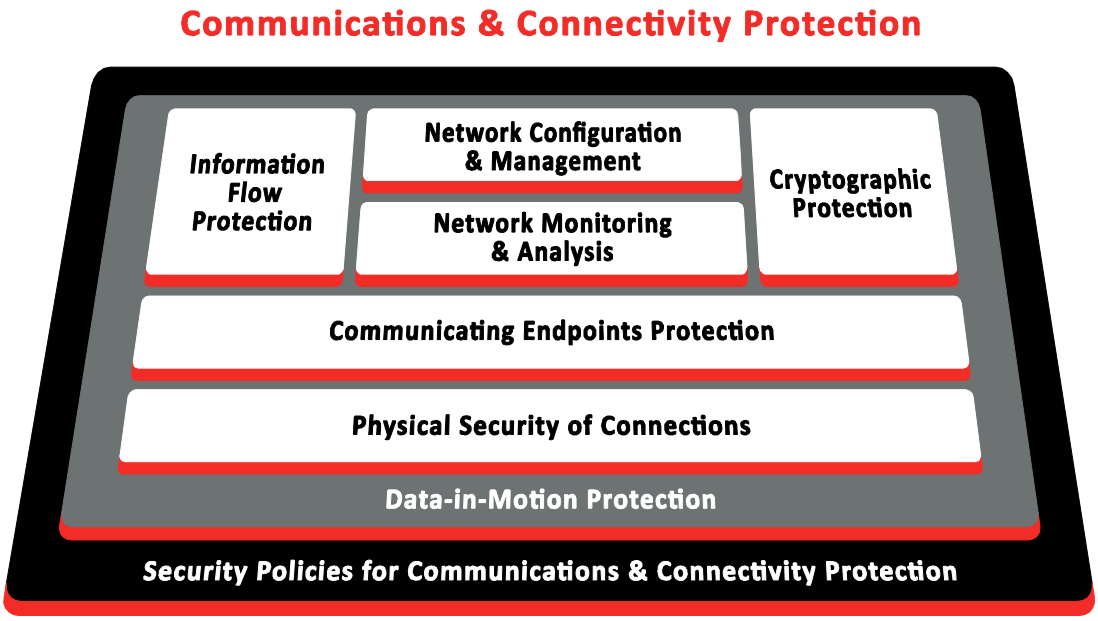

Figure 9-1: Functional Breakdown for Communications and Connectivity Protection

Historically, industrial systems have emphasized information flow protection over cryptographic technologies. More recently, IIoT applications employ cryptographic controls such as those applied at transport layer (e.g. TLS or DTLS) or middleware layer (e.g. DDS)[1]. IIoT systems will most likely use both classes of techniques, as each class of technology protects against different set of network attacks.

These measures can only be effective if communication and connectivity are available, so risks associated with denial of service attacks on networks should be assessed and controls put in place. These controls include physical security, capacity planning, load balancing and caching. Authorization techniques that enforce principle of least privilege and intrusion detection techniques that alert or block offending connections also help.

[1] See [IETF-RFC5246], [IETF-RFC6347] and [OMG-DDS]

9.1 Cryptographic Protection of Communications & Connectivity

Most IIoT applications should use standardized protocols whose functionality, including security and cryptography, have been evaluated and tested. IIC's 'Industrial Internet Reference Architecture'[1] identifies and discusses requirements for IIoT core connectivity protocols.

9.1.1 Security Controls in Communication and Connectivity Protocols

From an architectural standpoint, information exchange among different actors within a system happens over two abstract layers: a communication access and transport layer (corresponding to Layers 1 to 4 of the OSI model) that provides for exchange of bits and bytes, and a connectivity framework layer (corresponding to Layers 5 through 7) that uses the communication transport to provide syntactic interoperability among actors by exchanging structured data. The figure below shows these abstract layers.

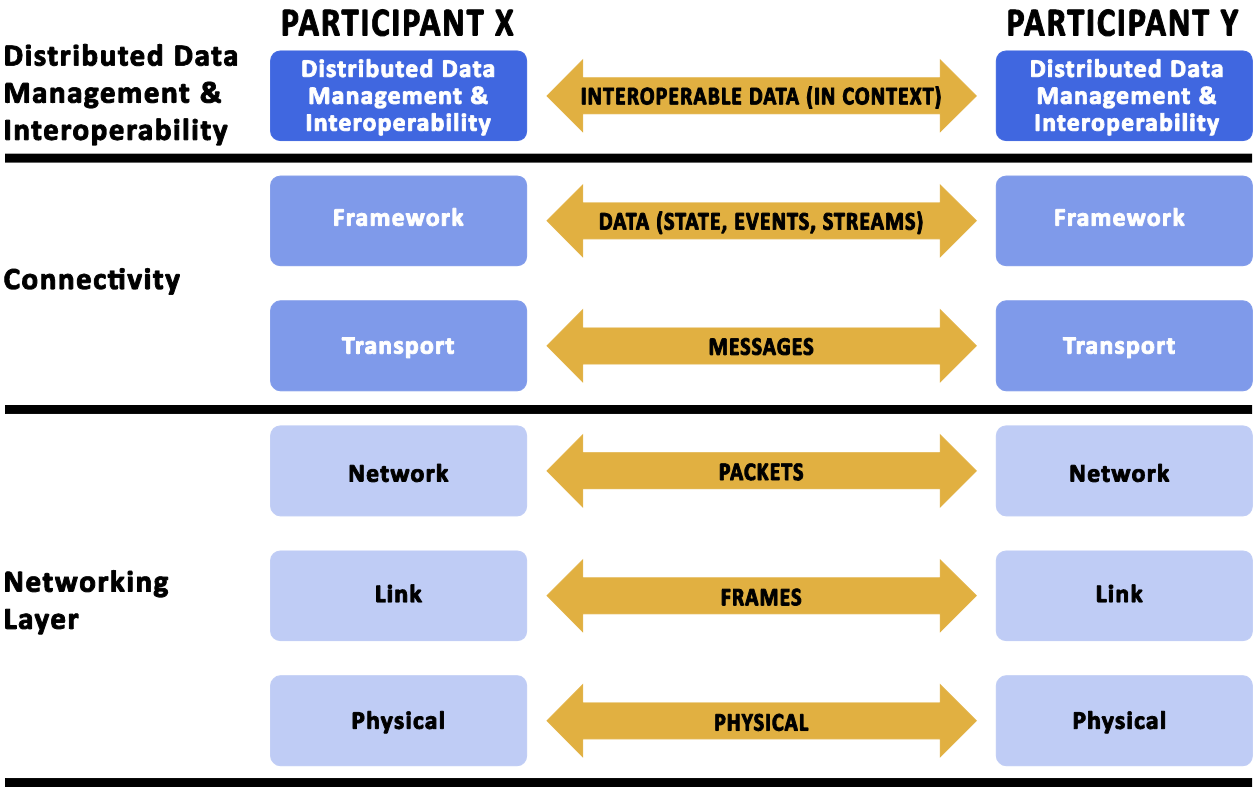

Figure 9-2: Communication and Connectivity Layers

Protecting communication links at each layer requires corresponding security controls and mechanisms applicable to that layer. Two important design questions are choosing which layer(s) to protect, and how to customize that protection for a given application.

Security controls in all layers may induce unacceptable performance costs, but securing communication only at the lower levels (e.g., IP level, with *internet protocol security* (IPsec) or

[1] See [IIC-IIRA2016]

transport level with TLS or DTLS) may not provide sufficient security for application-level traffic that requires fine-grained security controls.

9.1.2 Building Blocks for Protecting Exchanged Content

Where possible, information exchange security among communicating endpoints for sensitive networks and equipment should employ:

- explicit endpoint communication policies,
- cryptographically strong mutual authentication between endpoints,
- authorization mechanisms that enforce access control rules derived from the policy and
- cryptographically backed mechanisms to ensure confidentiality, integrity and freshness of exchanged information

A first step in establishing secured communication is mutual authentication using cryptographically backed authentication protocols (i.e., by exchanging identity certificates, if a public-key infrastructure is set up). The parties must then exchange data according to the access control rules defined in the policy. For example, an endpoint collecting medical metrics that has been deemed authentic may not be permitted to share some patient data.

Confidentiality and integrity of exchanged messages should be achieved using standard techniques for encryption (i.e., symmetric algorithms such as AES and asymmetric algorithms such as RSA) and message authentication (i.e., digital signature schemes such as DSA and message authentication codes such as HMAC). These techniques often use cryptographic keys established during the mutual authentication process; encryption without message authentication should be avoided.

Communication protocols that do not provide integrity and confidentiality of exchanged messages could be routed through encrypted and authenticated tunnels or otherwise be contained by information flow control techniques. This improves security of legacy protocols.

9.1.3 Connectivity Standards and Security

A core connectivity technology, as defined in 'Industrial Internet of Things, Volume G5: Connectivity Framework'[1], should:

- be an open standard with strong independent, international governance, such as IEEE, IETF, OASIS, OMG, or W3C,
- be horizontal and neutral in its applicability across industries,
- be applicable, stable and proven across multiple industries and
- have standard-defined gateways to all other connectivity standards.

[1] See [IIC-IICF2017]

The figure below shows prominent communication and connectivity standards at different OSI layers. An in-depth discussion of connectivity assessment is provided in the 'Industrial Internet Connectivity Reference Architecture'.

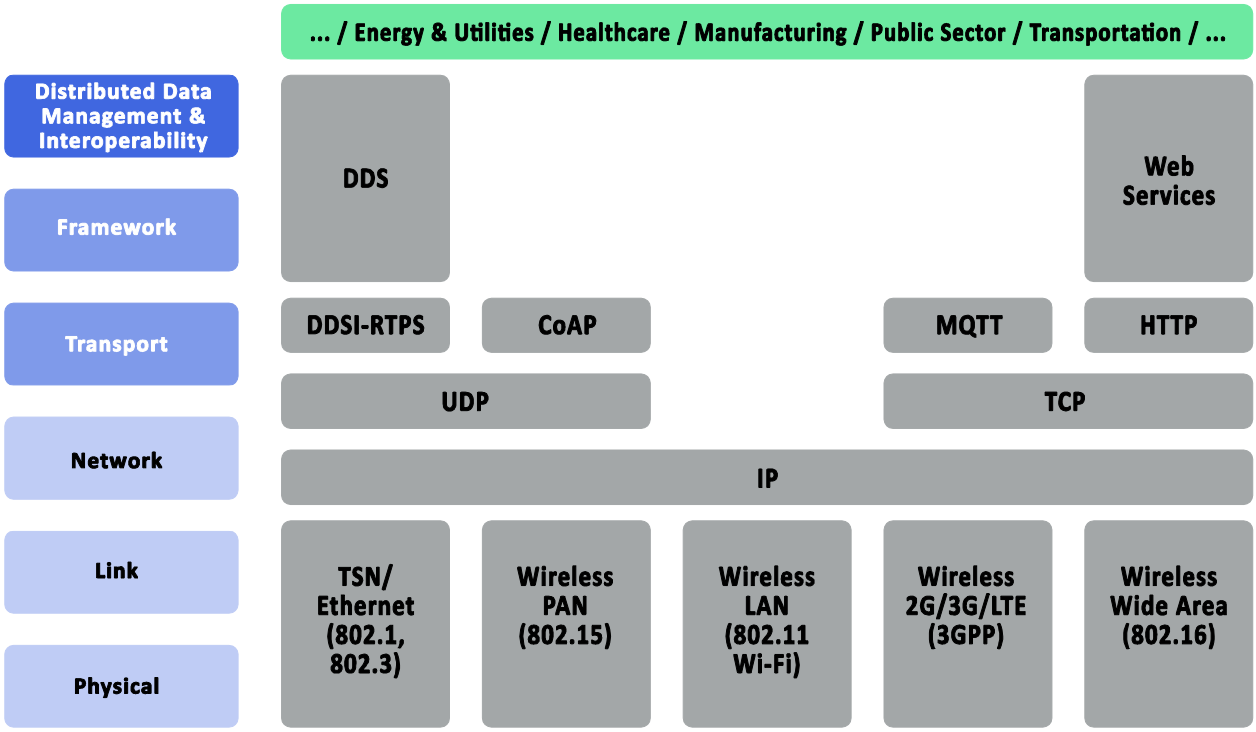

Figure 9-3: Example of IIoT core Communication & Connectivity Standards

9.1.4 Cryptographic Protection for Different Communications and Connectivity Paradigms

Different information exchange patterns have different security requirements. Widely used patterns in IIoT systems include request-response pattern and publish-subscribe pattern.

The *request-response pattern* can be used at any layer of the stack. Protocols using this pattern include Java Remote Method Invocation (Java RMI), Web Services/SOAP, Remote Procedure Call over Data Distribution Service (RPC-over-DDS), Open Platform Communication (OPC), Global Platform Secure Channel Protocol and Modbus. They vary in their support for security; For example, Modbus can't suppress broadcast messages, doesn't provide message checksums and lacks support of authentication and encryption.

The primary types of threats for *publish-subscribe communication pattern* are unauthorized subscription, unauthorized publication, tampering and replay and unauthorized access to exchanged data. Some implementations of this pattern (e.g., classic MQTT and AMQP) rely on intermediary message brokers store-and-forward messages, but the message broker could be a single point of failure. An alternative approach is broker-free, peer-to-peer implementations such as the DDS standard.

9.2 Information Flow Protection

Information flows are any information in motion, including IP messages, serial communications, data flows, control signals, removable media, printed reports and data carried in human minds. Controlling different types of information flows protects them against attackers.

Online information flows are generally the flows most accessible to remote attackers bent on sabotage or data theft by pivoting through intermediate systems and networks.

9.2.1 Controlling Information Flows in Brownfield Deployments

It can be costly to recertify the safety and reliability of hardware and software components. For example, regulations for discrete manufacturing in some jurisdictions demand that certain classes of automated equipment can operate at a manufacturing site only if all the equipment, hardware and software, has been safety-certified by a third party. None of it may be put into production without recertification. Vendors using commercial operating systems are often unwilling to pay the cost of recertification for security updates, technologies and methods. Consequently, equipment is often out of date. Even brand-new equipment may need:

- physical security measures to prevent unauthorized personnel from physical contact with sensitive equipment and networks,
- network perimeter security controls to prevent unauthorized messages from reaching sensitive equipment and networks and
- passive network intrusion detection to monitor suspicious communications patterns.

These approaches have been preferred for brownfield networks because they do not change any parts, and so do not require recertification. Whether that is sufficient for a given system should be determined during risk analysis.

9.2.2 Network Data Isolation

A *channel* is an independently identified, managed and monitored data flow at the transport, framework or application layer. There are three basic communications channels that are commonly defined: data, control and management channels. Each channel should be isolated from the others and managed and monitored separately, for example by using separate TCP connections, separate wireless frequencies, or separate publish/subscribe topics on a common event bus or message broker.

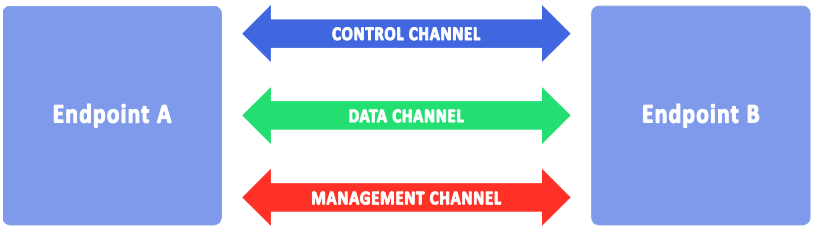

Figure 9-4 Communications Channels between IIoT Endpoints

The data channel, sometimes called the operational monitoring channel, is used to report operational information, and the state of the endpoint. The control channel is used to alter the behavior of the industrial process, and alter the state of the endpoint. The management channel carries administrative traffic such as machine profiles, security policies, endpoint configuration changes and access control settings. For example, a power meter may use separate data, control and management TCP/IP sessions, to report usage, remotely connect and disconnect electric service and update firmware versions, respectively.

Using separate communications channels can reduce the cost and complexity of managing and monitoring each kind of communication. There may be multiple instances of each type of channel active at any time on a given endpoint. Separate security controls can be defined for each channel. These include technical confidentiality controls such as encryption, network segmentation and communications authorization, as well as integrity controls such as message signing. Separate *quality of service* (QoS) requirements may also be applied to each of the channels to ensure message delivery within defined tolerances. See section 11.2 for a detailed discussion of the management channels.

When using bi-directional protocols to communicate across trust boundaries, even "pure" monitoring channels can pose the threat of potential unauthorized access of IIoT endpoints, since any message permitted into a safety-critical or reliability-critical network segment might encode a platform-level attack, such as those based on buffer overflows.

9.2.3 NETWORK SEGMENTATION

Networks cannot be interconnected indiscriminately. Industrial security standards such as ISA/IEC 62443-1-1, ISA/IEC 62443-3-3, ANSSI, NIST 800-82[1] and others all recommend separating networks into segments, each segment containing assets with similar security policies and communications requirements. They also recommend assigning each network segment a trust level, and protecting communications and connectivity through the perimeters of networks, especially between segments at different trust levels. For example, no site would intentionally expose a safety-critical device to the internet, because there's no reason to allow attackers to reach safety-critical equipment. There would always be a residual risk, no matter how thoroughly the device is hardened.

Network segmentation can be fine-grained or coarse-grained. Candidates for segmentation include public networks (such as the internet), business networks, operations networks, plant-wide networks, control networks, device networks, protection networks and safety networks. Fine-grained segmentation is generally better but it is usually costlier to maintain.

Security and device management networks are often candidates for segmentation. LAN and WAN networks permit IT-like management communications such as backups, security logging and updates to take place without interfering with time-critical or sensitive operations and communications. Segmentation can provide useful traffic management, but may be of limited security value because of the size of the attack surface—every dual-ported device with access to

[1] See [IEC-62443-11], [IEC-62443-33], [ANSSI-CMKM] and [NIST-800-82]

both management and operations networks can serve as a pivot point for an attack jumping from one network to the other.

9.2.4 GATEWAYS AND FILTERING

Gateways control information flow between network segments. 'Industrial Internet Reference Architecture'[1] defines a gateway as a "forwarding component enabling various networks to be connected." This definition is very general, and describes any computing device with two or more network interfaces that forwards information between those interfaces.

Gateways may transform and forward information between segments without additional controls; for example, a protocol-translating gateway may translate legacy, insecure communications protocols into modern, encrypted protocols. Gateways may also filter information flows in many ways; for example, a firewall forwards only messages that match specific rules, and a unidirectional gateway is physically able to transmit information in only one direction, and blocks all communications in the other direction.

Gateways with filters are used to implement network segmentation by controlling the flows of information passing between network segments. These filters may be bidirectional or unidirectional: a bidirectional filter forwards information both into and out of a connected network, while a unidirectional filter forwards information exclusively into or out of one or more network segments. Filters may also be message-based or information-based. Message-based filters preserve message structures at a certain layer of a protocol stack, and forward or do not forward messages at that layer. Information-based filters extract certain kinds of application-level information from one or more messages from a network interface, and forward that information into another network while preserving no part of the originating network's message structures.

Gateways may encode significant application functionality. For example, a dual-ported historian server at an IT/OT interface can be thought of as a bidirectional informational gateway with significant persistence and analysis capabilities. The historian server uses device communications protocols to gather data from the OT network via one network interface, and uses client/server protocols to publish data into the IT network via a second network interface. In another example, a Data Distribution Service (DDS) gateway often translates information streams at an application/middleware level, while also enabling secure persistence, secure distributed logging and secure data transformation.

Different kinds of gateways provide different degrees of security benefits. Legacy gateways can translate encrypted, authenticated communications into less-secure communications for legacy end devices so they can also participate in modern networks. Unidirectional gateways are physically unable to forward any information or attack back into protected networks. Gateway security capabilities should be matched to security needs carefully when they connect network segments at different trust levels. Unhardened gateways should not connect a network of legacy safety and control devices to a corporate network, or to the internet.

[1] See [IIC-IIRA2016]

Gateways with filters control the flow of information passing between network segments. Message filters control the flow of messages at some layer of a protocol stack, while application gateways tend to control information flows more abstractly. Firewalls are examples of bidirectional message-filtering gateways embodying many security features.

Examples of important IIoT filtering technologies include:

Air gaps are network segments with no online connection, wired or wireless, to any external network. Air gaps are the strongest form of filtering, but provide none of the connectivity benefits.

Layer 2 filters separate physical network signaling systems, but forward Open Systems Interconnection (OSI) Layer 2 network frames. Managed switches and bridging firewalls are examples of technologies that filter messages based on Ethernet Media Access Control (MAC) addresses or other device-level addressing. Virtual Local Area Networks (VLAN) switches are used for traffic management, but they are not security devices so they are not recommended as perimeter protection mechanisms between network segments at different trust levels.

Layer 3/4 filtering: The most commonly used IIoT message filters are firewalls able to filter messages based on network addresses, port numbers and connection state. Such filtering technologies are known as *packet filters* and *stateful inspection*.

Application and middleware layer content filtering: Some firewalls and other message filters understand specific communications protocols and are able to filter messages based on application content. For example, an application layer filter might permit device register read requests, but block write requests. Other filters might permit messages from a particular user, but not other users. This is called *deep packet inspection*.

Message rewriting: Some message filters modify messages as they pass through the filter. For example, *network address translation* (NAT) filters change IP addresses and port numbers, and v*irtual private network* (VPN) servers encrypt and decrypt message streams. VPN are often deployed in IIoT systems to help protect interactive remote access mechanisms, and to encapsulate and protect plain-text device communications protocols as they pass across WAN.

Proxies are application-layer message filtering with message-rewriting capabilities. Typically, proxies maintain at least two similar transport-level connections: one to a device on a protected network, and one to a device on an external network. Proxies may answer queries or serve other protocol requests out of their own caches and data storage, or they may forward requests to external data repositories.

Server replication: Server replication maintains a real-time copy of part or all of a protected industrial server on a less-trusted network segment, most commonly at IT/OT network perimeters. For example, a plant historian server may be replicated through an IT/OT firewall. The replication mechanism can act as a filter by replicating only a subset of historical data points out to the corporate network.

Virtual networks: Virtual networks may implement message filters in hypervisors or virtual firewall hosts.

Most of these message filters can be implemented in gateway host or device software, or as real or virtual network appliances. In hosts or devices, these filters control messages and information exchanges for a single endpoint. As real or virtual network appliances, gateways with filters can control messages and information flows for entire network segments.

9.2.5 NETWORK FIREWALLS

Network firewalls are message-oriented filtering gateways used extensively to segment IIoT systems. Most firewalls are Layer 2, 3 or 4 IP routers/message forwarders with sophisticated message filters. Firewalls may be deployed as either physical or virtual network devices. A firewall's filtering function examines every message received by the firewall. If the filter determines that the message agrees with the firewall's configured traffic policy, the message is passed to the firewall's router component to be forwarded. Firewalls may also rewrite messages, most commonly, via performing encryption or *network address translation* (NAT).

In addition, a full-featured firewall may include the following features:

- *virtual private networks* with the ability to forward messages through an encrypted tunnel,
- *user accounts* requiring users to authenticate with the firewall before message forwarding is enabled for that user or for the user's computer,
- *inline anti-virus scanning* allowing files to be scanned with anti-virus scanning engines while in motion via FTP, SMTP, HTTP or other protocols that commonly carry files,
- *inline intrusion detection* allowing packets in motion through the firewall to be scanned with intrusion detection engines and
- *inline intrusion prevention* allowing packets in motion through the firewall that match intrusion detection signatures to be dropped.

Device firewalls are designed to protect endpoints. They may be conventional firewalls with deep packet inspection capability or Layer 2 IP routers with deep packet inspection filters. The latter can be deployed without reconfiguring routes in existing, endpoint devices.

Learning-type filters and configurable filters may be used for device firewall application-level filtering. Learning filters monitor traffic for a period of time, and automatically create filtering rules to identify all observed traffic as normal and permitted. Once the learning mode is complete, the firewalls can be configured to forward only traffic that agrees with the filters, and to drop all other traffic. Configurable filters can be set up to permit some application-level content, and to forbid other content. For example, one might be configured to permit writes to certain device registers and not others, or to permit reads and writes of any registers, but not downloads of firmware.

9.2.6 UNIDIRECTIONAL GATEWAYS

The term *unidirectional gateways* is used by IEC 62443-1 and NIST 800-82[1] standards to refer to devices that can replicate servers and emulate devices via communications hardware that physically permits information to flow in only one direction.

Currently, unidirectional gateways are deployed most commonly at the IT/OT network interface in large industrial facilities and at the LAN/WAN interface in smaller facilities, such as remote substations and pumping stations. When they are deployed as the sole online connections to a trusted network segment, no online attack from any external segment can affect the operation of the trusted network segment.

Unidirectional gateways using optical isolation have a fiber-optic laser as a transmitter, but no receiving hardware. A receiving module contains a fiber-optic photocell as a receiver, but no transmitter. A short fiber-optic cable connects the two modules. Other unidirectional gateways use electrical isolation.

Unidirectional server replication copies queries servers on a source network, filters the information and transmits it unidirectionally to a destination network. In the destination network, the replication technology inserts data received from the unidirectional gateway into a replica server. Users and applications on the destination network query the replica for information. No query can be forwarded from the destination network to the source network.

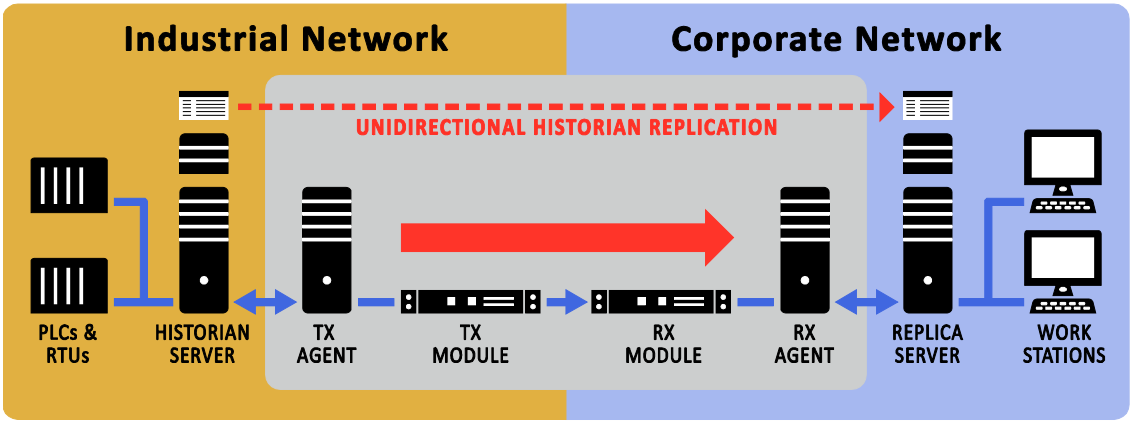

Figure 9-5: Unidirectional Plant Historian Replication

Figure 9-5 illustrates a typical unidirectional gateway deployed at an IT/OT interface replicating a plant historian server to a corporate database. The transmit (TX) agent queries the plant historian server for historical data points and pushes it to a corporate IT network across the unidirectional hardware. The receive (RX) agent uses the historical data to populate a replica historian server. External users and applications query the replica to access historical data. No attack from the corporate network or on the corporate historian server can affect the operation of a unidirectionally protected plant network.

[1] See [IEC-62443-11] and [NIST-800-82]

When emulating devices, unidirectional replication software on the source network sends snapshots of source device states to the destination network. The replication software on the destination network emulates the source devices, responding to polls or other queries as those devices would have responded. For example, Open Platform Communications (OPC) servers can be replicated unidirectionally to supply data to enterprise historian servers reducing risk of an attack.

Unlike firewalls, a unidirectional gateway generally does not forward messages from source networks to destination networks, as the gateway software maintains independent communications connections on each. The gateways are physically connected to the hosts running the unidirectional replication software packages, and so forward only unidirectional application replication information flows.

A periodically reversible unidirectional gateway can be deployed when periodically scheduled updates are needed for unidirectionally protected networks. Figure 9-6 illustrates an optical unidirectional gateway with electromagnetic switches to control copper connectivity to the optical hardware. The switching permits a unidirectional connection into a protected industrial network, or out of that network, but never both at the same time.

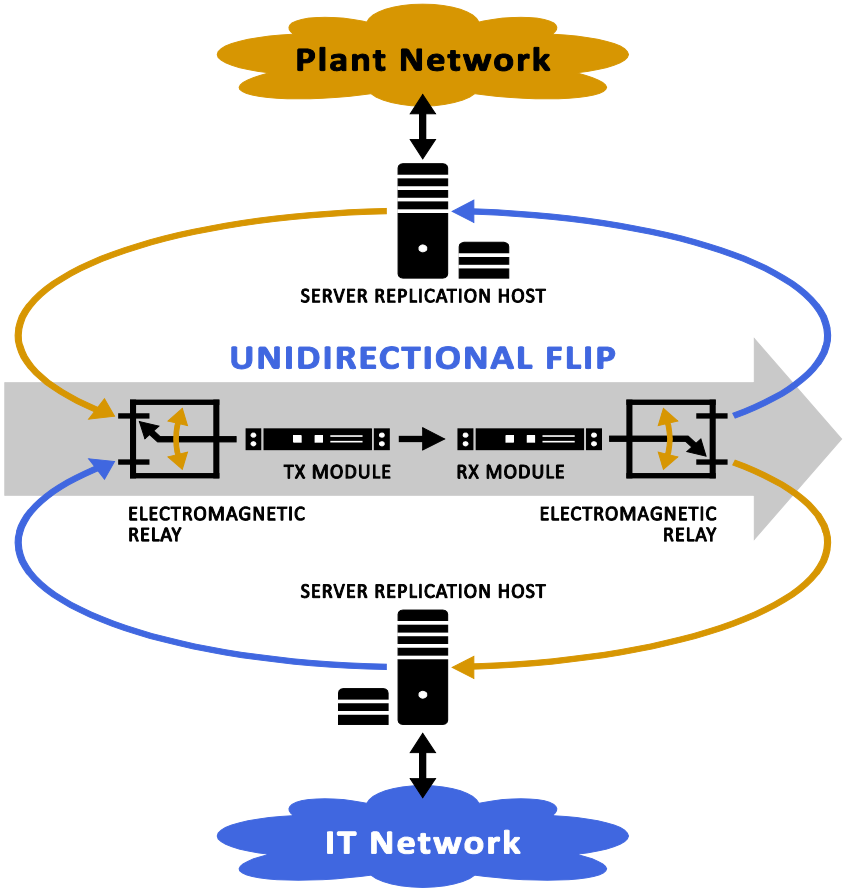

Figure 9-6: A Reversible Unidirectional Gateway

In this example, an air-gapped controller triggers periodic reversals of the gateway. In each orientation, the gateway replicates servers and emulates devices. The gateway may both

replicate a historian database from a plant network to a corporate network and replicate the security-update and anti-virus server databases from the corporate network back into the plant network.

When continuous inputs from an external source are required (for example, when a generating dispatch center must provide second-by-second control of an electric generator to balance generating capacity against power grid load conditions), unidirectional gateways may be positioned to permit data to flow continuously into more-trusted networks. In this case, the gateways replicate servers and emulate devices into more-trusted networks rather than out of such networks. When information, especially the control information, is permitted into more-trusted networks, it is essential to provide layers of defense-in-depth inspection and validation of inbound instruction streams to ensure the reliability of the physical process, as well as to protect both equipment and worker safety.

Unidirectional gateways may have information filters built into the replication software. As the server replication software extracts information from servers for replication, that information can be filtered according to sophisticated policies. In the generating dispatch center example above, the replicated server may be an *inter-control center communications protocol* (ICCP) server, and the filter may be configured to permit only select register numbers and values to enter the protected generating network.

9.2.7 NETWORK ACCESS CONTROL

Network access control (NAC) grants or restricts logical access to the communication network, combining network control and network security control. An example is a user connecting an Ethernet cable to a switch or router. The cable establishes the physical connection, and the switch or router assesses whether the end device will be granted logical access to the communication protocols. If access is not granted, the physical link will remain "dead" for network communication and the connected end device will remain locked out of the network.

A well-known mechanism for granting access is IEEE 802.1X[1]. Devices are either permitted or denied access to the network based on per-device credentials such as identity certificates as well as user names and password. IEEE 802.1X lets network operators maintain strong control over the set of devices that can communicate in the network.

Network access control based on the IEEE 802.1X authentication method is available in many modern Ethernet switches and wireless LAN access points. In Ethernet switches, 802.1X is usually performed on a per-port basis. The WLAN access point replaces the physical network port as the point of authentication in wireless LAN.

A device requesting access to the network must implement a *supplicant*. A switch, router or wireless access point implements the authenticating counterpart, the *authenticator*. In some cases, network equipment may implement both the authenticator and the supplicant feature.

[1] See [IEEE-802]

A supplicant requests access from an authenticator that forwards the access request to an authentication server for review. After authentication, the switch or the wireless access point enables the port or the wireless connection for traffic other than just the IEEE 802.1X authentication frames. The authentication server can be integrated into the device itself.

Authentication servers can also be made available as a centralized resource to the whole network, implemented through a *Remote authentication dial-in user service* (RADIUS) server. Access credentials such as user names and passwords can then be administrated centrally and accessed by all network devices are acting as authenticators. Also, user-specific configuration information can be rolled out via RADIUS and assigned via IEEE 802.1X, such as membership to specific VLAN.

9.2.8 USING SECURITY GATEWAYS TO PROTECT LEGACY ENDPOINTS, COMMUNICATION AND CONNECTIVITY

'Industrial Internet Reference Architecture'[1] suggests use of gateways to integrate multiple connectivity technologies, for example protecting legacy endpoints and communication links while enabling interoperability of brownfield and greenfield deployments in IIoT systems with a secured gateway acting as a mediator, as shown in Figure 9-7. A similar approach should be used to integrate legacy endpoints with limited support for security functions into modern IIoT systems.

An IIoT gateway enacts proxies to one or more legacy endpoints and transforms the legacy protocol expected by the legacy endpoint to the modern interoperability protocols used by new endpoints. It prevents exposure of legacy endpoint attack surfaces to networks. It can also mediate between IIoT systems with support for both per-user authentication and role-based authorizations, and legacy systems with no such support. In addition, the IIoT gateway can normalize the information into a few selected interoperability protocols so that applications can interoperate without having to support all of them. This can reduce the attack surface.

The link between the IIoT gateway and each of the legacy endpoints may also be protected using technologies transparent to the legacy protocols. For example, in LAN, VLAN technology may be used to separate devices on a legacy network segment, when those devices need to communicate to the IIoT gateway and have no need to communicate with each other. In WAN, vulnerable legacy communications protocols may be tunneled transparently through VPN that are implemented in firewalls deployed at IIoT/WAN network boundaries.

[1] See [IIC-IIRA2016]

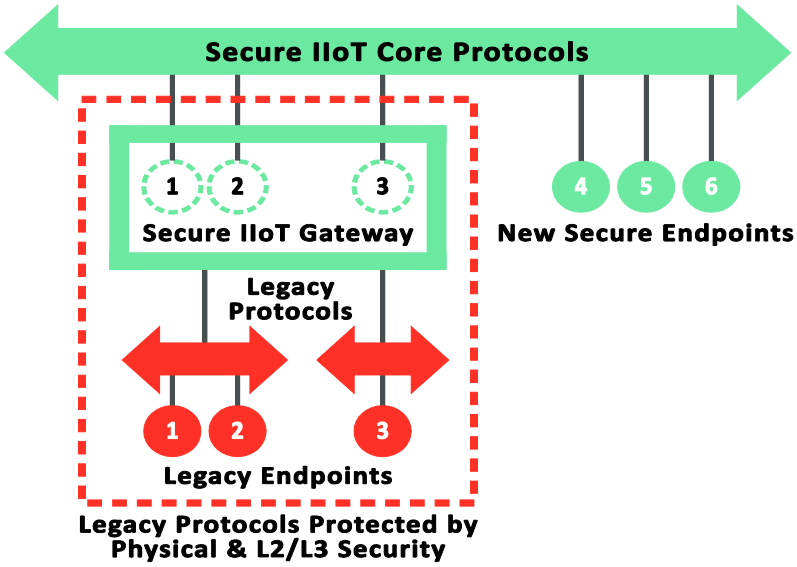

Figure 9-7: Protecting Legacy Endpoints and Communication Links Using Gateways

9.3 Security Model and Policies for Protecting Communication

Various system components in IIoT systems may be owned and deployed by one entity, but managed, maintained, or used by other entities. For example, a maintenance company must have access to the control and instrumentation and monitoring channels of the jet engine to do predictive maintenance. In some situations, this access takes place when the equipment is in operation, and the operation must not be affected by such access. Once proper security policies are in place, protection of transactions across software and hardware boundaries can be enforced using technologies such as SAML, OAuth, OpenID.

Security policies are often captured formally or semi-formally using security models. A security model specifies allowed and prohibited relationships between subjects and objects and therefore can define security policies more concretely. For example, the security model for Linux file system specifies what subjects (i.e. processes) can perform what operations (e.g. read, write, execute) on what objects (e.g. files). Similar security models exist for IIoT communications and connectivity protocols such as DDS.

Communication & connectivity security policies must be derived from comprehensive risk analysis. These policies specify how to filter and route traffic, how to protect exchanged data and metadata and what access control rules should be used. Communication and connectivity policies can be defined with a policy definition language (i.e., XML or XACML) and enforced with a combination of communication middleware and network administration rules. These policies should be explicitly tested for consistency and evaluated for comprehensiveness. Security testing should be conducted using test cases derived from the defined policies.

Security policies should be specified and enforced with fine granularity. The right policy must be defined in a detailed, consistent and comprehensive manner, and the defined policy must be enforced with security tests to provide evidence for such enforcement.

10 SECURITY MONITORING AND ANALYSIS

Security monitoring aggregates and stores a variety of types of data from running Industrial Internet of Things systems, enabling analysis into past compromises, current security events and the prediction of future risks. Security analytic tools provide useful feedback to the organization via parameters suitable for high-level dashboard display.

Monitoring parameters are most valuable when they relate directly to an organization's security concerns and are prioritized by stakeholders. They should represent well-defined actionable conditions understood by those who must take action. As an example, a parameter could report the fraction of meters that responded successfully to their most recent firmware validation request, and another could indicate the fraction of end-user sites whose power flows have been disabled by remote control by the utility.

Monitoring is related to the model of attack incidents and security and privacy policies. An incident model consisting of three phases includes a potential attacker performing reconnaissance to understand the system, an attack in progress, and recovery from an attack. Data collection considerations related to performance, scale and privacy should be considered, as well as the types of analysis possible and the various actions possible to implement additional security controls.

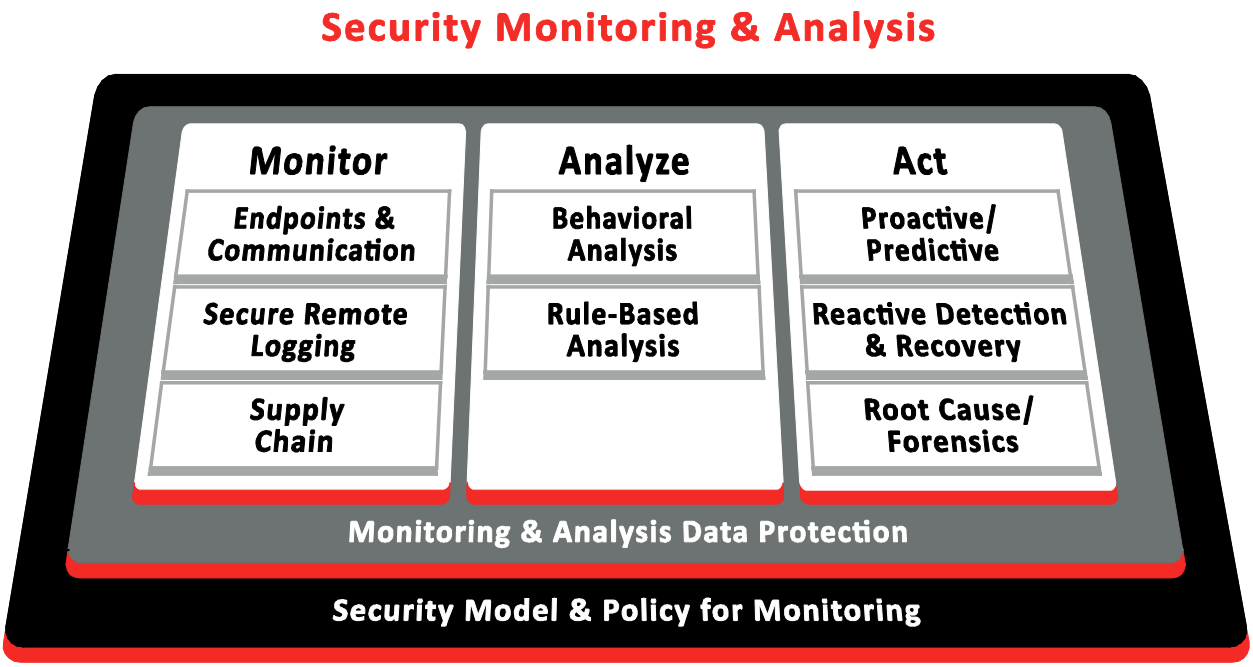

Figure 10-1: Functional Breakdown for Security Monitoring and Analysis

Monitored data can be collected from endpoints as well as the network and should be stored securely. Data may also be collected from devices and their components at stages in the supply chain process as IIoT components are manufactured to ensure that they themselves are secure as expected. Different types of analysis may be performed to provide indications of

vulnerabilities and attacks. This analysis allows actions to proactively implement security controls to reduce the potential for attack, actions to react to attacks in progress, to enable forensic analysis of previous attacks and to learn and to predict vulnerabilities that might be exploited in the future.

Greenfield systems can be designed with monitoring in mind, however it may be more difficult with brownfield implementations where endpoints may not support monitoring functionality.

Monitoring and analysis also applies to the supply chain, a series of processes that may span organizations in producing a component of an IIoT system. If an attack occurs in the supply chain it may have a major impact on an IIoT system, making integrity validation important.

The monitoring and analytics system must also be secured. It must prevent leaks of confidential and private information as well as leaks of data about the system security that could enable subsequent attacks. It must also prevent attackers from injecting false data to the security monitoring and analytics system that could result in a self-inflicted denial of service attack.

10.1 Incident Prevention, Detection, Analysis and Response

Security analytics are most valuable when they produce actionable conclusions that can be incorporated into automated incident response plans. Automatic responses should usually be limited in their effect. For example, if monitoring tools indicate that an intruder is on the network, only that segment of the network should be isolated and shut down, so the intrusion can be investigated before the entire network is shutdown causing a denial of service to all.

10.1.1 Prior To An Incident

Before an attack, there may be indications that it is likely to occur. An attacker may leave tracks as they perform reconnaissance to map and understand a system and its vulnerabilities. If these tracks are detected this can aid taking actions to understand and mitigate the attack. IIoT systems should relay potential indicators of security incidents promptly to analysis systems.

An *incident response plan* with roles and responsibilities must be in place prior to an incident and tested and updated on specified periods or as needed. During an attack, the following actions may be taken based on monitoring and analysis information and the incident response plan:

- Security incident events can be detected on the network and used to raise alerts after analysis suggests the likelihood of an attack.
- Security policies may be updated on systems reachable from suspect endpoints to enhance their defenses before the attack propagates.
- Appropriate personnel are notified, and dashboards, monitors and reports are updated.

10.1.2 During An Incident

During an incident, accurate data on what changes are occurring in the system is needed:

- Security policies on systems reachable from or affected by potentially compromised devices may be updated to provide elevated levels of defense during a security incident.

- This reactive security response may include modifying security control configurations, blocking services, turning off services and reverting changes.
- Prompt and enhanced forensic recording and secure logging can speed incident investigations and root cause analysis, and support future updates of analytics and operational processes.
- Appropriate personnel are notified, and dashboards, monitors and reports are updated.
- Policies and procedures defined in the incident response plan need to be followed.

10.1.3 After An Incident

After an incident, normal operation of the system should be restored as soon as is safe and practical. A decay algorithm can slowly reduce the risk rating to bring the system back to a normal, steady state, resetting policy along the way.

A lessons-learned exercise after an incident can enable the update of the incident response plan so it can be more robust and effective for future incidents. In addition, the reporting dashboard for alerts should be reviewed to ensure future events are detected.

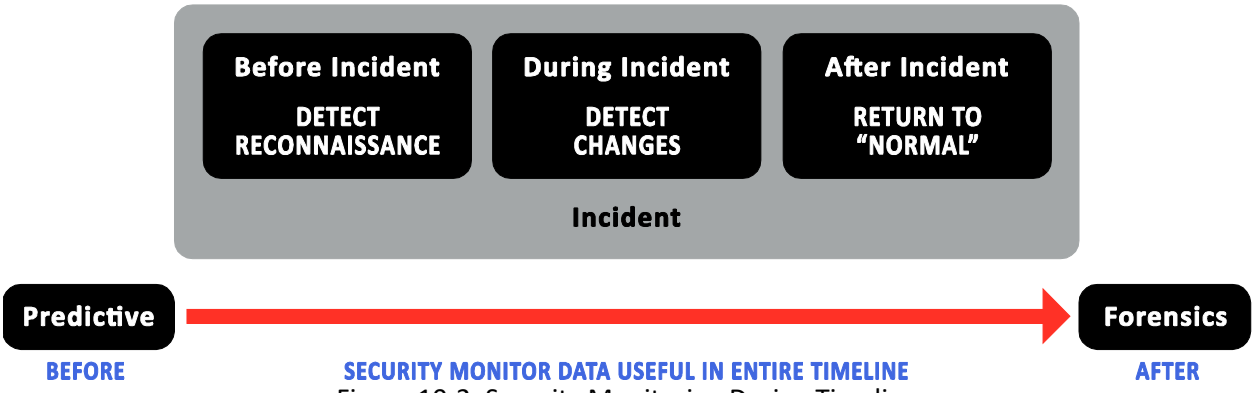

Figure 10-2: Security Monitoring During Timeline

10.2 Security Monitoring and Analytics

10.2.1 Purposes and Kinds of Security Monitoring

Monitoring and analysis systems support three purposes.

Forensic monitoring and analysis systems gather and store security data and make it available to security investigators seeking to determine which equipment and data was affected by a compromise and the specific sequence of events leading up to it. Recorded network traffic can help to identify where an attack came from and to which machines it may have spread.

Current monitoring and analysis systems gather and analyze data to identify attacks in progress, security policy violations in progress and currently compromised devices. Failed authentication requests and tamper sensor alerts can indicate an attack in progress.

Predictive monitoring and analysis systems gather and analyze data identify trends suggesting that new attacks are about to occur, or that IIoT systems have changed in ways that might make them more susceptible to future attacks. Examples of data that may suggest new attacks are an increase in the frequency of audit function shutdowns, system configuration changes and unexpected user account creation. These may suggest a system has become vulnerable to attacks and that proper policies and procedures are not being followed.

10.2.2 TYPES OF SECURITY ANALYTICS SYSTEMS

Security analytics traditionally tend to be either behavioral or rules-based. These are also known as anomaly-based or signature-based systems, respectively.

Behavioral/anomaly-based systems first learn the characteristics of "normal" operation. Once complete, the system generates alerts when tracked characteristics deviate significantly from that learned normal operation. *Anomaly-based network intrusion detection systems* and *file system monitoring systems* are examples of behavioral analytics. Safety-critical and reliability-critical systems are good candidates for behavioral analysis because they change slowly.

Rule/signature-based analytics rely on a library of rules or signatures to identify suspicious behavior. When a set of security values is received that matches a rule, an alert is raised.

Both kinds of analytics may result in false negatives (when the analytic engine fails to recognize an attack), or false positives (when the engine incorrectly diagnoses legitimate activity as an attack). There is a trade-off between false-negative and false-positive errors; the lower the threshold for suspicious behavior, the smaller the risk of false negative missed alarms, but the greater the number of false positive alarms.

Analysis should use both behavioral and rule-based indicators. Behavioral indicators detect those events that are difficult to define with rules alone. Rule-based indicators detect those events that are clearly never intended to occur and are difficult for the analytics to learn from training.

The rules and signatures must be kept up to date, a possible challenge. Behavioral systems may also require management to correct or modify the training if bad behavior is perceived as good.

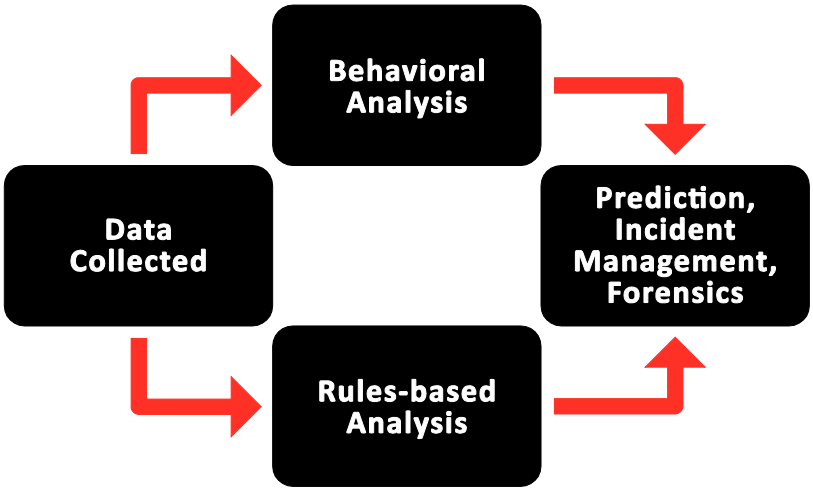

Figure 10-3: Security Monitoring Data Analysis Variants

10.3 Capturing and Storing Data for Analysis

10.3.1 Logging And Event Monitoring

All security monitoring designs must consider the risk that a successful intruder can erase all evidence of their activities. Transmitting the most important security monitoring data to external monitoring systems in a secure and timely manner mitigates this risk. Endpoints must log data based on both local endpoint events and communications events. Logging to a network log system can also mitigate attempts of intruders to interfere with the integrity of log data.

Security monitoring gathers security-related event data, then aggregates, correlates and analyzes it. It should be able to monitor and control the various endpoints and communications in a generic and consistent way. Common APIs help.

There is a distinction between operational monitoring and security monitoring. Operational monitoring concerns itself with such events as ensuring that the cooling tank water level remains at a certain height, the temperature of a sensor doesn't exceed a certain threshold, and the velocity on a conveyer belt remains constant. Security monitoring concerns itself with such events as detecting a successful login from an unexpected endpoint, followed by a blocked connection attempt or an application whitelisting violation, that together indicate a potential attack in progress.

10.3.2 Capturing and Monitoring Security Data

Monitoring data can come from many sources, in particular endpoints and the network. This data should be communicated securely to monitoring and analytics systems.

Greenfield endpoints should be able to report a variety of parameters and should support configuration of which parameters are reported and at which frequency. This configuration and reporting should be done securely. Performance is also important, so the amount of data reported needs to be the minimum needed and may be increased during an incident. Some data may be stored on the endpoint, or transmitted to a secure storage service.

Examples of endpoint data that may be monitored include:

- time and system information, including timestamps, IP addresses, port numbers, other network identifiers, system identifiers, process identifiers and filenames,
- user information describing the authenticated user responsible for causing the event, or which system user the event affects or is relevant to,
- physical process information describing aspects of the physical process the data relates to, such as physical equipment names, sensor types, names of monitored values, or physically-connected device register names or numbers and
- location information describing where the IIoT device was when the data was recorded.

Network monitoring can be achieved using network hardware that uses port mirroring to copy network packets from the network to a monitoring device. This enables network packet traffic to be analyzed for various aspects, such as the protocol types, sources and destinations, timing and other aspects. This can be used to detect attacks at various levels in the protocol stack.

Network and host information that may be monitored includes:

- full network traffic recordings that store every bit in every packet for a period of time,
- host execution activity and audit recordings that store every significant action taken by a CPU, process or software component, such as reading a value from a physical process, controlling some aspect of the process or accessing sensitive information such as personally identifiable information, or a private encryption key,
- network statistics, including connection setup and tear-down events, communications volume statistics for different kinds of data content and communications connections and
- data from security analysis systems that should also be treated as security data and made available to analysis engines for further correlation.

Only the minimum amount of data needed should be collected to avoid the costs and difficulties of storing, transmitting and analyzing large amounts of unnecessary data. Minimizing data collection also reduces the risk of exposing it.

Owners/operators should not collect sensitive end-user data as part of monitoring. Where it cannot be avoided, their own procedures and service level agreements (SLAs) should follow privacy and security regulations, especially when access to data is indirect, such as when a network packet trace includes user data as part of the payload. Secure logs, monitoring storage and audit mechanisms should be used, for example by storing logs remotely.

10.4 Security Data Protection

There are security policy and regulatory challenges for gathering, communicating and storing sensitive data used for monitoring and analysis. These include:

- regulations that prohibit certain kinds of monitoring of employees and other authorized users, or require notifying users or acquiring their permission before monitoring them,

- regulations that prohibit the transmission of personally-identifiable data across geographic boundaries, or the storage or analysis of such data in some regions,
- sensitive data may need to be protected at rest or
- sensitive data may need to be protected from modification, such as by writing it to a write-only, write-once medium and by providing a mechanism to compare on-device log data with centrally reported data.

10.5 Special Considerations for Monitoring

In addition to the general aspects to monitoring, special considerations apply to brownfield systems, supply chain systems, and the relationship to security and privacy policies. There may be limits on the data that can be collected from legacy brownfield endpoints that do not support monitoring directly. This might be addressed using a front-end system when feasible. A supply chain is a special case for monitoring, since it requires monitoring the stages in producing IIoT components to ensure their integrity. Finally, data monitoring should be compliant with privacy and security policies.

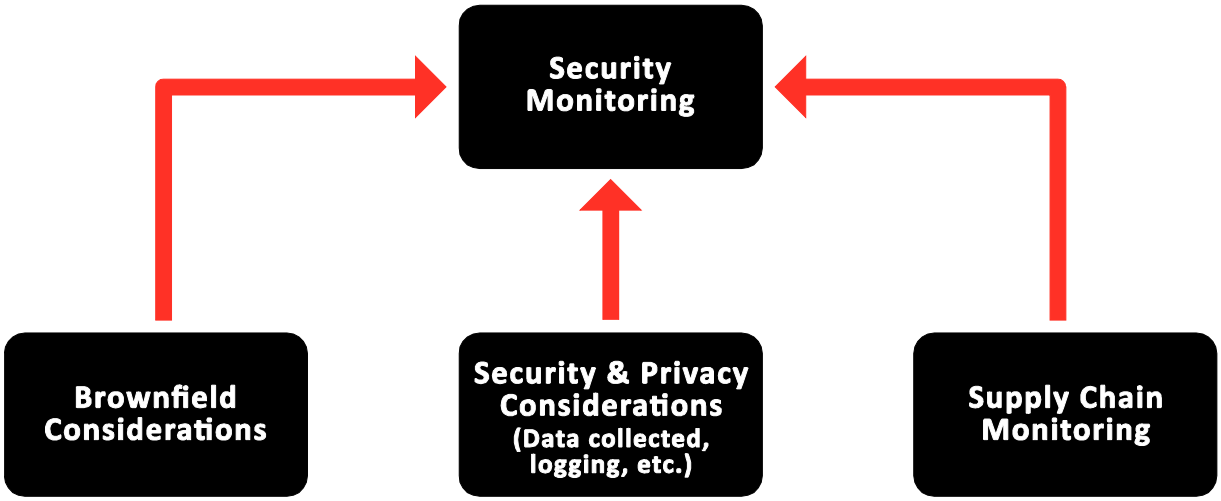

Figure 10-4: Security Monitoring Special Considerations

10.5.1 Security Model and Policy

Security monitoring is effective when there is a model of expected state and interactions allowing deviations from that model to be detected. Examples are the expected protocol interactions on the network, including their network destinations. The monitored data should be consistent with expected network and endpoint behavior, including security policies.

10.5.2 Greenfield versus Brownfield Considerations

Legacy industrial systems may have limited logging and reporting capabilities, and they cannot be upgraded to provide modern capabilities because of the cost of re-certification. Detailed logging at gateways to legacy systems and passive network monitoring systems for legacy communications can compensate. Passive network monitoring keeps track of normal network patterns, and uses analytics to find signs of compromise to the network. A network intrusion

detection system is an example of passive network monitoring device that can be deployed on a brownfield network to enhance monitoring without requiring changes to devices on the network.

Passive network monitoring systems create a forensic log of all communications on networks, may calculate communications connectivity and data volume summaries and may use analytics on security events. If a legacy device has no ability to record when they receive commands to modify sensitive control registers, a passive network monitor can log aspects of those messages on behalf of the legacy system.

Security monitoring and analytics of a new system may be more effective since it can be built into the system from the beginning. The techniques described here are limited to what can be observed "on the wire," for example, not having access to internal state.

10.5.3 Supply Chain Integrity Monitoring

The supply chain is the sequence of processes involved in the production of components, software and parts that together make up a system, spanning many organizations, including suppliers, vendors and multiple tiers of outsourcing. It is a complex, globally distributed system of interconnected networks that is logically long, with geographically diverse routes. It includes organizations, people, processes, products, and services and the infrastructure supporting the system development lifecycle, including research and development, design, manufacturing, acquisition, delivery, integration, operations and disposal of an organization's products and services. Trustworthiness should be assessed across all of these in an IIoT system.

Devices and systems have various phases in their lifecycle. They are:

- device (e.g., meter) module manufacturing/production (hardware/software),
- device module system integration,
- device initialization/configuration setting by owner (provisioning),
- deployment of devices by entity/third-party in field (activation),
- periodic field updates of price and service info,
- firmware upgrade and maintenance,
- remote deactivation/reactivation (temporary) and
- termination (end of life).

In order to detect and prevent unauthorized changes to endpoints being produced in the supply chain, the hardware, software and hardware sub-components need to be monitored to ensure their integrity. Unexpected changes should not occur in the process as different actors in the supply chain contribute to the overall product.

Integrity verification may rely on roots of trust, embedded identifiers and digital signatures, as well as monitoring and verification throughout the build process. Authentic parts, complete with integrity verification capabilities, help ensure there can be trust in the integrity of the chain of custody during the supply chain process. Incorporating these requirements enhances the integrity of an organization's supply chain process and mitigates supply chain risks.

During the integration phase, when new modules are added, integrity metrics should be updated in a cryptographically secure way. All the configuration settings, initialization parameters and other user settings should be similarly updated. The configuration settings information should be encrypted with the unique key of the device in protected storage of the secure hardware that has countermeasures in place to prevent from attacks and tampering.

Without appropriate controls, monitoring, and attestation capabilities, there are many stages in the supply chain where endpoint integrity cannot be assured and the endpoint should not be trusted. Creating attestation mechanisms for assessing the integrity of a device as it moves through the chain of custody stages enables trustworthiness measurements that would otherwise be absent.

Device manufacturers must ensure integrity throughout the supply chain and lifecycle, and provide mechanisms to measure its integrity. Service providers must then ensure that overall systems can be attested to all the way down to the manufacturers integrity measurements. Equipment owner/operators should measure all the endpoints and services in their environment to attest to their integrity throughout their lifecycle. Only then can a system be trustworthy.

11 SECURITY CONFIGURATION AND MANAGEMENT

Changes to the environment and the discovery of new vulnerabilities and threats will require updates to policy, firmware and software, so the security features of an Industrial Internet of Things system must be configurable and manageable, not statically defined. In addition, the deployed versions must be carefully controlled, configured and managed.

Periodic security compliance reports are often mandated and certainly advisable. Network and endpoint configurations should be analyzed periodically to report deviations from all relevant policies and to summarize compliance postures.

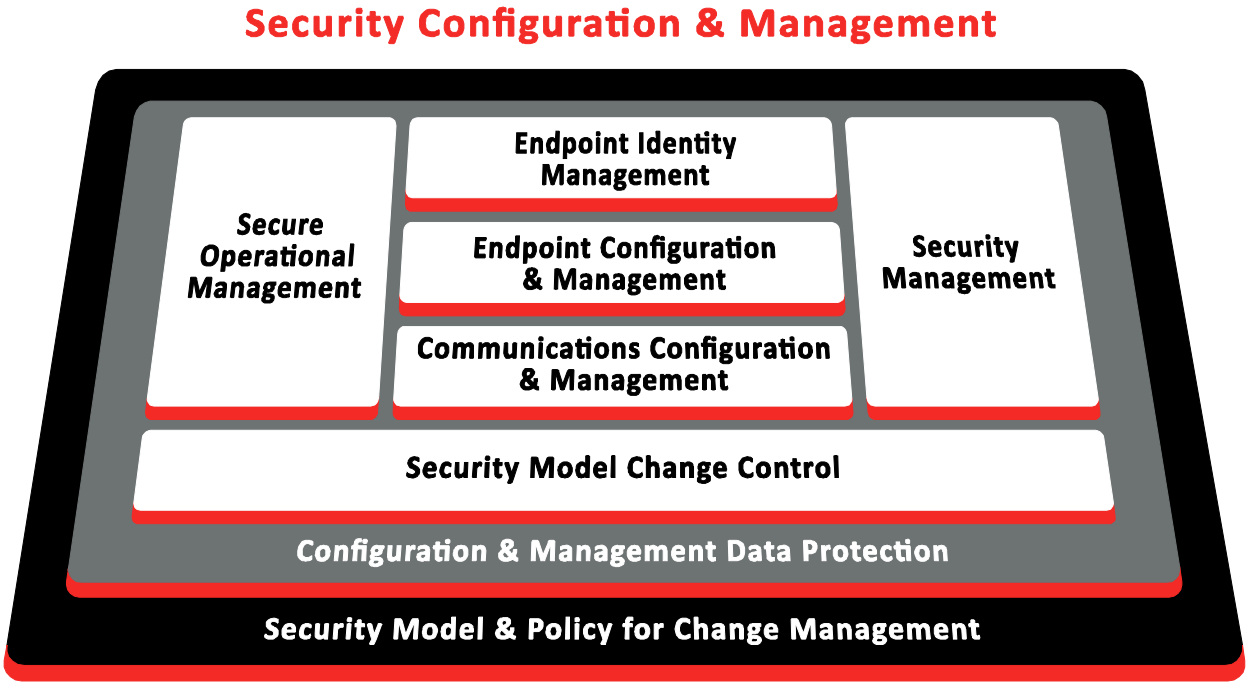

Figure 11-1: Functional Breakdown for Security Configuration and Management

Security management must determine the security objectives of the system to be managed. These security objectives should identify the techniques to be used to ensure the confidentiality of information, the integrity of the endpoint and communications, and the availability of the system functions required for management.

11.1 SECURE OPERATIONAL MANAGEMENT VS. SECURITY MANAGEMENT

IIoT system management has two related concerns.

Operational management is the configuration of the operational functionality of the system and its endpoints, including provisioning, operating system settings, physical and logical network settings, and the application configurations for the operational process.

Security management is the management of the security controls on an endpoint, including the addition and removal of security controls, the setting of security policy, and enablement of extracting security events and logs.

Operational management should be separated from security management so that security controls processes can evolve independently.

Secure operational management involves protecting the operational management process and functions to ensure the integrity and confidentiality of changes made to operational elements of the system including endpoints, communications, monitoring, and management systems.

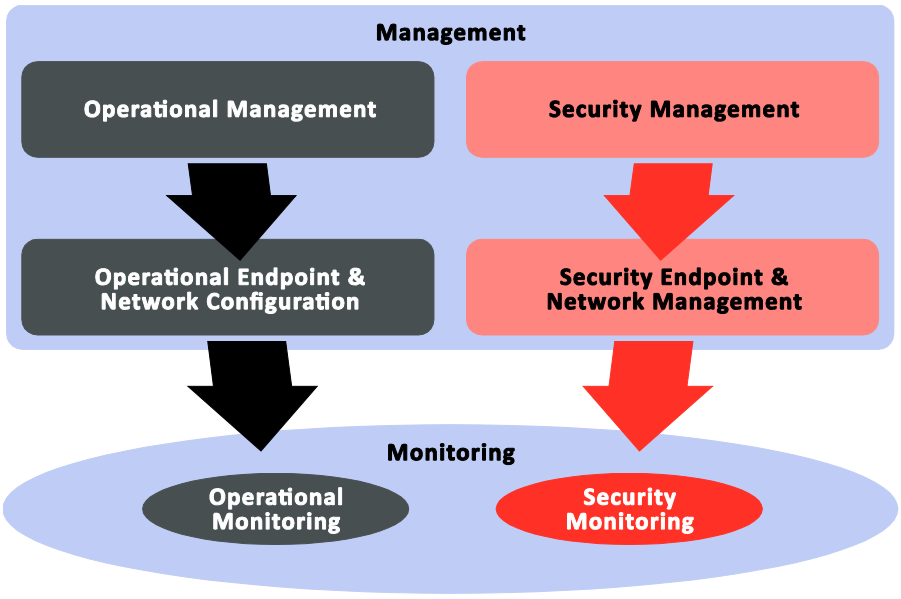

Figure 11-2: Secure Operational Management

Both implement policy to configure the settings on the endpoints, and a misconfiguration on either the system or the security may result in vulnerability. The sensitive nature of the endpoints, the applications and the data requires especially diligent care in the separation of concerns between these areas, though the line between them is often blurred.

Operational management must interact with operational monitoring. There should also be a separation between operational events and security events. Security events from the endpoints and communications are used by security monitoring to evaluate security and identify gaps that must be remediated. The operational management and monitoring controls are specific to the system's operational process, as opposed to the security management and monitoring, which can be the same across different operational processes. For example, the setting of credentials, the definition of network data channel rules and the identification of the destinations for security event data are all required across various operational processes, though the actual settings may vary from endpoint to endpoint.

A common security API across all the endpoints making up the operational process isolates the security process from the operational process, independent of the make, model and

manufacturer of the endpoint. To realize this, endpoints require three security-related APIs as shown in Table 11-1.

API Call	Description
Receive Policy	*Receiving policy* exposes an API to receive system configuration and security management policy from a management component. The policy is parsed and the sections delivered to the associated controls. This enables the remote management capability.
Gather Logs and Communicate Events	Communicating events by collecting log data to be offloaded from the endpoint ensures that attacks on the endpoint can be tracked and hinders attackers' ability to hide the evidence of their activity.
Gather Endpoint Properties	*Gathering endpoint properties*, including hardware capabilities, software on the endpoint (including OS), and application settings, from the endpoint, ideally via trusted introspection mechanism.

Table 11-1: APIs for Interoperable Endpoint Security

An interoperability standard defining these common APIs explicitly would unify the implementation of a significant portion of the management and monitoring infrastructure. The NIST SCAP standards for defining interoperable content automation for vulnerability, measurement and policy compliance, and IEC 62351, Part 7[1] for network data and security management for the power industry go some way towards this goal, but there is no published standard in existence to date, so each management and monitoring implementation is different from the others.

11.2 Security Communications Channels

Communication channels include a data channel and a control channel with management as a sub-channel of the control channel. The control channel enforces policy on the data channel.

The management channel carries several types of messages requiring independent handling. For example, security message flows containing policy flowing to endpoints should be separated from security event flows flowing back to an aggregation point to enforce the separation of concerns between policy management and event monitoring.

The security channel may be divided into a security configuration channel and security monitoring channel. The security configuration channel contains the policy definition.

The hierarchical channels are shown in Figure 11-3 below.

[1] See [NIST-SCAP] and [IEC-62351-7]

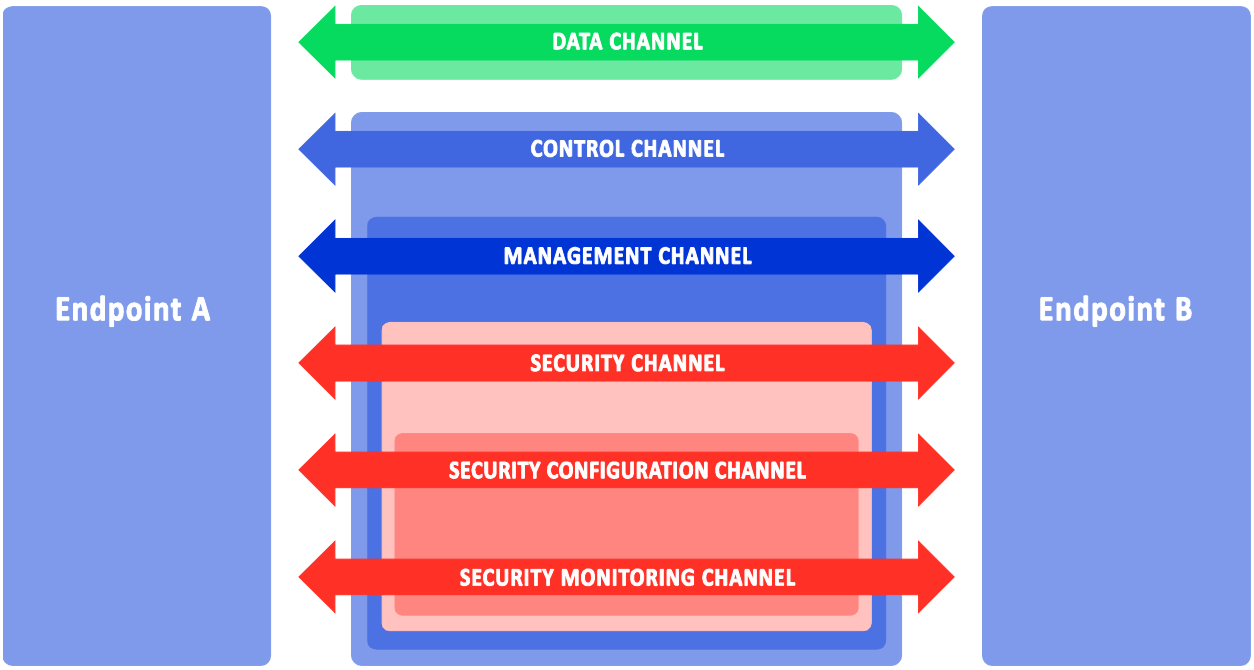

Figure 11-3: Hierarchical Communications Channels

11.3 Secure Operational Management

Operational management configures and controls the IIoT system and its components to implement the organization's business process. Trustworthiness of the system depends on trust in all the system elements, as well as assurance that all of the system elements are working together correctly. Security must be managed across the entire system, and must not disrupt the operational processes or diminish the safety or reliability of the system.

There may be safety considerations in operational management. In assessing risk in the operational system, the criticality of each endpoint must be assessed. The security management system must have higher criticality than the most critical endpoint it manages, and so requires compliance with all safety policies required by that level of endpoint. Different standards define different mechanisms for determining the criticality of systems.[1]

There may be safety implications that traverse operational management systems. Multiple operational management systems each manage a set of devices at a specific safety level, so that devices that do not wish to inherit safety regulations of OT devices are naturally segregated. Security management does not suffer from this same inheritance dependency; therefore, it is desirable to isolate the security functionality from the safety functionality.

11.4 Security Management

The security model for a system is based on a number of sources ranging from regulatory policy to industry standards, organizational directives and personal experience. The security policy must

[1] See [NERC-CIP-002]

be applied to a security model that can be implemented within the organization and periodic reviews should be scheduled to update it if necessary.

11.4.1 Security Policy Management

Security policy is an overarching term; there are actually three types of policy. *Machine policy* comprises a digital document that contains the settings for the technical security controls on an endpoint. *Organizational policy* documents the expected behaviors, both technical and non-technical, for an environment (for example, firewalls do not allow incoming event communications, or every room must have a fire extinguisher). *Regulatory policy* compels behavior at a high level (state, country, or global) by distinguishing good behavior from bad.[1]

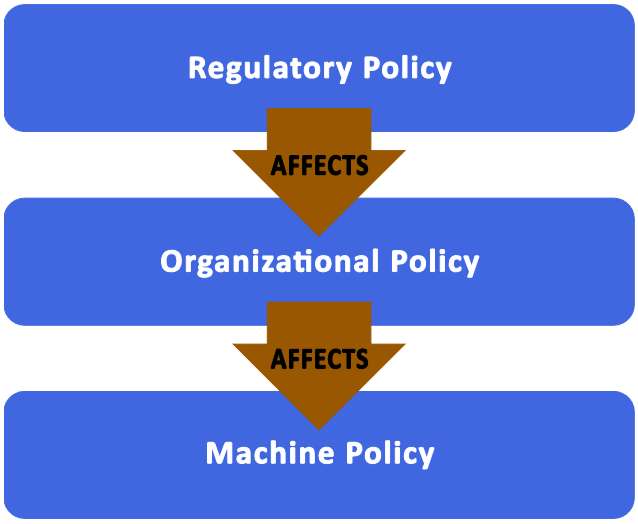

Figure 11-4: Policy Relationship

Security policy describes the expected behavior of the security elements of a system; security monitoring describes what is actually happening in the environment. *Security management* is the iterative process that configures and updates the system to maintain the same level of security.

A security management platform provides the ability to define policy for each of the endpoints' security controls, communications streams and software and firmware updates. The platform provides an infrastructure for event monitoring and raising alarms at appropriate times. Analytics provides situational and contextual awareness and the results update machine policy settings. Automation pushes policies to endpoints and collects and analyzes events coming from them.

Security management adjusts security capabilities to address changes in conditions. The user interface and workflow should be simple enough for a person to define, update and monitor security status accordingly. If security management is difficult to use, people will have difficulty applying security effectively, and security incidents will be more likely.

[1] For example, see [NERC-CIP] and [EU-2016/679]

To limit the risks of cross-contamination between operational and security concerns, different teams, with different roles and responsibilities, should each have the minimum level of access possible for any particular operation.

The security management platform should balance security and the other key system characteristics.

11.4.2 Policy Authoring and Definition

Security policy is assigned to an endpoint or a group of them. The policy should be composite in nature. Creating *baseline policies* simplifies applying them to many endpoints in complex IIoT systems. The baseline policy is then adjusted for individual endpoints or groups of endpoints, eliminating the need to rebuild the entire policy each time. Providing a consistent policy format, and enabling the endpoints to interpret the policy eases identifying security gaps.

It must be possible for a person to understand how the security is expected to behave, based on regulatory or organizational policy, and translate that into machine policy settings. There are at least two places where security must be simplified for human understanding: policy definition and the results of the event analysis. *Policy definition* begins with a person defining the desired behaviors in the IIoT environment. These are then translated into security settings that are stored in the machine policy sent to the endpoint. *Event analysis* begins with security events being sent from the endpoint to an adequately secure location for analysis. That may trigger alarms and generate notifications in the form of dashboards, UI alerts, email notifications and reports.

A person must be able to initially define the organizational security policy in terms of machine policies. Applying appropriate updates to the security policy based on security event analysis creates a feedback loop by which the security can be maintained (or even increased) over time. It may be possible to automate this feedback loop.

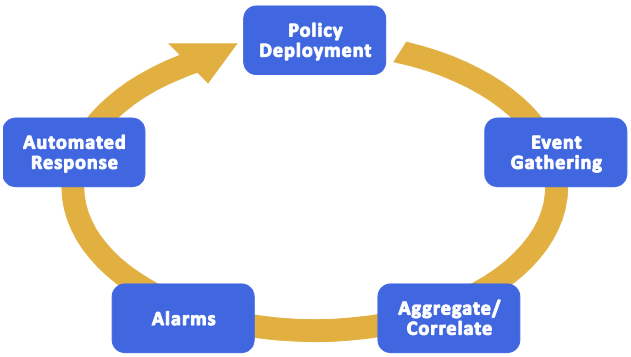

Figure 11-5: IIoT Management and Monitoring Feedback Loop

The feedback loop in Figure 11-5 begins with pushing machine policy to the endpoints. The endpoints gather events based on the machine policy settings, especially violations to the security policy, and communicate the events out for security analytics. The security event data is correlated, alarms triggered when security thresholds are exceeded, and automated responses executed to mitigate the security events. The automated responses may be as simple as setting an alert on a dashboard or sending an email, or as complex as sending out new machine policy

updates to reconfigure affected systems. The feedback loop is analogous to the Monitor (Event Gathering), Analyze (Aggregate/Correlate and Alarms), and Act (Automated Responses and Policy Deployment) as described in chapter 10.

11.4.3 Policy Assignment and Delivery

Within the management process, the security policy must be defined for each endpoint. Having a coarse-grained mechanism to define this policy across a number of endpoints in an efficient manner is desirable. The management process must be scalable, and optimized for a human to be part of the process.

Appropriate policy settings should be exposed to the human so she can correctly configure the controls on the endpoint. Reuse and automation is needed to simplify policy management.

Policy should be pushed to a number of endpoints at once. A structured policy with policy sub-elements enables reuse of the elements across a number of different policies. A policy library can be built up that combines the various policy sub-elements in different combinations without redefining policy for a minor change. Default policy sub-elements ease defining policy by managing divergence from the default, rather than redefining the entire policy.

Endpoint security policy should be assignable to an endpoint or a group of endpoints. This allows the endpoint to be managed at an atomic level, or with other endpoints that share common functionality, without needing to create a policy for each individually.

An automated mechanism to deliver the machine policy to the endpoints is strongly advised. Tracking the policy and sub-elements in place on each endpoint allows for more oversight of the policy management process. Automation scales, and minimizes the impact of human error.

11.5 Endpoint Configuration and Management

Once machine policy has been set on an endpoint, the policy settings configured during the policy authoring and delivered to the endpoint must be activated and enforced. This includes parsing the policy and providing each affected security control with the appropriate policy sub-element.

Each control should receive the policy sub-element related to it, be able to understand the configuration and act upon it. The management platform need not have insight into the control's configuration; it only exposes the UI controls and stores the results. Nor do the policy delivery steps need insight into the control's configuration; they only ensure the policy is delivered to the right endpoint with proper integrity and confidentiality. Only the control itself, on the endpoint, needs to understand the configuration settings saved on the management console.

To implement new security controls, only two components need to be built: the policy element that a human configures, and the security control parser that resides on the endpoint and translates the machine policy sub-elements into the appropriate settings.

Once the policy has been applied to the endpoint, any events that result from policy violations should be communicated off the endpoint.

To author and deliver the policy, as well as during the extraction of the events, the metadata about the policy and about the events must be carefully guarded. Access control over these must be strictly enforced or the best security implementations will be rendered vulnerable to compromise. Security metadata from any data monitoring requires policies defining how it is handled and who has ownership and access to it. There may be privacy implications to some security data collected.

11.5.1 Secure Software Patching and Firmware Update

As the amount and complexity of software increases, so does the number of defects, some of which will be exploitable vulnerabilities. Others may cause unpredictable system failures, timing issues, reduction in system performance, reliability or other unknown problems. Once discovered, these defects can often be fixed by patching. If over-the-air updates are implemented, network-related vulnerabilities that affect the integrity of the over-the-air process should be addressed first.

IEC TR 62443-2-3:2015 'Patch Management in the IACS Environment'[1] defines relevant terminology, lays out patching requirements for both asset owners and product suppliers, and defines a schema for patch information exchange. It also provides guidance for qualifying, verifying and deploying software patches in operational systems.

Sometimes it is not possible to update an endpoint. For example, if an endpoint is too important to continued operation to risk any modifications. Some updates may invalidate a certification or compliance with a standard until the requisite safety assessment is rerun.

A wide range of methods provides software and firmware updates to endpoints. Some endpoints require direct physical access to the device to update it (i.e., by attaching a serial cable or a USB drive). Others allow users to download an update from a remote location and install it locally via command line or agent commands. Clearly, automatic upgrades are easier for administrators, more easily validated to ensure the integrity of the update and its provenance, more likely to be applied and easier to verify that they were applied. As a result, they are more efficient and less costly than update approaches requiring physical intervention at each device.

Software and firmware updates add security, safety, reliability or functionality features, especially in brownfield scenarios. Systems with strong safety and availability requirements often use a staging area to test updates prior to updating all the endpoints. Without confidence that they work, software updates will be ignored, as the operational risk is too great.

Secure update of endpoints can be implemented using software or a combination of software and hardware—with hardware features adding additional layers of protection, integrity and trust. Using hardware containers such as an HSM, TPM or other TEE is strongly recommended. Keys used in upgrades can be managed by a third-party certificate authority and updated as needed. The same mechanism used to update firmware or software securely can also be used for updating system configurations and ensuring that the software is from the expected source.

[1] See [IEC-62443-23]

Gateways may simplify the update process. If a gateway sits in front of a number of endpoints from the same vendor, then secured download and validation of software updates from the vendor update repository is possible. The gateway should include enough security functionality to authenticate the update repository server and the source of the software updates, securely download the update via encrypted channel or by downloading an encrypted update and then verifying the integrity of the downloaded update. Then, the gateway may be able to act as the update server for the endpoint behind the gateway, providing the validated update directly to the endpoint, thereby minimizing the attack surface of the update process.

Manufacturers may try to perform automatic software and firmware updates on their customer's devices, but this is risky. If an update fails, the device may be left inoperable or operating in an unknown and unpredictable manner. Worse, an adversary may commandeer the device as a platform for attacks on other devices.

The integrity of the update must always be assured, regardless of the method of retrieval for the update. Digital signatures enable validation of the update file, and provide stronger security than hashes (see section 8.8.2).

11.6 Communications Configuration and Management

As with endpoint security enforcement, there must be security management and control of the network communications. The policy may be applied at the communicating endpoints, or at intermediary communications devices between them. Mitigating controls that enforce the network security policy on intermediary devices may include firewalls and packet filters, routers, intrusion detection system (IDS), intrusion prevention systems (IPS), network access control, and other security controls and devices.

Of specific interest in managing the security of IIoT communications is network access control (see section 9.2.7). NAC is a management control that prevents endpoints from getting onto the network. It relies on information from network security controls that monitor traffic. By integrating security management systems and security monitoring systems, NAC functionality enables detection of unauthorized endpoints on the IIoT network segments and forcibly disconnects them. Gateways, firewalls, routers and active network monitoring and control devices enable forcible disconnection of unauthorized endpoints.

11.7 Identity Management

Identity management includes the processes and policies involved in managing the lifecycle and value, type and optional metadata of attributes in identity known in a particular identity *domain*, which is the environment where an entity can use a set of attributes for identification. Identity management is one of the primary functions in endpoint security (see section 8.5) and is fundamental to authentication and authorization (see section 8.6).

There are standards and recommendations in place today for identity management. The 'Entity Authentication Assurance Framework' (EAAF) in ISO/IEC 29115[1] is an authentication standard describing the life cycle for credentials and authenticating entities. The NIST 800-57, 'Recommendation for Key Management'[2] applies similar approaches to the management of credentials and identity material. Also, the 'Functional Model Representation of the Identity Ecosystem'[3] is a model for identity solutions, including the various components and interactions.

If the credential management process is not correctly implemented and adhered to, then the results of the endpoint authentication may not produce the level of trust desired.

Applying an IIoT perspective to the existing identity management recommendations yields a variant of the lifecycle process. The treatment of identity for a human entity does not differ greatly from existing IT models, so non-person entities are the focus here. The IIoT management life cycle comprises three phases as shown below.

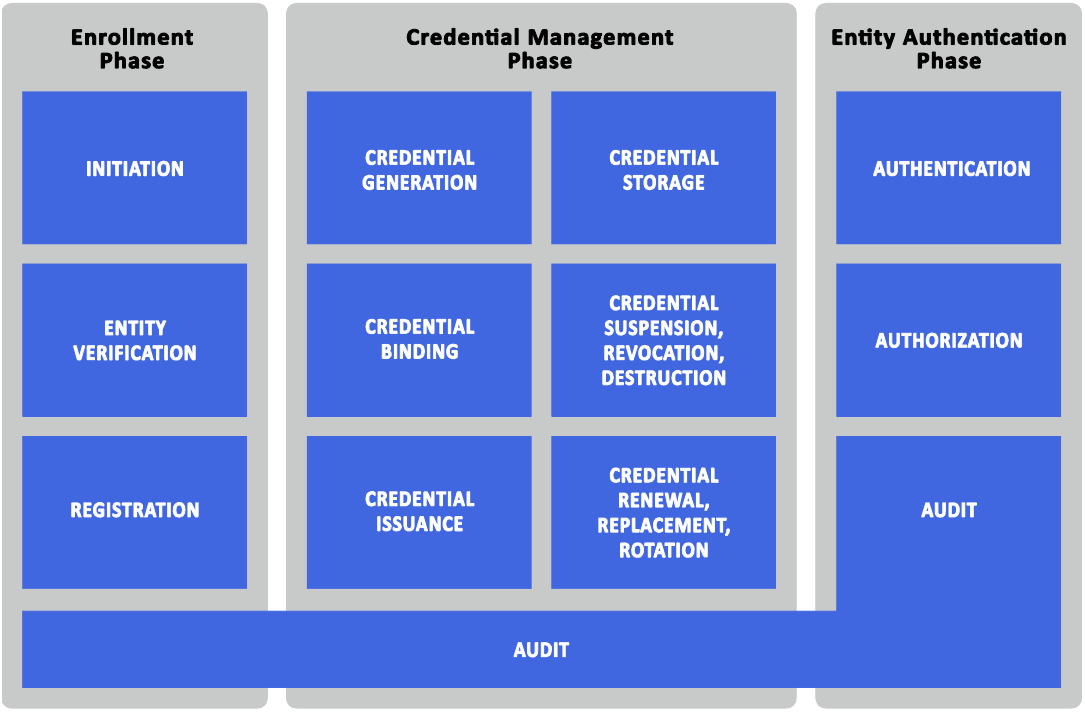

Figure 11-6: IIoT Identity Management Lifecycle

The *enrollment phase* ensures that the appropriate entity is to receive the appropriate identity material. This requires participation from the component builders to establish trust during manufacturing, procurement, delivery and commissioning. The entity may change ownership several times during these steps, so an audit trail tracking the chain of custody should be kept,

[1] See [ISO-29115]
[2] See [NIST-800-57]
[3] See [IDESG-IDEF]

and the identity and integrity of the component should be verified at the end of the enrollment phase prior to the credential management phase. It is difficult to attest to the trust of hardware components purely in software; hardware support is strongly recommended.

The *credential management phase* is where provisioning to the owner/operator organizational environment is implemented. This process includes generation of credentials, or enablement of credential generation capabilities on endpoints. An audit trail tracking the provisioning of the identity material into the entity and the manner of storage and other security implementation properties should be retained. The integrity of relevant entities such as devices and endpoints, and their data-in-motion, and data-at-rest, should be verified to ensure that the credential management phase is correct.

The *entity authentication phase* is where the actual authentication and authorization process occurs during the day-to-day operation of the device and endpoint entities. An audit trail tracking the authentication and authorization attempts and results is retained for an organization-defined period of time based on policy.

11.7.1 Enrollment Phase

The enrollment phase provides the basis for establishing trust in an entity. There must be a mechanism to ensure that it is the correct entity, either manually or automatically, prior to issuing credentials. In order to scale, an automated approach is recommended.

There are three steps to the enrollment phase: initiation, entity verification and registration. *Initiation* declares the desire to bring the entity under management and give it identity and credentials. *Verification* involves proving that the entity is the one for which the identity is to be created and issued. *Registration* (see Figure 11-6) means the entity is ready to have credentials created and delivered, or to have the entity generate the credentials itself. Always validate that the identity that was registered was the one bound to the credential that was generated for the entity.

The enrollment initiation step requires that there be some way to track the entity through the enrollment phase until owner/operator credentials are issued. The entity should have a manufacturer identifier, ideally an endorsement key and certificate, that is assigned and managed by the manufacturer and embedded during the manufacturing process. The manufacturer identifier allows the component builder to validate the entity and establish trust in its authenticity and integrity.

Entity verification extends the enrollment process to assert that the entity to receive the credential is eligible and has the manufacturer identifier for tracking through the enrollment phase. This approach encourages component builders and system builders to expose APIs on the endpoint entity to access the various identifiers and the services to validate them.

During the registration step, the entity, now verified by the component builder or system builder, is present in the organization's asset tracking system and is available for provisioning. To enable the secure enrollment process, the manufacturer should expose a service to associate the device with the new owner, enable the entity to make contact and verify itself upon initial power-on,

submit the manufacturer identifier, and provide a cryptographic binding between manufacturer identifier and the credential. This process should be repeatable for future changes of ownership.

There may be multiple identifiers present on a single endpoint for several reasons. One reason is that the endpoint is managed by multiple entities, as would be the case for predictive maintenance scenarios. Each component builder embeds an identifier, but some endpoints may comprise multiple subcomponents, each with its own identifier, resulting in multiple identifiers associated to the endpoint entity before the owner/operator initiates the enrollment phase to issue his own identifier. The system builder may also add an identifier during system assembly. These identifiers all have a lifecycle independent of the owner/operator identifier.

Throughout the enrollment phase, an audit trail should be created to track the steps as they are executed. The audit data should be retained for a time defined by policy. The audit trail data integrity should be assured and attestable, and treated as confidential.

11.7.2 Credential Management Phase

After the enrollment phase, the credential management phase comprises a number of steps broken down into two categories (see Figure 11-6). The first category comprises the steps required to generate credentials, bind them to an entity, and issue them to the entity to which the credential should be issued. The second category comprises the steps for storing credentials, and end-of-life as well as extending the useful life of the credential.

The first category of steps for credential management brings the entity into the state where the credentials are in place and ready to use. *Credential generation* includes any steps required to create the credential itself, or to enable or direct the entity to create the credential. Then, during *credential binding*, the credential, or the means to create it, is associated to the identity assigned to the entity. Finally, during *credential issuance*, the credential, or the means or directive to create it, is delivered to the entity using a secured and auditable process. The specific process depends on the organizational policy for the environment.

For example, with a HRoT such as TPM in place, a key pair credential should not be generated externally and delivered to the entity. Rather it should be created inside the HRoT and only the public key be reported externally for binding.

The second category of steps for credential management addresses the normal day-to-day usage of the credentials and the edge conditions within its lifecycle. *Credential storage* must be implemented to the level required by the organization policy based on the level of authorization for a particular endpoint. The higher the level of authorization required, the more stringent the credential storage requirements must be. The level of authorization should be enforced in the communications policy so that endpoints that do not have strong enough credential storage are not allowed to connect to the endpoint.

Entities should be categorized into categories of criticality. Each level of criticality should be associated with a level of authentication that defines the level of trust to place in a successful authentication. The level of authentication also defines what controls must be in place to minimize the risk of false attestation, or impersonation. For example, in very low criticality

endpoints, it may be acceptable to authenticate with a plaintext credential using the IP address or MAC address as the identity. But for slightly more critical entities, multifactor authentication may be needed to protect against attacks on stored and transmitted credentials. In the higher and highest criticality entities, authentication should be cryptographically protected and tamper-resistant hardware should be used to store all secrets and credentials at rest and in use.

Credential storage must meet strict criteria on certain endpoints that have a high level of criticality. There may be organizational policy requirements that highly critical entities with strong authentication and credential storage may not trust entities with insufficient authentication and credential protection in place.

At the end of the credential's lifecycle, the credential must be appropriately removed from service. When a credential is identified for suspension, it is temporarily blocked from being used for authentication. This applies to any credential, or generation process, that is suspected of potential compromise in a system. If the compromise is likely for the credential or the generation process, then the credential must be revoked.

Other reasons to revoke a credential in IIoT systems is due to credential expiration or as part of the key rotation process. In either case, a newer credential has replaced the revoked one.

To limit the risk of credential compromise, credentials should be replaced at a specific frequency, as defined in the organization's credential rotation policy. In some cases, it is possible to renew credentials, rather than to replace them, to extend their useful lifespan, if this complies with the credential rotation policy.

All credential management operations must be tracked for audit purposes. The audit data should be retained for a period of time defined by organizational data retention policy. The audit trail data integrity should be assured and attestable, and treated as confidential.

11.7.3 Entity Authentication Phase

Entity authentication establishes the level of trust in the identity of the remote endpoint. Successful authorization based on successful authentication, results in the granting of privileges on resources. Proper authentication and authorization policies must be instituted to control access to resources based on the identity of the remote entity (see section 8.6).

All authentication and authorization operations must be tracked for audit purposes. The audit data should be retained for a period of time defined by organizational data retention policy. The audit trail data integrity should be assured and attestable, and treated as confidential.

There must be accountability across the system by tracking employees and contractors of the OT process. Privacy concerns arise whenever personal information is tracked. An employee identifier may reduce these concerns, so accountability will trump privacy. However, when customer, partner and other data is tracked, care must be taken to protect the PII and other personal sensitive data.

It is possible to have both strong authentication and strong privacy. For example, there exist authentication schemes that limit the disclosure of identity. They provide anonymous cryptographic identity attestation through anonymous credentials and group signatures.[1]

11.8 Security Model Change Control

A number of lifecycle transitions occur over the lifetime of an endpoint. For example, implementing the entity enrollment and credential management phases for an endpoint. Similarly, the security model must change for each endpoint depending on its lifecycle state.

Commissioning provides the endpoint with temporary identity and a policy that locks it down to communicate only with a provisioning server. Ideally, the component builder, the system builder or both should commission the endpoint.

Provisioning replaces the identity in the trust root with the organization's identity, credentials are issued, and new policy is set to put the endpoint into normal use.

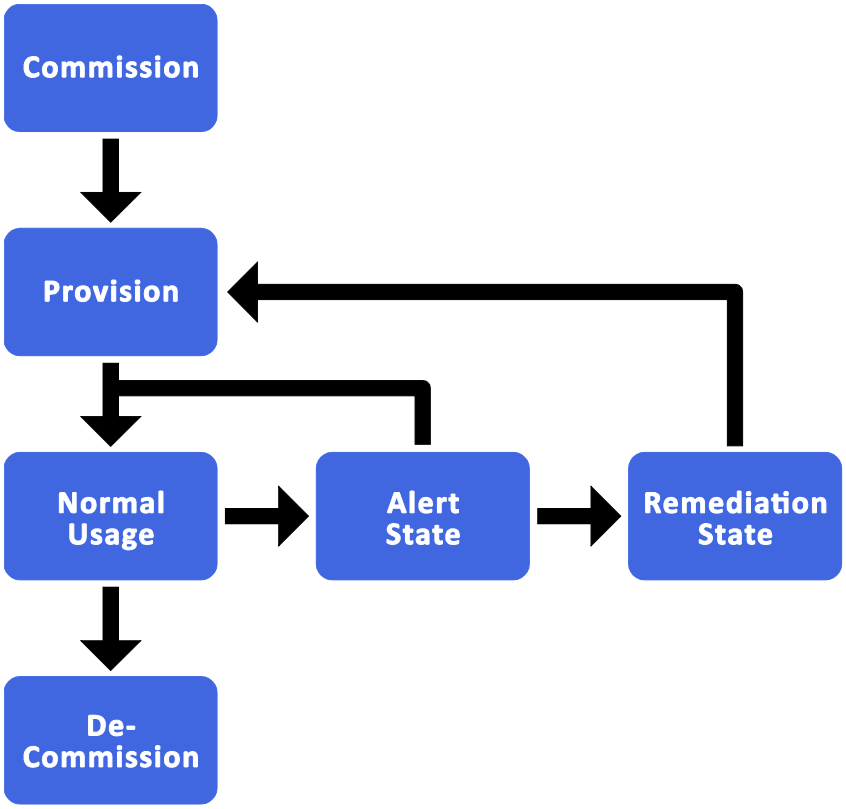

Figure 11-7: Endpoint Security Lifecycle

Endpoint provisioning configures the desired security controls, including deploying the identity material into the roots of trust, setting initial policy settings, and starting the business processes for which the endpoint is designed. In some cases, the endpoint may require the provisioning to occur on a designated network (physical or virtual), to ensure that the onboarding process is

[1] See [ISO-20008] and [ISO-20009]

complete before communications with OT equipment can occur. The endpoint may be reprovisioned both on a schedule (perhaps for key rotation) or based on need.

During *normal usage* the endpoint applies the optimum endpoint security policy. Based on security events, the endpoint may transition to an *alert state* that tightens security controls down to the minimum operational functionality or a later *remediation state* where the endpoint is reset. This may need reprovisioning before normal usage resumes.

Endpoint decommissioning terminates the useful lifecycle of the endpoint and transitions it into an end-of-life state. A decommissioned endpoint may be reused, so it must be able to be recommissioned and reprovisioned for another purpose.

Endpoint availability should be considered throughout the security lifecycle.

11.9 Configuration and Management Data Protection

Security management maintains the consistency of security over time, and must not interfere with operational processes.

Security metadata such as connection status and characteristics (encrypted or authenticated), and the state of security controls on the device should be gathered and shared with operation management systems so that it can be tracked. The security metadata should be sent on a separate communications channel from the operational application data.

In some cases, security management data should be sent on a separate physical network adapter, such as what may be found on a gateway device, or a larger device with multiple physical adapters. In other cases, if the device only has one physical network adapter, security management data should be separated logically (i.e., on its own VLAN).

Security data should conform to the requirements of the specific network. For example, if the network is bandwidth-constrained by operational technology data, then the security metadata may need to be bandwidth-limited through the connection, or may be transmitted in bursts at intervals when network load is lower. Control of the frequency, throughput, volume and duration of metadata updates to the management server is desirable.

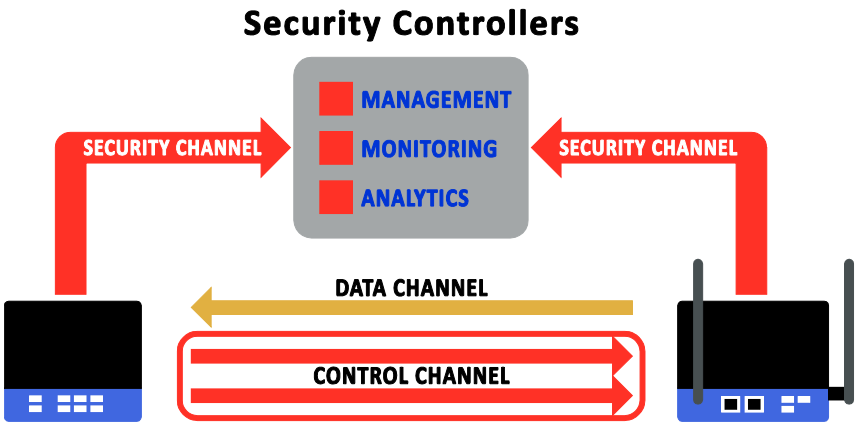

Figure 11-8: Flow of Management Data

Privacy should be factored into the system design to protect the sensitive data, anonymize it, and control the data's retention period and storage location, ensuring that it is properly deleted.

Privacy-sensitive data should be documented to ensure that there is adequate awareness of it. It should be managed based on policies governing access rights and consent/revocation, and sharing with third parties.

Careful management over the ownership of data is required to keep the security data safe from unintended modification. The access control must be enforced on the endpoint, such as in the configuration on a device or in the database of the management server, and in the communications between endpoints.

11.10 Security Model & Policy for Change Management

Changes to regulatory policy, industry standards and new directives should trigger review of the security model. Any update affects the organization policy hierarchy. For example, when regulatory policy strengthens network access controls, these changes must be reflected in the organizational policy by setting access rights to certain networks to match the directives from the regulatory policy. Changes in organizational security policy similarly require adjustment to the machine policy for security control settings, configurations and security controls.

All policy updates must be carefully controlled and tracked with an audit trail.

12 LOOKING AHEAD—THE FUTURE OF THE IIOT

This document outlined best practices and considerations needed to address security risks associated with the Industrial Internet of Things. An overview of the differences between IIoT systems and traditional information technology systems was presented, as was the importance of considering key system characteristics and their relationship to risks, security assessments and risk analysis. The Functional and Implementation Viewpoints, described technologies and practices that affect the security and privacy of IIoT systems. This chapter provides an overview of the security implications that new technologies and trends may have on the future of the IIoT.

The industrial internet shows great promise for vastly increasing the capabilities of the devices in a variety of applications, including factory automation, medical systems and a wide variety of other systems. By connecting devices to enable communication among each other and the cloud, it opens the possibilities of making the devices “intelligent”, potentially delivering unprecedented capabilities.

Advances in technology will enable a new generation of devices that are more intelligent and have greater efficiency than their predecessors. This should lead to an inflection point where the promise of greater functionality, efficiency and intelligence will lead to more rapid updates of brownfield systems or perhaps even a wholesale set of greenfield upgrades. In OT, where system lifespans have historically been measured in decades, such inflection points are few and far between. As these high-value industrial systems are increasingly connected, it is even more critical that security and privacy risks be addressed.

Broad forces such as increasingly powerful microcontrollers, microelectromechanical system (MEMS) sensors, battery-friendly wireless protocols, horizontally scalable computing infrastructures, high-assurance microkernels, additive manufacturing, desktop milling, venture funding and crowd funding will likely continue to disrupt industries in unpredictable ways. New wireless protocols stream richer sensor data and new microcontrollers may be energy friendly enough to work strictly via energy harvesting, opening untold possibilities for instrumenting discrete manufacturing, refineries and countless production and treatment plants.

Of course, these revolutions in the core of industrial systems are not happening in isolation. Similar advances in sensing, instrumentation and automation of building controls are driving growth of smart building and smart cities. At the same time, falling costs of bandwidth and embedded processing are making it easier and easier to manage supply chains spanning tiers of suppliers scattered across multiple continents, including such exciting use cases as mass customization through just-in-time manufacturing of pipelined single-piece workflow. Traditional management models and operational architectures often don’t scale for such uses.

Traditional operational architectures revolve around a centralized management and monitoring capability to ensure proper functionality. This is analogous to the human brain and the series of nerves that allow the brain to push commands to muscles (actuators) and receive information (from sensors). Increasingly, though, in industrial settings, we have the opportunity to leverage tremendous distributed computing power for intelligence at the edge of the network.

Enabling edge devices to make decisions more autonomously can lead to more efficient processes, and provides the ability to react more quickly to events at the edge. This is analogous to the human reflex operation. There, the stimuli do not travel all the way to the brain for a response to be sent, rather, they are intercepted in the spinal column so that a response, to very specific stimulus, is returned very quickly. In industrial terms, this leads to a more scalable solution since all the raw data does not need to travel to central management, but rather faster reflexive actions are enabled through intelligence at the edge. Of course, as we empower the edge more, we need to protect those edge devices better.

Security management will likely face a similar shift from centralized to decentralized as the number of devices skyrockets as predicted. Similarly, as the sheer volume of data required for managing devices increases, it becomes clear there's a point where centralized management ceases to be effective and efficient. Instead, embedding security into each piece of equipment individually, and empowering the equipment with the security context required to make safe decisions, might become a far more scalable approach.

Meanwhile, the security aspects of such an autonomous world, if not handled correctly, could be devastating. Imagine a multitude of autonomous smart devices, all making decisions on their own. We must ensure that security can be implemented to maintain the integrity of the devices against attack. Therefore, a malicious entity cannot compromise the devices and cause them to make the wrong decisions at critical times. The key elements in this security scenario are ensuring integrity of the endpoints, enabling communications security, and providing the ability to update the endpoints more securely. To ensure compatibility across all of the various types of devices, it is desirable to have a common infrastructure that enables communications and management (and monitoring) across them all.

There are many related advances in technology that may enable improving the security of IIoT systems. Some of these technologies have been available for some time with varying degrees of deployment.

A software-defined network can separate networks and prevent packets from crossing between them, thus increasing security. They also allow IP addresses to be dynamically changed, making it harder for attackers to learn about the network and benefit from previous explorations.

Software-defined platforms and virtual machines allow separating computer systems and reducing the risks of an attack on a system affecting multiple functionalities on that system.

Protecting the confidentiality of private keys in endpoint devices and simplifying the provisioning can improve IIoT. Technologies such as *physical unclonable function* (PUF) [MIT-PUF] allow endpoint devices to behave as if they have private keys without storing a key, reducing the risk associated with attacks on hardware to retrieve a stored key. The adoption of this technology has been slow, possibly due to concerns with stability over time.

Privacy could be enhanced while still allowing analytics through techniques that allow calculations to be performed on encrypted text, such as homomorphic encryption. This may impose constraints on the data design used in the system.

Split key technology could be used to enable multi-party control (i.e., N of M) of IIoT system components such as actuators.

Not all technological advances will benefit the security of IIoT systems. Some, such as quantum computing, may reduce the viability of some cryptographic techniques while others may still be useable [UWAT-QC]. Thus, algorithm agility is appropriate in IIoT systems, including the ability to update algorithms in hardware securely.

Computing done at the network edge or perimeter areas is sometimes referred to as *fog* computing.[1] In fog computing, more processing is done at the network edge before being moved to the core network and the optional cloud repository. With the potential for billions of IoT devices creating data, it becomes challenging to move all of it at adequate speed through the network—this creates a data management issue at the edge of the network that must be addressed. The fog may become a viable deployment method to address these issues for IIoT. New consortia are in the early stages of defining reference architecture for fog. Once their reference architecture becomes better defined, the aspects of fog can be applied to IIoT security.

Similarly, management paradigms, especially for brownfield deployments will gain in capability with microservices. A *microservice* is an element that results from the architectural decomposition of an application's components into loosely coupled patterns consisting of self-contained services that communicate with each other using a standard communications protocol and a set of well-defined APIs, independent of any vendor, product or technology.

Industrial microservices are small autonomous software components that work to manage a particular aspect of a physical asset. Although the physical asset remains the same for years, the microservices used to manage them can be easily upgraded. There are many types of industrial internet microservices such as data microservices, common microservices, intelligent cities microservices and others.

Blockchain is a permission-less distributed database often used as a public ledger with integrity assurance. It maintains a continuously growing list of data records hardened against tampering and revision. Each block includes the hash of the prior block, linking the blocks together.

Blocks are in turn defined as small sets of transactions that have taken place within the system. Each new block includes a hash of the previous transaction, which "chains" it to all previous blocks. Blocks are computationally difficult to create, taking multiple specialized processors and significant amounts of time to generate.

In some IIoT systems creating a tamper-proof log of transactions or other information may have value. The blockchain technology could possibly support this as well as enabling multiple secure records of broadcast updates. Supply chain management is a key area where blockchains could be leveraged in the IIoT environment. Some of the advantages of blockchains are the ability for independent nodes to converge on a consensus of the latest version of a large data set such as a ledger. This provides consistency, validity of transactions and automated conflict resolution.

[1] See [OpenFog-Res]

All of these emerging technologies should have an impact on the IIoT, as well as the broader trends like decentralization.

Though arguably not as transformational as the industrial revolution in the 1800's, the Industrial Internet revolution will certainly bring about major improvements in the quality of our day-to-day lives. The world may see quicker adoption of IIoT in emerging countries thanks to more opportunity for new greenfield deployments. But, we must take care and apply the appropriate level of forethought and wisdom to ensure that the technological advances do not cost us dearly in the end. There may be a finite period—a window of opportunity—where we can design a cohesive security vision that realizes endpoint-to-endpoint secure communication and enables security management and monitoring.

The material in this document should enable developers in a wide variety of verticals, including the energy, healthcare manufacturing, transportation and public sectors—as well as developers of IIC testbeds—to incorporate security and privacy into their work.

This document is a living document that will be updated with experience from IIC testbed implementations, as well as changes in the risks and technologies in the surrounding environment.

Annexes

Annex A INDUSTRIAL SECURITY STANDARDS

Numerous guidelines, standards and regulations relate to the protection of Industrial Internet of Things systems. We discuss here the role of standards and compliance and introduce those that relate most to IIoT systems.

These could stem from the need to control access to financial systems (for example, Sarbanes-Oxley legislation), protect credit card information (from the PCI DSS standard), to protect critical infrastructure (such as NERC CIP, the ANSSI critical infrastructure standards or FDA 510(k) premarket submissions).[1] Equally from the OT side, there are a number of OT regulations that could be applicable to IIoT systems such as: Cybersecurity with ISA 99, IEEE PC37.240, Safety Integrity Level (SIL), Critical Infrastructure Protection (CIP), Critical Infrastructure Security (CIS), Current Good Manufacturing Practices (CGMP), Emissions control with Environment Protection Agency (EPA) and Marine Pollution (MARPOL), Facilities Standards with Energy Performance of Building Directive (EPBD) and Motor Efficiency with Minimum Energy Performance Standards (MEPS)[2].

A.1 ROLE OF STANDARDS AND COMPLIANCE IN SECURITY

Security standards guide and enforce a common level of security capability across an industry. Compliance with a standard requires taking steps to achieve the prescribed alignment, theoretically avoiding financial or other penalties for deviations from the standard's requirements. Standards rarely govern implementations, so a solution may be compliant with the standard but the resulting security posture may not be optimal. Design tradeoffs may also be necessary between levels of compliance and cost, ease of operation and maintainability.

The objective of securing IIoT systems is to address their availability, integrity and confidentiality requirements. The realization of an adequately secure environment should be guided by a series of informed decisions intended to ensure that the identified threats, vulnerabilities and countermeasures are commensurate with an acceptable level of risk. Security standards compliance is intended to guide an organization in best security practices, but it does not imply that the organization's products will be free of vulnerabilities or impenetrable to exploit.

Ideally, security implementations should also be updated periodically to adapt to newfound threats, possibly triggering the need to reassess standards compliance. Unfortunately, making such security updates may be infeasible or too costly. The operational functions and safety

[1] See [SarOxl], [PCI-DSS], [NERC-CIP], [ANSSI-CMKM] and [FDA-510K]

[2] See [ISA-99], [IEEE-C37-240], SIL at [IEC-61508], [NERC-CIP], [DHS-CIS], [FDA-CGMP], [EPA-SRG], [IMO-MARPOL], [EU-CA-EPBD], and [IEA-MEPS]

functions have traditionally been tightly coupled in industrial systems. Therefore, updates, upgrades or bug fixes could potentially require recertification according to the regulations (e.g., IEC 61508 or ISO 13849[1]) with the inclusion of notified bodies.

An example is the challenge presented by security updates for devices that have to be compliant with European Machinery Directive 2006/42/EC[2]. Since December 2009, this directive is binding in all member states of the European Union (EU). Only machines that comply with the directive may be sold within the EU. Original equipment manufacturers (OEMs) are responsible for this compliance. They must document the functional safety of every machine and must include the documentation in the delivery. The affixing of the CE marking on the machine symbolizes the self-declaration by the manufacturer that he is convinced that all the essential health and safety requirements of the relevant EC directives are met. [3] As a consequence of the machinery directive any update, upgrade or bug fix of the software or firmware that affects the safety aspects is only possible according to the steps following CE conformity, else the safety certification is voided.

The advent of Industrial Internet certainly adds to the challenges posed by combination of safety, security, and compliance requirements. Moving towards newer scalable regulatory paradigms[4] in addition to utilizing techniques for proper separation of safety, security and operational functions would pave the way for addressing those challenges.

A.2 COMMON STANDARDS AND REGULATION

The International Electro-technical Commission (IEC) publishes the IEC 62443[5] series of standards for industrial automation and control systems security. The series, broadly encompasses the concepts of manufacturing and control systems electronic security, covering different types of systems, facilities, and plants in various industries.

Presently, the series of standards under IEC 62443 is comprised of four groups. Group 1, labeled 'General,' presents a standardized terminology and aims at providing consistent models, references and metrics referred to by other groups. Policies and procedures for the creation of effective Industrial Automation and Control Systems (IACS) security programs are discussed in Group 2, labeled 'Policies & Procedures.' Group 3, labeled 'System' covers cybersecurity technologies, design methodologies, assessment approaches, security requirements and assurance levels. Requirements for secure development lifecycle for IACS and secure component development are discussed in Group 4, labeled "Component." As an example, IEC 62443-2-4

[1] See [IEC-61508] and [ISO-13849]
[2] See [EU-2006/42]
[3] See [EU-CE]
[4] A good example can be found in FDA's draft guidance on post-market management of cybersecurity in medical devices. The document develops the concept 'cybersecurity routine updates and patches', pointing out that they do not need to be reported under [FDA-CFR-21].
[5] See [IEC-62443-11]

'Security Program Requirements for IACS Service Providers'[1] standardizes security capabilities for integration and maintenance activities, allowing asset owners to select those most appropriate for their sites. In addition, parts 62443-2-4, 62443-3-1 and 62443-3-3[2] define a distinction of security levels based on an attackers' strength, which is valuable for system design. This standard is in the process of accreditation under the IEC System of Conformity Assessment Schemes for Electro-technical Equipment and Components (IECEE).[3] In time, it may be adopted as a security certification for use in securing operators' supply chains.

The National Institute of Standards and Technology (NIST) has published NIST SP 800-82 'revision 2'.[4] It provides guidance on improving security in Industrial Control Systems (ICS), including Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and other control system configurations such as Programmable Logic Controllers (PLC). Performance, reliability and safety requirements are also considered. Comprehensive security controls, presented in this document, map to additional NIST recommendations such as those listed in SP 800-53, 'Recommended Security Controls for Federal Information Systems and Organizations.'[5] A framework for considering networks of things is described in NIST SP 800-183[6].

NISTIR 7628, 'Guidelines for Smart Grid Cyber Security, Volume 1'[7], is a recommendation for addressing security concerns across the electric smart grid. NIST created this document with Smart Grid Interoperability Panel Cyber Security Committee. It is a three-volume compendium that contains sections that describe risk assessment and vulnerability analysis, and analyzes secure information exchange for electric grid systems.

NERC CIP Standards[8], published by the North American Electric Reliability Corporation (NERC), aim at improving the security and reliability of the electric industry by defining auditable requirements for critical infrastructure protection (CIP). Security guidance provided by NERC CIP primarily targets the automation systems used in generation and transmission facilities. NERC CIP is applicable to utilities in US, Canada and parts of Mexico.

The IEEE 1686 'Standard for Intelligent Electronic Devices Cyber Security Capabilities'[9] defines functions and features to be provided in Intelligent Electronic Devices (IEDs). The document addresses access, operation, configuration, firmware revision and data retrieval of an IED.

[1] See [IEC-62443-24]
[2] See [IEC-62443-24], [IEC-62443-31] and [IEC-62443-33]
[3] See [IECEE]
[4] See [NIST-800-82]
[5] See [NIST-800-53]
[6] See [NIST-800-183]
[7] See [NISTIR-7628]
[8] See [NERC-CIP]
[9] See [IEEE-1686]

A.3 METHODOLOGIES TO ASSESS SECURITY PROGRAMS

Several methodologies exist to assess security programs, the security posture of organizations and their process for secure development and maintenance of their products. They include the Cyber-Security Capability Maturity Model (C2M2)[1] and its vertical-specific variants (ES-C2M2 and ONG-C2M2 for energy and oil and gas subsectors, respectively), the tiers of the NIST framework focused on critical infrastructures, the CERT Resilience Management Model (CERT-RMM) focused on operational resilience management and the Building Security In Maturity Model (BSIMM) focused on secure software development. They work best when tailored for the organization.[2]

A.4 STANDARDS FOR EVALUATING SECURITY PRODUCTS

Common criteria and Federal Information Processing Standard (FIPS) standards, briefly discussed below, focus on certification of security products rather than evaluating security processes or policies. Within this context, these standards allow technical evaluations by third parties such as trusted labs.

Use of such evaluation approaches requires extra care, especially in terms of how they adapt to change and respond to the progress in attack technologies. There are many products with practically meaningless evaluations, because they've been evaluated in very restricted configurations, or because only some of their basic features have been evaluated.

A.4.1 COMMON CRITERIA

Common Criteria for Information Technology Security Evaluation, a.k.a. Common Criteria (CC), is an international standard (ISO/IEC 15408[3]) used to evaluate security capabilities of IT products, including secure integrated circuits, operating systems and application software. CC is used to assess a product's ability to meet security requirements utilizing two key notions: evaluation assurance levels and protection profiles.

The rigor with which an assessment is carried out is referred to as the Evaluation Assurance Level (EAL), which ranges from EAL1 up to EAL7. As an example, functional testing is sufficient to meet EAL1 requirements but to achieve EAL7 thorough testing as well as formally verified designs are required.

A protection profile consists of security requirements and their rationale as well as an EAL. A protection profile should describe objectives, assumptions and both functional and assurance requirements. When customers (i.e., owners or operators) plan to buy a product that has Common Criteria Evaluation, they should ensure that they understand and agree with the protection profile against which the product has been evaluated.

[1] See [ENER-C2M2] and Annex B
[2] See [CERT-RMM] and [BSIMM]
[3] See [ISO-15408]

A.4.2 FEDERAL INFORMATION PROCESSING STANDARD (FIPS)

The FIPS publication 140 Publication Series establishes requirements for cryptographic modules that include both hardware and software components. Topics covered by FIPS 140-2[1] include implementation and use of asymmetric and symmetric keys, message authentication, secure hashing, and random number generation. FIPS 140-2 specifies four qualitative security assurance levels, with each level representing increasingly more stringent controls to prevent physical access to information stored or managed by the modules.

A.5 SAFETY STANDARDS AND THEIR RELATIONSHIP WITH SECURITY

Most commonly accepted safety standards and guidance documents build on, adapt or derive from 'IEC 61508 Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems.'[2] Examples include ISO 26262 (automotive), IEC 62279 (rail), IEC 61511 (industrial process control), and IEC 61513 (nuclear reactor instrumentation and control).[3] The IEC 61508 family of standards generally requires safety to be handled as a first-class system property throughout the complete lifecycle of the system from requirements elicitation through end-of-life. These standards require that all hazards must be identified and the risks associated with those hazards must be reduced to As-Low-As-Reasonably-Possible (ALARP). Since security vulnerabilities enable adversaries to precipitate hazards, existing safety standards can be viewed as directly implying that system security too must be seriously considered at each stage of the system lifecycle (i.e., security must be *built-in* from the beginning and not added *post hoc*).

In addition to the IEC 61508 family of derived standards, there are a number of other guidance standards focused mainly on the development of software for safety critical systems. Examples include MISRA C (guidance on use of the C programming language in critical systems), DO-178B/C (software in aviation systems), and ARINC 653 (a standard separation kernel interface for avionics software systems).[4] Generally, these standards require a rigorous and well-documented development process for safety critical software. Often, extensive unit testing with high coverage or even formal methods must be used for verification. While these standards have been developed with safety in mind, the prescribed development processes can help reduce the introduction of exploitable defects.

A.6 PRIVACY STANDARDS, FRAMEWORKS AND REGULATION

A.6.1 ISO/IEC AND NIST PRIVACY STANDARDS

The International Organization for Standardization (ISO) has been working on a set of standards relating to privacy and protection of personally identifiable information (PII).

[1] See [FIPS-140-2]
[2] See [IEC-61508]
[3] See [ISO-26262-1], [IEC-62279], [IEC-61511] and [IEC-61513]
[4] See [MISRA-C], [RTCA-DO-178B], [RTCA-DO-178C] and [ARINC-653]

ISO/IEC 29100, 'Privacy framework'[1] provides a guideline that specifies a common privacy terminology, defines actors and roles in processing PII, describes privacy safeguarding considerations, and includes references to known privacy principles for Information technology.

ISO/IEC 29101, 'Privacy architecture framework'[2] specifies concerns for information and communication systems processing PII, lists components for implementation of such systems and provides architectural views that contextualize these components.

ISO/IEC 29190, 'Privacy capability assessment model'[3] provides high-level guidance to organizations about assessing their capability to manage privacy-related processes.

ISO/IEC 27018, 'Code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors'[4] establishes commonly accepted control objectives, controls and guidelines for implementing measures to protect PII in accordance with principles described in ISO/IEC 29100.

ISO/IEC 29134, 'Privacy impact assessment–Guidelines'[5] proposes a methodology to conduct assessments on the impact of privacy.

NISTIR 8062, 'Privacy Risk Management for Federal Information Systems'[6] describes a privacy risk management framework focused on privacy engineering objectives and a privacy risk model.

A.6.2 PRIVACY FRAMEWORKS

A framework is a conceptual structure for organizing activities in pursuit of a specific goal; e.g., transatlantic data flow. The European Union and the United States have agreed on a new framework for transatlantic data flows called the 'EU-US Privacy Shield'[7], because the European Court of Justice declared the earlier Safe-Harbor framework invalid in October 2015. Privacy Shield strengthens cooperation between the US Federal Trade Commission and EU Data Protection Authorities, providing independent, vigorous enforcement of the data protection requirements set forth in the Privacy Shield. At the time of writing this document, EU-US Privacy Shield is still a work in progress.

A.6.3 PRIVACY REGULATIONS

Many countries have published guidelines, standards or regulations to protect the Personally Identifiable Information (PII) and Protected Health Information (PHI) of their citizens. Notable

[1] See [ISO-29100]
[2] See [ISO-29101]
[3] See [ISO-29190]
[4] See [ISO-27018]
[5] See [ISO-29134]
[6] See [NISTIR-8062]
[7] See [US-EU-Prv-Sh]

examples are those of European Union (GDPR) and North America (HIPAA and PIPEDA).[1] Since regulations are mandatory, non-adherence could mean fines and even jail time. Best practice is to have privacy by design, default and deployment approach. Some of the privacy requirements might overlap with security requirements and should be considered concurrently.

More information on data privacy standards and regulations can be found at Baker & McKenzie's 'Global Privacy Matrix' and Electronic Frontier Foundation (EFF)'s 'International Privacy Standards'.[2]

A.7 PROTOCOL RESOURCES

Detailed evaluation of the security properties, weaknesses and strengths of industrial network protocols is not within the scope of the current draft of this document. However, pointing to corresponding resources, including associated security considerations is likely of interest to security engineers and architects:

Object Management Group manages the open specifications of the Data Distribution Service (DDS), including 'DDS Security Specification'. More information about the specification, its users and comparison with other technologies can be found at the OMG website.[3]

OPC Foundation maintains the open specifications of the OPC protocol. Information on OPC Classic, OPC UA, and OPC .NET (formerly OPC Xi) can be found at the OPC website.[4]

DNP User Group maintains the Distributed Network Protocol (DNP3). Technical information, conformance testing, and listing of conformant products can be found at the DNP website.[5]

Modbus Organization manages the development and use of Modbus protocols. Information about the Modbus protocols, as well as technical resources for development and testing of Modbus-based industrial systems can be found at the Modbus website.[6]

PROFIBUS and PROFINET International manages the PROFIBUS and PROFINET industrial protocols. Protocol specifications, technical documents and software tools can be found at the PROFIBUS website.[7]

Other standards and protocols that might be of interest to IIoT architects are MQTT and AMQP, both OASIS standards, and XMPP.[8] Common protocol definitions and standards, such as HTTP[9]

[1] See [EU-GDPR], [HHS-HIPAA] and [CA-PIPEDA]
[2] See [BaMcK-GPH] and [EFF-IPS]
[3] See [OMG-DDS]
[4] See [OPC-classic], [OPC-NET] and [OPC-UA]
[5] See [DNP]
[6] See [Modbus]
[7] See [PI-pbus] and [PI-pnet]
[8] See [MQTT], [AMQP], [OASIS] and [XMPP]
[9] See [IETF-RFC7230]

are maintained by IETF. The more secure version, HTTP/TLS,[1] is recommended whenever possible over HTTP.

A.8 CLOUD SECURITY STANDARDS

There are a great number of guidelines and standards pertaining to cloud security, devised and used in various countries. We briefly describe a few notable ones below.

The ISO/IEC 27017[2] standard provides guidance on the information security elements of cloud computing. It assists with the implementation of cloud-specific information security controls, supplementing the guidance in ISO 27000 series standards, including ISO/IEC 27018 on the privacy aspects of cloud computing, ISO/IEC 27031 on business continuity, and ISO/IEC 27036-4 on relationship management, as well as all the other ISO 27nnn standards.[3]

NIST has also published the following standards on cloud computing: NIST SP 800-146, 'Cloud Computing Synopsis and Recommendations', NIST SP 500-291, 'Cloud Computing Standards Roadmap', NIST SP 800-144, 'Guidelines on Security & Privacy in Public Cloud Computing', NIST SP 500-292, 'Cloud Computing Reference Architecture' and NIST SP 500-293, 'US Cloud Computing Technology Roadmap'.[4]

European Union Agency for Network and Information Security (ENISA) has published an auditable standard titled 'Cloud Computing: Benefits, risks and recommendations for information security'[5] to which many cloud providers are certified.

'Cloud Computing Security Considerations'[6] by the Australian Signals Directorate provides analysis and measurement of risk that will be considered by cloud SaaS customers when evaluating the cloud as a potential solution.

Cloud Security Alliance has published many guidelines, including:

'Security Guidance for Critical Areas of Focus in Cloud Computing Version 3.0,'[7] that contains practical, current guidance and advice for both cloud computing customers and providers.

'Practices for Secure Development of Cloud Applications'[8] provides practical guidance relevant to cloud SaaS such as secure design recommendations for multi-tenancy and data encryption, and secure implementation recommendations for securing APIs.

[1] See [IETF-RFC2818], commonly known as HTTPS
[2] See [ISO-27017]
[3] See [ISO-27000], [ISO-27018], [ISO-27031] and [ISO-27036-4]
[4] See [NIST-800-146], [NIST-500-291], [NIST-800-144], [NIST-500-292] and [NIST-500-293]
[5] See [ENISA-CCRA]
[6] See [AU-CCSC]
[7] See [CSA-SGCA]
[8] See [CSA-SCCSA]

'Cloud Controls Matrix Version 3.0,'[1] which is an auditable standard that is mapped to a large set of other standards including COBIT, ISO/IEC 27001:2005, NIST SP 800-53, FedRAMP, PCI DSS, HIPAA/HITECH, NERC CIP[2]. The Cloud Controls Matrix provides fundamental security principles to guide cloud vendors and to assist prospective customers in assessing the overall security risk of a cloud provider. A cloud provider offers transparency into how its security controls are designed and managed by completing an assessment against the Cloud Controls Matrix.

A.9 STANDARD REPOSITORIES

The Smart Grid Interoperability Panel (SGIP) has created a compendium of standards and practices pertaining to the development and deployment of the Smart Grid. A table of the documents contained in the SGIP Catalog of Standards on the SGIP website.[3]

Specific guidance for securing industrial control systems using the TCG standards is included in these documents: 'TCG Architect's Guide for ICS Security', 'TCG Architect's Guide for IoT Security,' and 'TCG Guidance for Securing IoT.'[4] These documents present approaches to industrial control systems security, addressing communications security, system integrity, firmware updates and detection and recovery from sophisticated attacks.

SAE standards target safety, quality and effectiveness of products and services across the mobility engineering industry. The more than 10,000 standards in the SAE database now include historical standards and can be accessed at the SAE website.[5]

A.10 SUPPLY CHAIN INTEGRITY RESOURCES

Manufacturers should apply best practices of supply chain risk assessment and risk management. The NIST 'Supply Chain Risk Management: Practices for Federal Information Systems and Organizations'[6] provides guidance to US federal agencies on identifying, assessing and mitigating supply chain risks at all levels of their organizations. It also integrates ICT supply chain risk management (SCRM) into federal agency risk management activities by applying a multi-tiered, SCRM-specific approach, including guidance on assessing supply chain risk and applying mitigation activities.

Manufacturers should also follow best practices for supply chain security. One example is ISO 'Information Security for Supplier Relationships'[7]. Another is the NEMA 'Supply Chain Best Practices'[8]. This document identifies a recommended set of supply chain best practices and

[1] See [CSA-CCM]
[2] See [ISACA-COBIT], [ISO-27001], [NIST-800-53], [FedRAMP], [PCI-DSS], [HHS-HIPAA], [HHS-HITECH] and [NERC-CIP]
[3] See [SGIP-CoS]
[4] See [TCG-AG-ICS], [TCG-AG-IoT] and [TCG-GS-IoT]
[5] See [SAE]
[6] See [NIST-800-161]
[7] See [ISO-27036-1]
[8] See [NEMA-CPSP]

guidelines that electrical equipment and medical imaging manufacturers can implement during product development to minimize the possibility that bugs, malware, viruses or other exploits can be used to negatively impact product operation.

Annex B CYBER SECURITY CAPABILITY MATURITY MODEL (C2M2)

The *Cyber Security Capability Maturity Model* (C2M2) evaluates the maturity of an organization's security posture and processes. The model allows for:

- assessment of the existing state of the security posture of the organization and its products,
- establishment of a target security profile, which states what security goals are to be achieved given the current state, existing risks, and business strategies and
- prioritization of the gaps identified between the current state and the target profile and identification of required security activities for addressing those gaps.

The model presents a holistic approach to securing Industrial Internet of Things systems and their components throughout their lifetime, from early design to implementation, deployment, maintenance and retirement. It includes evaluation of specific security technologies and the managerial and business context in which they are used. Such context is absolutely necessary to identify threats and manage risks to IIoT systems.

C2M2 was developed by the U.S. Department of Energy in conjunction with industry representatives. Interested readers can refer to the C2M2 framework [ENER-C2M2] for more detailed information. The model is summarized here because it is this model the Industrial Internet Consortium uses as part of the IIC testbed process.

The C2M2 maturity model is a set of characteristics, indicators or patterns that represent capability and progression of behaviors, practices and processes in a particular discipline. An associated assessment methodology defines best-practice activities, typically grouped into practice areas. Each requirement is given a score corresponding to a discrete *maturity level* that rates the extent to which a best practice is repeatable, practiced and its effectiveness measured. Each practice area has its own score, as an averaged score is not useful in guiding corrective actions.

B.1 LOGICAL GROUPINGS

The C2M2 model comprises ten logical groupings of security activities (*domains*). Each domain[1] has a number of objectives, each of which has a progression of practices at different levels of maturity—a *Maturity Indicator Level* (MIL). For example, the *Supply Chain and External Dependencies Management* domain is a group of practices that an organization can perform to manage the risks associated with services and assets that are dependent on external entities. This domain includes the following objectives: identify dependencies, manage dependency risk and management activities. The first objective in this domain includes practices such as identification of Information Technology (IT) and Operational Technology (OT) supplier dependencies, identification of customer dependencies, and prioritization of dependencies. Each

[1] In the IIRA domains are used to segment functionality in the IIoT. In the C2M2 they are used to segment security activities.

of these practices could have different associated maturity levels, from being not implemented to fully implemented.

The domains are:

- *Risk Management* addresses establishment, operation and maintenance of a risk management program to identify, analyze and address security risks as they relate to the organization, business units, subsidiaries, related infrastructure and stakeholders.
- *Asset, Change and Configuration Management* targets management of the IT and OT assets, including hardware, software and integrated subsystems.
- *Identity and Access Management* targets creation and management of identities for entities that may be granted access to an organization's assets. Credentials may include access by individuals, shared roles (e.g., *operator*), devices or distributed services within or across network zones.
- *Threat and Vulnerability Management* targets establishment and maintenance of plans, procedures and technologies to detect, identify, analyze, manage and respond to security threats and vulnerabilities.
- *Situational Awareness* is an understanding of the relevant environment. The establishment and maintenance of activities and technologies to collect, analyze, alert, present and use operational and security information contribute to a holistic operating picture. This includes status summaries from other C2M2 domains.
- *Information Sharing and Communication* establishes and maintains relationships with internal and external entities to collect and provide security information, including threats and vulnerabilities, to reduce risks and increase operational resilience.
- *Event and Incident Response, Continuity of Operations* establishes and maintains plans, procedures and technologies to detect, analyze and respond to security events and to sustain operations throughout a security event.
- *Supply Chain and External Dependencies Management* establishes and maintains controls to manage security risks associated with services and assets that are dependent on external entities, including third-party component and service providers and open source component inclusion.
- *Workforce Management* creates a culture of security and ensures the ongoing suitability and competence of all personnel.
- *Security Program Management* targets establishment and maintenance of a security program that provides governance, strategic planning and sponsorship to align security objectives with organizational strategic objectives and risk to its critical infrastructure.
- The recommended approach for using the framework is to evaluate, identify and analyze gaps in capability, prioritize those gaps to be addressed, develop plans to address the gaps and implement plans for addressing them. This process should be repeated as the business objectives and risk environment changes over time.

B.2 ASSESSMENT PROCESS

Assessors are responsible for leading security evaluations. Such assessors are referred to as facilitators in C2M2 model. Details about how facilitators should use C2M2 can be found in C2M2 Facilitator guide [ENER-C2M2].

An assessment has *assessors* and *participants*. Assessors score and document their observations clearly and objectively; it is not their role to set priorities or dictate implementation details. Multiple assessors can compare notes and reconcile scoring discrepancies; they should be familiar with the content of the model and its artifacts.

Participants are stakeholders in the organizational, system definition, development and maintenance functions. A single participant acts as the primary point of contact with the assessors and takes responsibility for preparation, execution and follow-up. Participants may include product managers, systems and software architects, field service engineers, network engineers, security engineers, software managers and engineers, quality process managers and those involved in testing, validation, deployment and incident response.

The assessors describe the current security posture of the system by generating a scoring report. The scores identify gaps in the performance of model practices. A scoring report can be generated using a Microsoft Excel sheet[1], a scoring report with an example file[2] is shown in Figure B-1. Numbers in the white circles indicate total number of activities for a given domain and MIL level. Numbers in dark green, light green, light red, and dark red represent fully implemented, largely implemented, partially implemented and not implemented activities for each domain at each MIL level.

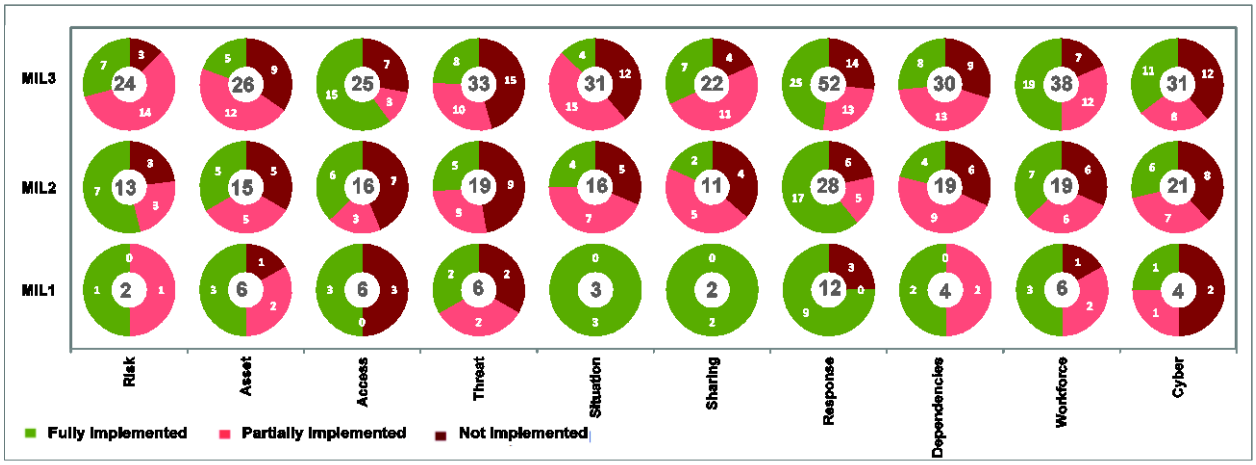

Figure B-1: A Sample C2M2 Score Report

The next step is to determine whether the gaps are important for the organization to address. Note that achieving the highest maturity level for every domain in the assessment might not be

[1] See [NRECA-Tmpl]
[2] See [NRECA-Smpl]

optimal. For this reason, the organization should have a *target security profile* with desired MIL ratings for each domain to meet its business objectives and security strategy. Comparing the target security profile with the assessment guides investment priorities for improving system security posture.

To reach the desired maturity level as defined by the target security profile, perform a cost-benefit analysis of the gaps and identify the activities to address them. In the process, objective criteria should be used, such as how gaps affect organizational objectives, how important the business objectives supported by the domain are, and what costs are associated with implementing the required practices. Based on this analysis, plans to address the gaps should be developed, implemented and tracked to ensure progress. It is required to cycle through evaluation, gap analysis, prioritization, planning and implementation as the business, technology, market, risks and threat environment change.

B.2.1 Assessment Process Requirements

An organization's assessment procedures should be properly documented, with materials available for the training of new members. The process should include a presentation to the parties involved that outlines expectations, rationale and expected outcomes. The *scoring activity* is solely as an attempt at quantification for the purposes of process improvement. Participants must understand that an assessment is not a corporate audit, and that no penalties apply for non-compliance. Full cooperation and truthful exchange of information is necessary for accurate measurement, and anecdotal information about activities should be supported with documented evidence of repeatable procedures.

B.2.2 Assessment Artifact Requirements

Artifacts used as evidence to support stated activities should be classified and handled accordingly. The assessment generates observations and action plans that must also be managed appropriately. The quantitative portions of an assessment should be recorded and tracked over time to indicate and analyze trends. Assessments should be scheduled regularly, with slightly greater frequency at the beginning of a program rollout.

This high-level process helps organizations ensure they methodically capture and prioritize required security activities within the constraints set by business strategy, risks and availability of resources.

Annex C SECURITY CAPABILITIES AND TECHNIQUES TABLES

This annex lists some security techniques and processes, their mapping to important security objectives, and their high-level requirements. With the ever-changing attack landscape, new techniques and processes are devised every day. This list cannot be comprehensive.

Cryptographic Technique		Example Objective	Example Requirements	
Symmetric key cryptography	MACs	Message authentication; Message integrity	Securely generated, distributed and maintained, shared secret key	Secure standardized and up-to-date MAC algorithm
	Symmetric encryption	Confidentiality		Secure standardized and up-to-date encryption algorithm
Asymmetric key cryptography	Digital signatures	Authorship; Integrity; Non-repudiation	Public-key infrastructure	Standard-based securely generated, distributed and maintained, public and private keys; Standardized and up-to-date signature schemes
	Asymmetric encryption	Confidentiality		Standardized and up-to-date asymmetric encryption algorithm
	Shared secret establishment	Forward secrecy		Standardized and up-to-date shared secret establishment algorithm
Hash function		Message/data integrity		Standardized and up-to-date hashing algorithm
Random number generator		Random key and other data	Proper random seed	Standardized and up-to-date random generator

Table C-1: Cryptographic Techniques, their Objectives and Requirements

Table C-1 identifies fundamental cryptographic building blocks in wide use in IT systems. For example, digital signatures are a type of asymmetric key cryptography designed to ensure authorship, integrity, and non-repudiation of data. Correct implementation of digital signatures, however, depends on existence of a public-key infrastructure (PKI), standard-based securely generated, distributed, and maintained key pairs, and standardized and up-to-date signature schemes.

Objective: Integrity		Example Technique/Process	Example Requirements
Endpoint integrity	Integrity for roots of trust	Protected key store	Integrity of protected storage for key management
	Integrity of endpoint identity	Identity certificate signed by trusted certificate authority	Trusted public-key infrastructure
	Hardware integrity	Side channel measurements; silicon scanning	Open, standards-based specification
	Software integrity	Code signing	Trusted public-key infrastructure
		Secure software development; Risk-based security testing; Static analysis	Secure software development methodology
		Boot process integrity	Trusted hardware manufacturer; Hardware security module or proprietary implementation of hardware backed cryptographic boot protection; Standardized OS firmware interface (e.g. UEFI)
		Secure patch management	Patch management plan
	Runtime integrity	Runtime verification	Code execution modeling, instrumentation and monitoring
	Integrity of data-at-rest	MACs, hashes/digests; Digital Signatures	Securely generated, distributed and maintained keys; Standardized and up-to-date algorithms
Integrity of communications		Mutual authentication between endpoints; use of MACs and/or digital signatures during communication	Securely generated, distributed and maintained keys; Standardized and up-to-date algorithms for mutual authentication and message exchange integrity
Integrity of management and monitoring operations		Authentication of management and monitoring assets (including workforce); Integrity verification of asset changes, asset monitoring solutions and asset Updates; Maintaining integrity of logs and reports	Endpoint integrity for management and monitoring; Communication integrity for monitoring, logging and management of assets; Security procedures for managing management and monitoring operations; Integrity of analytical algorithms; Integrity of audit or audit path
Architectural integrity	Integrity of data-in-motion	Holistic assessment of data integrity in its lifecycle across the entire IIoT system	Endpoint, communication, monitoring and management integrity in system segments
	Mutual impact of integrity controls on other key system characteristics	Architectural integrity evaluation	Holistic security evaluation methodology; Domain-specific expertise
	Mitigating impact of both insider and outsider attacks on system integrity	Enforcing principle of least privilege; Access control	Granular access control policies

Table C-2: Techniques and Processes for Enabling System Integrity

Table C-2 lists techniques and processes that aim at realizing integrity requirements in IIoT systems. Integrity requirements could be categorized into those aiming at:

- integrity of endpoints,

- integrity of communications,
- integrity of management and monitoring operations and
- holistic, architectural integrity of the entire system.

For example, techniques for ensuring the integrity of the software run on (or as) endpoints include secure software development and risk-based security testing.

Note that correct implementation of a technique or process may depend on correct implementation of another technique or process. For example, integrity of communications depends in part on proper implementation and usage of cryptographic techniques such as MACs and digital signatures, which in turn depend on proper generation, distribution, and management of keys.

Table C-3 summarizes techniques and processes that aim at realizing availability requirements. Notable examples include techniques for architectural availability of the system, mitigating or preventing denial of service attacks. Successful deployment of these techniques, which include load balancing and fault tolerance measures among others, depends on architectural threat modeling.

Objective: Availability	Example Technique/Process	Example Requirements
Endpoint availability	Physical protective enclosure	Trusted manufacturing and deployment
Availability of communications	Physical availability of communications media; Network load management; Anti-jamming techniques	Trusted manufacturing and deployment
Availability of management and monitoring operations and solutions	Resource allocation; Planning for frequent iterative security evaluation	Evaluation methodology; Endpoint, communications and architectural availability for management and monitoring components
Architectural availability	Redundancy; Avoiding single points of failure; Fault tolerance; Load balancing; Honeypots	Architectural threat modeling

Table C-3: Techniques and Processes for Enabling System Availability

Table C-4 summarizes techniques and processes that aim to realize confidentiality requirements for endpoints, communications and connectivity, and management and monitoring operations. Confidentiality requirements should also be evaluated architecturally for the whole IIoT system as indicated in the table. For example, access control techniques should be used to enforce the principle of least privilege, thereby reducing the impact of a possible breach by insiders. This requires policies derived from architectural threat modeling.

Objective: Confidentiality		Example Technique/Process	Example Requirements
Confidentiality at endpoints		Encrypted data storage	Securely generated, distributed, and maintained keys; Protective storage of sensitive key material; Standardized and up-to-date encryption algorithms
Confidentiality of communication		Encrypted communication	Securely generated, distributed, and maintained keys; Standardized and up-to-date encryption algorithms
Confidentiality of management and monitoring operations and solutions		Encrypted communication	Endpoint confidentiality and communications confidentiality
Architectural confidentiality	Confidentiality of data in its lifecycle		Endpoint confidentiality; communications confidentiality; Confidentiality of management and monitoring
	Mutual impact of confidentiality controls on other key system characteristics	Architectural confidentiality evaluation	Holistic security evaluation methodology; Domain-specific expertise
	Mitigating impact of both insider and outsider attacks on confidentiality	Enforcing principle of least privilege; Access control	Granular access control policies

Table C-4: Techniques and Processes for Enabling System Confidentiality

Availability, integrity, and confidentiality are generally referred to as core security objectives. Other security objectives are often derived from one or more of these requirements. An important example is access control. Due to the prominence of access control for IIoT systems, a list of techniques and processes associated with it is mentioned in Table C-5.

Objective: Access Control		Example Technique/Process	Example Requirements
Endpoint access control	Confinement and information flow protection within endpoint	Sandboxing (application); Fine-grained data-centric access control (middleware); Separation kernels (OS); Trusted computing environments (hardware)	Comprehensive and consistent security policies
Communications access control	Cryptographic protection of communications and connectivity	Use of protocols at different layers; Forcible disconnection of unauthorized endpoints;	Correct and trusted implementation of cryptographic techniques; Network access control for endpoints
	Information flow control	Network segmentation; Gateways and filtering; Network firewalls; Unidirectional gateways	Comprehensive and consistent security policies; Trusted manufacturing of devices
Access control for management and monitoring operations			Access control for monitoring, logging and managing assets (e.g. endpoints, communication, data, workforce); Control procedures for managing and monitoring operations; Controlling access to data that is fed into analytics solutions; Separation of duties; Role-based access control (RBAC)
Architectural access control	Controlling access to data in its lifecycle		Access control within endpoints, communication, management and monitoring
	Mutual impact of access controls on other key system characteristics	Architectural access control evaluation	Holistic security evaluation methodology; Domain-specific expertise
	Mitigating impact of both insider and outsider attacks on access control	Enforcing principle of least privilege	Granular access control policies

Table C-5: Techniques and Processes for Enabling System Access Control

Annex D REVISION HISTORY

Revision	Date	Editor	Changes Made
V1.0	2016-09-19	Mellor, Buchheit et al.	Initial Release

Annex E ACRONYMS

AES	Advanced Encryption Standard
AIC	Availability, Integrity, Confidentiality
AICPA	American Institute of CPAs
AMQP	Advanced Message Queuing Protocol
ANSSI	Agence Nationale de la Sécurité des Systèmes d'Information
API	Application Program Interface
ASIC	Application-Specific Integrated Circuit
BIOS	Basic Input/Output System
BSIMM	Building Security in Maturity Model
C2M2	Cyber-Security Capability Maturity Model
CA	Certificate Authority
CERT	Computer Emergency Response Team
CIA	Confidentiality, Integrity and Availability
CIP	Critical Infrastructure Protection
COTS	Commercial off the Shelf
CPU	Central Processor Unit
CRC	Cyclic Redundancy Check
CRL	Certificate Revocation List
CRM	Customer Relationship Management
CWE	Common Weakness Enumeration
DAR	Data-at-Rest
DDS	Data Distribution Service
DIM	Data-in-Motion
DIU	Data-In-Use
DLP	Data Loss Prevention
DSA	Digital Signature Algorithm
DSS	Data Security Standard
DTLS	Datagram Transport Layer Security
EAAF	Entity Authentication Assurance Framework
ECC	Elliptic Curve Cryptography
EFF	Electronic Frontier Foundation
ENISA	European Union Agency for Network and Information Security
EU	European Union
FDA	Food and Drug Administration
FIPPS	Fair Information Privacy Principles
FIPS	Federal Information Processing Standard
FPGA	Field-Programmable Gate Array
FTC	Federal Trade Commission
FTP	File Transfer Protocol
GAPP	Generally Accepted Privacy Principles
GDPR	General Data Protection Regulation
HID	Host Intrusion Detection

HIP	Host Intrusion Protection
HIPAA	Health Insurance Portability and Accountability Act
HITECH	Health Information Technology for Economic and Clinical Health
HMAC	Keyed-Hash Message Authentication Code
HRoT	Hardware Root of Trust
HSM	Hardware Security Module
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
IACS	Industrial Automation and Control System
ICCP	Inter-Control Center Communications Protocol
ICS	Industrial Control System
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IIC	Industrial Internet Consortium
IIoT	Industrial Internet of Things
IIRA	Industrial Internet of Things, Volume G1: Reference Architecture
IISF	Industrial Internet of Things, Volume G4: Security Framework
IoT	Internet of Things
IP	Internet Protocol
IPS	Intrusion Prevention System
IPsec	Internet Protocol Security
ISA	International Society of Automation
ISO	International Organization for Standardization
IT	Information Technology
KPI	Key Performance Indicator
LAN	Local Area Network
LOA	Levels of Authentication
MAC	Media Access Control
MCU	Microcontroller Unit
MEMS	Microelectromechanical System
MQTT	Message Queuing Telemetry Transport
NAC	Network Access Control
NAT	Network Address Translation
NERC	North American Electric Reliability Corporation
NIST	National Institute of Standards and Technology
NPE	Non-Person Entity
OASIS	Organization for the Advancement of Structured Information Standards
OCSP	Online Certificate Status Protocol
OECD	Organization for Economic Co-operation and Development
OEM	Original Equipment Manufacturer
OMG	Object Management Group
ONG	Oil aNd Gas

OPC	Open Platform Communications
OS	Operating System
OSI	Open Systems Interconnection
OT	Operational Technology
OWASP	Open Web Application Security Project
PaaS	Platform as a Service
PCI	Payment Card Industry
PII	Personally Identifiable Information
PIPEDA	Personal Information Protection and Electronic Documents Act
PLC	Programmable Logic Controller
PPTF	Protection of Privacy and Transborder Flows of Personal Data
PUF	Physical Unclonable Function
QoS	Quality of Service
QR	Quick Response
RADIUS	Remote Authentication Dial-In User Service
RAM	Random Access Memory
RFC	Request for Comment
RFID	Radio Frequency Identification
RMI	Remote Method Invocation
RMM	Resilience Management Model
RNG	Random Number Generator
RoT	Roots of Trust
RPC	Remote Procedure Call
RSA	Rivest Shamir Adleman (encryption algorithm)
RTU	Remote Terminal Units
RX	Receiver
SaaS	Software as a Service
SAML	Security Assertion Markup Language
SCAP	Security Content Automation Protocol
SLA	Service Level Agreement
SLC	Short Lived Certificates
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SoC	System on Chip
SSD	Solid State Disk
STRIDE	Spoofing Identity, Tampering with Data, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege
TCG	Trusted Computing Group
TCP	Transmission Control Protocol
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TPM	Trusted Platform Module
TX	Transmission
UEFI	Unified Extensible Firmware Interface

UI	User Interface
UL	UL LLC
US	United States
USB	Universal Serial Bus
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WAN	Wide Area Network
WASC	Web Application Security Consortium
WLAN	Wireless Local Area Network
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language
X-ray	X-radiation

Annex F GLOSSARY

This document uses specific words and phrases which are defined in the Industrial Internet Consortium (IIC) 'Industrial Internet of Things, Volume G8: Vocabulary' [IIC-IIV2016] .

Annex G REFERENCES

[AAMI-TIR2014] Association for the Advancement of Medical Instrumentation (AAMI) Technical Information Report (TIR) 38:2014

[AICPA-GAPP] GAPP: Generally Accepted Privacy Principles, AICPA (American Institute of CPAs), retrieved 2016-09-02
http://www.aicpa.org/InterestAreas/InformationTechnology/Resources/Privacy/GenerallyAcceptedPrivacyPrinciples

[AMQP] Advanced Message Queuing Protocol (AMQP), retrieved 2016-09-05
http://www.amqp.org/

[Andr-Trusty] The Android Source Code: Trusty TEE, Android Open Source Project, retrieved 2016-09-02
https://source.android.com/security/trusty/

[ANSSI-CMKM] Agence nationale de la sécurité des systèmes d'information: Classification Method and Key Measures, Cybersecurity for Industrial Control Systems, 2014
http://www.ssi.gouv.fr/uploads/2014/01/industrial_security_WG_Classification_Method.pdf

[ARINC-653] Rockwell Collins/Aeronautical Radio, Inc (ARINC): ARINC 653 (Avionics Application Standard Software Interface), retrieved 2016-09-05
http://store.aviation-ia.com/cf/store/catalog_detail.cfm?item_id=496
from
http://store.aviation-ia.com/cf/store

[ARM-TrustZ] ARM Ltd: ARM Trustzone, retrieved at 2016-09-02
http://www.arm.com/products/security-on-arm/trustzone

[ATT-CK] ATT&CK: Adversarial Tactics, Techniques & Common Knowledge, MITRE
https://attack.mitre.org/wiki/Main_Page

[AU-CCSC] Australian Government, Department of Defence: Cloud Computing Security Considerations, September 2012, retrieved 2016-09-05
http://www.asd.gov.au/publications/protect/Cloud_Computing_Security_for_Cloud_Service_Providers.pdf
from
http://www.asd.gov.au/infosec/cloudsecurity.htm

[BaMcK-GPH] Baker McKenzie: Global Privacy Handbook, retrieved 2016-09-05
http://globalprivacymatrix.bakermckenzie.com/

[BDI-CRTM] Der Core Root of Trust for Measurement (CRTM). Bundesamt für Sicherheit in der Informationstechnik (in German)
https://www.bsi.bund.de/DE/Themen/Cyber-Sicherheit/Aktivitaeten/TrustedComputing/TrustedPlatformModuleTPM/TrustedPlatformModuleTPM/CRTM.html

[BIOS] BIOS Boot Specification V1.01, 1996,
http://www.uefi.org/sites/default/files/resources/BIOSBootSpecs1.0.1_20091031.zip

[BSIMM] Building Security in Maturity Model (BSIMM)
https://www.bsimm.com/download/ from
http://www.bsimm.com

[CAPEC] CAPEC: Common Attack Pattern Enumeration and Classification: A Community Resource for Identifying and Understanding Attacks, MITRE
https://capec.mitre.org/

[CA-PIPEDA] Office of the Privacy Commissioner of Canada: The Personal Information Protection and Electronic Documents Act (PIPEDA), retrieved 2016-09-05
https://www.priv.gc.ca/leg_c/leg_c_p_e.asp

[CCSS-AIOT] Cloud Standards Customer Council (CSCC): Cloud Customer Architecture for IoT
http://www.cloud-council.org/deliverables/CSCC-Cloud-Customer-Architecture-for-IoT.pdf

[CERT-RMM] CERT: CERT Resilience Management Model, Carnegie Mellon University, Software Engineering Institute, Version 1.2, 2016-February
https://www.cert.org/resilience/products-services/cert-rmm/rmm-pa-download.cfm from
http://www.cert.org/resilience/products-services/cert-rmm/

[CFR-236] Title 49 Code of Federal Regulations (CFR) Part 236, Subpart H, US Department of Transportation, Federal Railroad Administration,
https://www.gpo.gov/fdsys/pkg/CFR-2011-title49-vol4/pdf/CFR-2011-title49-vol4-part236.pdf

[CSA-CCM] Cloud Security Alliance (CSA): Cloud Controls Matrix, version 3.0.1, 07/10/2014, retrieved 2016-09-05
https://cloudsecurityalliance.org/download/cloud-controls-matrix-v3-0-1/ from
https://cloudsecurityalliance.org/group/cloud-controls-matrix/

[CSA-IOT] Cloud Security Alliance (CSA): Internet of Things Working Group
https://cloudsecurityalliance.org/group/internet-of-things/

[CSA-SCCSA] Cloud Security Alliance (CSA): SAFEcode/CSA: Practices for Secure Development of Cloud Applications, December 2013, retrieved 2016-09-05 *https://downloads.cloudsecurityalliance.org/initiatives/collaborate/safecode/SAFECode-CSA-Cloud-White-Paper.pdf* from *https://cloudsecurityalliance.org/group/security-guidance/*

[CSA-SGCA] Cloud Security Alliance (CSA): CSA Security Guidance Version 3, 11/14/2011, retrieved 2016-09-05 *https://cloudsecurityalliance.org/download/security-guidance-for-critical-areas-of-focus-in-cloud-computing-v3/* from *https://cloudsecurityalliance.org/download/safecode-csa-whitepaper/*

[CVE] CVE: Common Vulnerabilities and Exposures (CVE), MITRE *https://cve.mitre.org/about/*

[CWE] CWE: Common Weakness Enumeration: A Community-Developed Dictionary of Software Weakness Types, MITRE, retrieved 2016-09-02 *https://cwe.mitre.org/*

[DHS-CIS] Department of Homeland Security: Critical Infrastructure Security, retrieved at 2016-09-02 *https://www.dhs.gov/topic/critical-infrastructure-security*

[DNP] Distributed Network Protocol (DNP): DNP3, retrieved 2016-09-05 *http://www.dnp.org*

[Docker] Docker Inc: Build, Ship, Run. An open platform for distributed applications for developers and sysadmins, retrieved 2016-09-02 *https://www.docker.com*

[EFF-IPS] Electronic Frontier Foundation (EFF): International Privacy Standards, retrieved 2016-09-05 *https://www.eff.org/issues/international-privacy-standards*

[ENER-C2M2] Office of Electricity Delivery & Energy Reliability: Cybersecurity Capability Maturity Model (C2M2) *http://energy.gov/sites/prod/files/2014/03/f13/C2M2-v1-1_cor.pdf* from *http://energy.gov/oe/services/cybersecurity/cybersecurity-capability-maturity-model-c2m2-program/cybersecurity*

[ENISA-CCRA] European Union Agency for Network and Information Security (ENISA): Cloud Computing Risk Assessment, 2009-November-20, retrieved 2016-09-05 *https://www.enisa.europa.eu/publications/cloud-computing-risk-assessment*

[ENISA-SFGC] European Union Agency for Network and Information Security (ENISA): Security Framework for Governmental Clouds, 2015-February-26, retrieved 2016-09-05
http://www.enisa.europa.eu/publications/security-framework-for-govenmental-clouds

[EPA-SRG] US Environmental Protection Agency: Emission Standards Reference Guide for On-road and Nonroad Vehicles and Engines, retrieved 2016-09-05
https://www.epa.gov/emission-standards-reference-guide

[EU-2006/42] European Union: Directive 2006/42/EC of the European Parliament and of the council of 17 May 2006 on machinery, and amending Directive 95/16/EC, retrieved 2016-09-05
http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32006L0042&from=EN

[EU-2016/679] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), retrieved 2016-09-13
http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679&qid=1473816357502&from=en
from
http://data.europa.eu/eli/reg/2016/679/oj

[EU-CA-EPBD] European Union: Concerted Action Energy Performance of Buildings, retrieved at 2016-09-02
http://www.epbd-ca.eu/

[EU-CE] European Union: European Commission, Blue Guide (Growth, Internal Market, Industry, Entrepreneurship and SMEs), retrieved 2016-09-05
http://ec.europa.eu/DocsRoom/documents/18027/attachments/1/translations/de/renditions/pdf
from
http://ec.europa.eu/growth/single-market/ce-marking/

[EU-GDPR] European Union General Data Protection Regulation (GDPR), retrieved 2016-09-05
http://www.eugdpr.org/

[FDA-510K] U.S. Food and Drug Administration: Medical Devices, 510(k) Clearances, January 1, 2016, retrieved 2016-09-02
http://www.fda.gov/MedicalDevices/ProductsandMedicalProcedures/DeviceApprovalsandClearances/510kClearances/

[FDA-CFR-21] U.S Food and Drug Administration: Medical Devices, Title 21, Part 806, Medical Devices: Reports of Corrections and Removals, retrieved 2016-09-06 *http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=806&showFR=1*

[FDA-CGMP] U.S. Food and Drug Administration: Current Good Manufacturing Practices (CGMPs), August 9, 2004, retrieved 2016-09-02 *http://www.fda.gov/food/guidanceregulation/cgmp/*

[FedRAMP] Federal Risk and Authorization Management Program (FedRAMP), retrieved 2016-09-05 *https://www.fedramp.gov/about-us/about/*

[FIPS-140-2] Federal Information Processing Standards Publication: FIPS PUB 140-2, Security Requirements for Cryptographic Modules, May 25, 2001, retrieved 2016-09-02 *http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf* from *http://csrc.nist.gov/groups/STM/cmvp/standards.html*

[FTC-FIPPS] Federal Trade Commission's Fair Information Practice Principles (FIPPs), FTC *https://www.ftc.gov/reports/privacy-online-fair-information-practices-electronic-marketplace-federal-trade-commission*

[Fuji-MAT] Fujitu Group Information Security Report: Mutual Authentication Technology between Devices in the IoT Era, Research and Development into Security Technology for Supporting a Safe Lifestyle, 2015 *https://www.fujitsu.com/global/documents/about/resources/reports/securityreport/2015-securityreports/security2015-09-e.pdf*

[GloP-TEE] GlobalPlatform made simple guide: Trusted Execution Environment (TEE) Guide (Media & Resource Center), retrieved 2016-09-02 *http://www.globalplatform.org/mediaguidetee.asp*

[HHS-HIPAA] U.S. Department of Health & Human Services: Heath Information Privacy, retrieved 2016-09-02 *http://www.hhs.gov/hipaa*

[HHS-HITECH] U.S. Department of Health & Human Services: Heath Information Privacy, retrieved 2016-09-05 *http://www.hhs.gov/hipaa/for-professionals/special-topics/HITECH-act-enforcement-interim-final-rule/index.html*

[HIT-ISA-65-5] Hitachi Review Vol. 65 No. 5, International Standardization Activities for Hitachi System Security Concept and Social Infrastructure Security Based on It, 2016 *http://www.hitachi.com/rev/pdf/2016/r2016_05_109.pdf*

[IDC-IOT2015] IDC Research Inc: Explosive Internet of Things Spending to Reach $1.7 Trillion in 2020, According to IDC, Pres Release, June 2, 2015, retrieved 2016-09-02 *http://www.idc.com/getdoc.jsp?containerId=prUS25658015*

[IDESG-IDEF] The Identity Ecosystem Steering Group (IDESG): Functional Model Representation of the Identity Ecosystem, version 1.0, 2014, retrieved 2016-09-05
https://workspace.idesg.org/kws/public/download/81/IDEF-Functional-Model-v1.0.pdf
from
https://workspace.idesg.org/kws/public/documents

[IEA-MEPS] International Energy Agency: Minimum Energy Performance Standards (MEPS), retrieved 2016-09-05
http://www.iea.org/policiesandmeasures/pams/newzealand/name-21549-en.php

[IEC-61508] International Electrotechnical Commission: IEC 61508:2010 CMV, Commented version, Functional safety of electrical/electronic/programmable electronic safety-related systems, 2010, retrieved 2016-09-05
https://webstore.iec.ch/publication/22273
from
http://www.iec.ch/functionalsafety/explained/

[IEC-61511] International Electrotechnical Commission: IEC 61511:2016, Functional safety - Safety instrumented systems for the process industry sector - Part 1: Framework, definitions, system, hardware and application programming requirements, 2016, retrieved 2016-09-05
https://webstore.iec.ch/publication/24241

[IEC-61513] International Electrotechnical Commission: IEC 61513:2011, Nuclear power plants - Instrumentation and control important to safety - General requirements for systems, 2011, retrieved 2016-09-05
https://webstore.iec.ch/publication/24241

[IEC-62279] International Electrotechnical Commission: IEC 62279:2015, Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems, 2015, retrieved 2016-09-05
https://webstore.iec.ch/publication/5532

[IEC-62351-7] International Electrotechnical Commission: IEC TS 62351-7:2010, Power systems management and associated information exchange - Data and communications security - Part 7: Network and system management (NSM) data object models, 2010, retrieved 2016-09-05
https://webstore.iec.ch/publication/6910

[IEC-62443-11] International Electrotechnical Commission: IEC TS 62443-1-1:2009, Industrial communication networks - Network and system security - Part 1-1: Terminology, concepts and models, 2009, retrieved 2016-09-02 *https://webstore.iec.ch/publication/7029*

[IEC-62443-21] International Electrotechnical Commission: IEC 62443-2-1:2010, Industrial communication networks - Network and system security - Part 2-1: Establishing an industrial automation and control system security program, 2010, retrieved 2016-09-02 *https://webstore.iec.ch/publication/7030*

[IEC-62443-23] International Electrotechnical Commission: IEC TR 62443-2-3:2015, Security for industrial automation and control systems - Part 2-3: Patch management in the IACS environment, 2015, retrieved 2016-09-02 *https://webstore.iec.ch/publication/22811*

[IEC-62443-24] International Electrotechnical Commission: IEC 62443-2-4:2015, Security for industrial automation and control systems - Part 2-4: Security program requirements for IACS service providers, 2015, retrieved 2016-09-02 *https://webstore.iec.ch/publication/22810*

[IEC-62443-3] International Electrotechnical Commission: IEC PAS 62443-3:2008, Security for industrial process measurement and control - Network and system security, 2008, retrieved 2016-09-02 *https://webstore.iec.ch/publication/7561*

[IEC-62443-31] International Electrotechnical Commission: IEC TR 62443-3-1:2009, Industrial communication networks - Network and system security - Part 3-1: Security technologies for industrial automation and control systems, 2009, retrieved 2016-09-02 *https://webstore.iec.ch/publication/7031*

[IEC-62443-33] International Electrotechnical Commission: IEC 62443-3-3:2013, Industrial communication networks - Network and system security - Part 3-3: System security requirements and security levels, 2013, retrieved 2016-09-02 *https://webstore.iec.ch/publication/7033*

[IECEE] IEC System for Conformity Assessment Schemes for Electrotechnical Equipment and Components, retrieved 2016-09-05 *http://www.iecee.org/*

[IEC-FOTF] International Electrotechnical Commission Whitepaper: Factory of the Future, 2015, retrieved 2016-09-02 *http://www.iec.ch/whitepaper/pdf/iecWP-futurefactory-LR-en.pdf*

[IEEE-1686] IEEE Standards Association: 1686-2013 - IEEE Standard for Intelligent Electronic Devices Cyber Security Capabilities, retrieved 2016-09-05 *https://standards.ieee.org/findstds/standard/1686-2013.html*

[IEEE-802] IEEE Standards Association: IEEE 802, Overview & Architecture, retrieved 2016-09-05
http://standards.ieee.org/about/get/802/802.html

[IEEE-C37-240] IEEE Standards Association: C37.240-2014 - IEEE Standard Cybersecurity Requirements for Substation Automation, Protection, and Control Systems, retrieved 2016-09-05
https://standards.ieee.org/findstds/standard/C37.240-2014.html

[IETF-RFC2119] Internet Engineering Task Force (IETF), Bradner, S.: RFC 2119, Key words for use in RFCs to Indicate Requirement Levels, 1997
http://ietf.org/rfc/rfc2119.txt

[IETF-RFC2818] Internet Engineering Task Force (IETF), Rescorla E: RFC 2818, HTTP Over TLS, 2000-May
http://ietf.org/rfc/rfc2818.txt

[IETF-RFC5246] Internet Engineering Task Force (IETF), Dierks T., Rescorla E.: RFC 5246, The Transport Layer Security (TLS) Protocol, 2008
https://tools.ietf.org/html/rfc5246

[IETF-RFC5480] Internet Engineering Task Force (IETF), Turner S., Brown D., Yiu K., Housley R., Polk T.: RFC 5480, Elliptic Curve Cryptography Subject Public Key Information, 2009-Mar
https://tools.ietf.org/html/rfc5480

[IETF-RFC6090] Internet Engineering Task Force (IETF), McGrew D, Igoe K., Salter M: RFC 6090, Fundamental Elliptic Curve Cryptography Algorithms, 2011-Feb
https://tools.ietf.org/html/rfc6090

[IETF-RFC6347] Internet Engineering Task Force (IETF), Rescorla E, Modadugu N: RFC 6347, Datagram Transport Layer Security Version 1.2, 2012-Jan
https://tools.ietf.org/html/rfc6347

[IETF-RFC7027] Internet Engineering Task Force (IETF), Merkle J., Lochter M.: RFC 7027, Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS), 2013-Oct
https://tools.ietf.org/html/rfc7027

[IETF-RFC7230] Internet Engineering Task Force (IETF), Fielding E., Reschke E: RFC 7230, Elliptic Curve Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing, 2014-Jun
https://tools.ietf.org/html/rfc7230

[IIC-IICF2017] Industrial Internet Consortium (IIC): Industrial Internet of Things, Volume G5: Connectivity Framework (IICF), 2017
(not yet published)

[IIC-IIRA2016] Industrial Internet Consortium (IIC): Industrial Internet of Things, Volume G1: Reference Architecture, version 1.71, 2016-August-12
http://www.iiconsortium.org/IIRA.htm

[IIC-IIV2016] Industrial Internet Consortium (IIC): Industrial Internet of Things, Volume G8: Vocabulary, version 2.0, 2016-August-30
http://www.iiconsortium.org/vocab/index.htm

[IMO-MARPOL] International Maritime Organization: International Convention for the Prevention of Pollution from Ships (MARPOL), retrieved 2016-09-05
http://www.imo.org/en/About/Conventions/ListOfConventions/Pages/International-Convention-for-the-Prevention-of-Pollution-from-Ships-(MARPOL).aspx

[Ind4.0-SecId] Industrie 4.0, Working Paper, Technical Overview: Secure Identities, 2016, retrieved 2016-09-05
https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/secure-identities.pdf
from
http://www.plattform-i40.de/I40/Navigation/EN/InPractice/Online-Library/online-library.html

[Intel-AMT] Intel Corp: Getting Started with Intel Active Management Technology (AMT), retrieved 2016-09-02
https://software.intel.com/en-us/articles/getting-started-with-intel-active-management-technology-amt

[Intel-SGX] Intel Corp: Intel Software Guard Extensions (Intel SGX), retrieved 2016-09-02
https://software.intel.com/en-us/sgx

[ISA-99] International Society of Automation (ISA): ISA99, Industrial Automation and Control Systems Security, retrieved 2016-09-05
https://www.isa.org/isa99/

[ISACA-COBIT] ISACA: Control Objectives for Information and Related Technologies (COBIT), COBIT 4.1: Framework for IT Governance and Control, retrieved 2016-09-05
http://www.isaca.org/knowledge-center/cobit/pages/overview.aspx

[ISO-13849] International Organization for Standardization: ISO/IEC 13849-1:2015, Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design, 2015, retrieved 2016-09-05
https://www.iso.org/obp/ui/#iso:std:iso:13849:-1:ed-3:v1:en

[ISO-15408] International Organization for Standardization: ISO/IEC 15408-1:2009, Information technology—Security techniques—Evaluation criteria for IT security—Part 1: Introduction and general model, 2009, retrieved 2016-09-05
https://webstore.iec.ch/publication/10255

[ISO-19770] International Organization for Standardization: ISO/IEC 19770-5:2015: Information Technology–IT asset management, part 5: Overview and vocabulary
http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=68291

[ISO-20008] International Organization for Standardization: ISO/IEC 20008-1: Information technology—Security techniques—Anonymous digital signatures, part 1 and part 2, 2013
https://webstore.iec.ch/preview/info_isoiec20008-1%7Bed1.0%7Den.pdf and *https://webstore.iec.ch/preview/info_isoiec20008-2%7Bed1.0%7Den.pdf*

[ISO-20009] International Organization for Standardization: ISO/IEC 20009-2: Information technology—Security techniques—Anonymous entity authentication, part 1: General and part 2: Mechanisms based on signatures using a group public key, 2013
https://webstore.iec.ch/preview/info_isoiec20009-1%7Bed1.0%7Den.pdf and *https://webstore.iec.ch/preview/info_isoiec20009-2%7Bed1.0%7Den.pdf*

[ISO-20243] International Organization for Standardization: ISO/IEC 20243: Information Technology—Open Trusted Technology ProviderTM Standard (O-TTPS)—Mitigating maliciously tainted and counterfeit products, 2015-Sep-15
https://webstore.iec.ch/preview/info_isoiec20243%7Bed1.0%7Den.pdf

[ISO-24760-1] International Organization for Standardization: ISO/IEC 24760-1:2011: Information Technology—Security techniques—A framework for identity management, 2011-12-15, retrieved 2016-09-02
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=57914

[ISO-26262-1] International Organization for Standardization: ISO 26262-1:2011: Road vehicles—Functional safety—Part 1: Vocabulary, 2011, retrieved 2016-09-02
http://www.iso.org/iso/catalogue_detail?csnumber=43464

[ISO-27000] International Organization for Standardization: ISO 27000:2016: Information technology—Security technique—Information security management systems—Overview and vocabulary, 2016, retrieved 2016-09-02
http://www.iso.org/iso/catalogue_detail?csnumber=66435

[ISO-27001] International Organization for Standardization: ISO 27001:2013: Information technology—Security technique—Information security management systems—Requirements, 2013, retrieved 2016-09-02
http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=54534

[ISO-27017] International Organization for Standardization: ISO/IEC 27017:2015: Information technology—Security technique—Code of practice for information security controls based on ISO/IEC 27002 for cloud services, 2015, retrieved 2016-09-05
http://www.iso.org/iso/catalogue_detail?csnumber=43757

[ISO-27018] International Organization for Standardization: ISO/IEC 27018:2014: Information technology—Security technique—Code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors, 2014, retrieved 2016-09-05
http://www.iso.org/iso/catalogue_detail.htm?csnumber=61498

[ISO-27031] International Organization for Standardization: ISO 27031:2011: Information technology—Security technique—Guidelines for information and communication technology readiness for business continuity, 2011, retrieved 2016-09-02
http://www.iso.org/iso/catalogue_detail?csnumber=44374

[ISO-27036-1] International Organization for Standardization: ISO 27036-4: Information technology—Security technique—Information security for supplier relationships—Part 1: Overview and concepts, retrieved 2014-04-01
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=59648

[ISO-27036-4] International Organization for Standardization: ISO 27036-4: Information technology—Security technique—Information security for supplier relationships—Part 4: Guidelines for security of cloud services (under development), retrieved 2016-09-02
http://www.iso.org/iso/catalogue_detail.htm?csnumber=59689

[ISO-29100] International Organization for Standardization: ISO/IEC 29100:2011: Information technology—Security technique—Privacy framework, 2011, retrieved 2016-09-05
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=45123

[ISO-29101] International Organization for Standardization: ISO/IEC 29101:2013: Information technology—Security technique—Privacy architecture framework, 2013, retrieved 2016-09-05
http://www.iso.org/iso/catalogue_detail.htm?csnumber=45124

[ISO-29115] International Organization for Standardization: ISO/IEC 29115:2013: Information technology—Security techniques—Entity authentication assurance framework, 2013
https://webstore.iec.ch/preview/info_isoiec29115%7Bed1.0%7Den.pdf

[ISO-29134] International Organization for Standardization: ISO/IEC DIS 29134: Information technology—Security technique—Privacy impact assessment—Guidelines, 2016-07-08, retrieved 2016-09-05 *http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=62289*

[ISO-29190] International Organization for Standardization: ISO/IEC 29190:2015: Information technology—Security technique—Privacy capability assessment model, 2015, retrieved 2016-09-05 *http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=45269*

[LinuxC-LXC] Linux Containers: What's LXC, retrieved 2016-09-02 *https://linuxcontainers.org/lxc/*

[McDer1999] McDermott, John and Fox, Chris: Using Abuse Case Models for Security Requirements Analysis, Department of Computer Science, James Madison University, Harrisonburg, Virginia 222807, 1999 *http://www.acsa-admin.org/1999/papers/wed-b-1030-john.pdf*

[MISRA-C] The Motor Industry Software Reliability Association (MISRA): Guidelines for the Use of the C Language in Critical Systems, ISBN 978-1-906400-10-1 (paperback), ISBN 978-1-906400-11-8 (PDF), March 2013 retrieved 2016-09-05 from *https://www.misra.org.uk/Publications/tabid/57/Default.aspx*

[MIT-Kerb] Massachusetts Institute of Technology (MIT): Kerberos: The Network Authentication Protocol. MIT, 2016-Apr-20 *http://web.mit.edu/kerberos/*

[MIT-PUF] Massachusetts Institute of Technology (MIT): Physical Unclonable Functions and Applications, *http://people.csail.mit.edu/rudolph/Teaching/Lectures/Security/Lecture-Security-PUFs-2.pdf*

[Modbus] Modbus Organization: Modbus Protocol, retrieved 2016-09-05 *http://www.modbus.org/specs.php*

[MQTT] MQ Telemetry Transport (MQTT), retrieved 2016-09-05 *http://www.mqtt.org*

[MS-STRIDE] Microsoft: Commerce Server 2002, The STRIDE Threat Model, 2005. Accessed 2016-07-11 *https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx*

[MS-STRIDE-IOT] Internet of Things Security Architecture: Security in IoT, 2016-June-3 *https://azure.microsoft.com/en-us/documentation/articles/iot-hub-security-architecture/*

[NASA-CR2015] National Aeronautics and Space Administration (NASA) Langley Research Center Report – NASA/CR_2015-218678, retrieved 2016-09-05 *http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20150017614.pdf* from *http://ntrs.nasa.gov/search.jsp?R=20150017614*

[NEMA-CPSP] National Electrical Manufacturers Association (NEMA): Supply Chain Best Practices (NEMA CPSP 1-2015), June 25, 2015, retrieved 2016-09-05 *https://www.nema.org/Standards/Pages/Supply-Chain-Best-Practices.aspx#download* from *https://www.nema.org/Standards/Pages/Supply-Chain-Best-Practices.aspx*

[NERC-CIP] North American Electric Reliability Corporation (NERC): CIP Standards *http://www.nerc.com/pa/Stand/Pages/CIPStandards.aspx*

[NERC-CIP-002] North American Electric Reliability Corporation (NERC): CIP Standard CIP-002-3 *http://www.nerc.com/files/cip-002-3.pdf*

[NIST-500-291] National Institute of Standards and Technology (NIST): NIST Cloud Computing Standards Roadmap, Special Publication 500-291, version 2, July 2013, retrieved 2016-09-02 *https://www.nist.gov/sites/default/files/documents/itl/cloud/NIST_SP-500-291_Version-2_2013_June18_FINAL.pdf*

[NIST-500-292] National Institute of Standards and Technology (NIST): NIST Cloud Computing Standards Roadmap, Special Publication 500-292, January 2014 *(not available as PDF download but as book ISBN-13-978-1495323461)*

[NIST-500-293] National Institute of Standards and Technology (NIST): US Government Cloud Computing Technology Roadmap, Volume I, release 1.0 (Draft), Special Publication 500-293, draft, November 2011, retrieved 2016-09-02 *https://www.nist.gov/sites/default/files/documents/itl/cloud/SP_500_293_volumeI-2.pdf*

[NIST-7608] MIST Interagency Report 7608, Software Assurance Using Structure Assurance Case Models, NIST May 2009 *http://nvlpubs.nist.gov/nistpubs/ir/2009/ir7608.pdf*

[NIST-800-32] Introduction to Public Key Technology and the Federal PKI Infrastructure, NIST Feb 2001 *http://csrc.nist.gov/publications/nistpubs/800-32/sp800-32.pdf*

[NIST-800-53] National Institute of Standards and Technology (NIST): Special Publication 800-53, Security and Privacy Controls for Federal Information Systems and Organizations, Revision 4, 2013 April, retrieved 2016-09-02 *http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf*

[NIST-800-57] National Institute of Standards and Technology (NIST): Recommendation for Key Management, Part 1: General, 2016
http://dx.doi.org/10.6028/NIST.SP.800-57pt1r4

[NIST-800-82] National Institute of Standards and Technology (NIST): Guide to Industrial Control Systems (ICS) Security, Special Publication 800-82, revision 2, 2015-May, retrieved at 2016-09-02
http://dx.doi.org/10.6028/NIST.SP.800-82r2

[NIST-800-144] National Institute of Standards and Technology (NIST): Guidelines on Security and Privacy in Public Cloud Computing, Special Publication 800-144, December 2011, retrieved 2016-09-02
http://dx.doi.org/10.6028/NIST.SP.800-144

[NIST-800-146] National Institute of Standards and Technology (NIST): Cloud Computing Synopsis and Recommendations, Special Publication 800-161, May 2012, retrieved 2016-09-02
http://dx.doi.org/10.6028/NIST.SP.800-161

[NIST-800-160] National Institute of Standards and Technology (NIST): Systems Security Engineering, Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems, Special Publication 800-160, second public draft, May 2016, retrieved 2016-09-02
http://csrc.nist.gov/publications/drafts/800-160/sp800_160_second-draft.pdf

[NIST-800-161] National Institute of Standards and Technology (NIST): Supply Chain Risk Management: Practices for Federal Information Systems and Organizations, Special Publication 800-161, April 2015, retrieved 2016-09-02
http://dx.doi.org/10.6028/NIST.SP.800-161

[NIST-800-183] National Institute of Standards and Technology (NIST): Networks of 'Things', Special Publication 800-183, July 2016, retrieved 2016-09-02
http://dx.doi.org/10.6028/NIST.SP.800-183

[NIST-CPS] National Institute of Standards and Technology (NIST): CPS PWG Cyber-Physical Systems (CPS) Framework Release 1.0, retrieved 2016-09-02
https://s3.amazonaws.com/nist-sgcps/cpspwg/files/pwgglobal/-CPS_PWG_Framework_for_Cyber_Physical_Systems_Release_1_0Final.pdf
from
https://pages.nist.gov/cpspwg/

[NIST-FFAQ] National Institute of Standards and Technology (NIST): Cybersecurity Framework Frequently Asked Questions, National Institute of Standards and Technology, retrieved 2016-09-02
http://www.nist.gov/cyberframework/cybersecurity-framework-faqs-relationship-between-the-framework-and-other-approaches-and-initiatives.cfm

[NIST-FICIC] National Institute of Standards and Technology (NIST): Framework for Improving Critical Infrastructure Cybersecurity, Version 1.0, National Institute of Standards and Technology, 2014-February-12
http://www.nist.gov/cyberframework/upload/cybersecurity-framework-021214.pdf from
http://www.nist.gov/cyberframework/

[NIST-KEYM] National Institute of Standards and Technology (NIST): Key Management. Computer Security Division, Computer Security Resource Center, 2016-Mar-16
http://csrc.nist.gov/groups/ST/toolkit/key_management.html

[NISTIR-7628] National Institute of Standards and Technology (NIST): Guidelines for Smart Grid Cybersecurity, Volume 1 - Smart Grid Cybersecurity Strategy, Architecture, and High-Level Requirements, September 2014, retrieved 2016-09-02
http://dx.doi.org/10.6028/NIST.IR.7628r1

[NISTIR-8062] Privacy Risk Management for Federal Information Systems, draft, May 2015, retrieved 2016-09-02
http://csrc.nist.gov/publications/drafts/nistir-8062/nistir_8062_draft.pdf

[NIST-SCAP] National Institute of Standards and Technology (NIST): The Security Content Automation Protocol (SCAP), retrieved 2016-09-02
https://scap.nist.gov/

[NRECA-Smpl] National Rural Electric Cooperative Association (NRECA): Cyber Security Template Scoring Worksheet - Sample Data (Michael E. Lynch), May 2014, retrieved 2016-09-02
http://groups.cooperative.com/smartgriddemo/public/CyberSecurity/Documents/CyberSecurityTemplateScoringWorksheet-SampleData.xlsm
from
https://groups.cooperative.com/smartgriddemo/public/CyberSecurity/Documents/Forms/AllItems.aspx

[NRECA-Tmpl] National Rural Electric Cooperative Association (NRECA): Cyber Security Template Scoring Worksheet (Michael E. Lynch), May 2014, retrieved 2016-09-02
https://groups.cooperative.com/smartgriddemo/public/CyberSecurity/Documents/CyberSecurityTemplateScoringWorksheet.xlsm
from
https://groups.cooperative.com/smartgriddemo/public/CyberSecurity/Documents/Forms/AllItems.aspx

[OASIS] Advanced Open Standards for the Information Society, retrieved 2016-09-05
https://www.oasis-open.org/

[OECD-PPTF] Organisation for Economic Co-operation and Development (OECD): Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, OECD 2013, retrieved 2016-09-07
http://www.oecd.org/sti/ieconomy/oecdguidelinesontheprotectionofprivacyandtransborderflowsofpersonaldata.htm

[OMG-DDS] Object Management Group: What is DDS? The Proven Data Connectivity Standard for the Internet of Things, retrieved 2016-09-05
http://portals.omg.org/dds/what-is-dds-3/

[OPC-classic] OPC Foundation: OPC Classic, retrieved 2016-09-05
https://opcfoundation.org/about/opc-technologies/opc-classic/

[OPC-NET] OPC Foundation: OPC .NET 3.0 (formerly known as OPC Express Interface (Xi)), retrieved 2016-09-05
http://www.opcconnect.com/xi.php

[OPC-UA] OPC Foundation: OPC Unified Architecture, retrieved 2016-09-05
https://opcfoundation.org/about/opc-technologies/opc-ua/

[OpenFog-Res] Open Fog Consortium: Resources, retrieved 2016-09-05
https://www.openfogconsortium.org/resources

[OWASP] Open Web Application Security Project (OWASP): The free and open software security community
http://www.owasp.org

[OWASP-IOT] Open Web Application Security Project (OWASP): The free and open software security community, Internet of Things Project
http://www.owasp.org/index.php/OWASP_Internet_of_Things_Project

[PCI-DSS] PCI Security, PCI Security Standards Council
https://www.pcisecuritystandards.org/pci_security/

[PI-pbus] PROFIBUS and PROFINET International (PI): PROFIBUS, retrieved 2016-09-05
http://www.profibus.com/technology/profibus/

[PI-pnet] PROFIBUS and PROFINET International (PI): PROFINET, retrieved 2016-09-05 *http://www.profibus.com/technology/profinet/*

[RTCA-DO-178B] Radio Technical Commission for Aeronautics (RTCA, Inc.): DO-178B Software Considerations in Airborne Systems and Equipment Certification, 1992 *http://www.rtca.org/store_product.asp?prodid=581*

[RTCA-DO-178C] Radio Technical Commission for Aeronautics (RTCA, Inc.): DO-178C Software Considerations in Airborne Systems and Equipment Certification, 2011 *http://www.rtca.org/store_product.asp?prodid=803*

[Ruan2014] Ruan, Xiaoyu: Platform Embedded Security Technology Revealed: Safeguarding the Future of Computing with Intel Embedded Security and Management Engine, 1st Edition, 2014, ISBN 978-1-430265719

[SAE] SAE (originally Society of Automotive Engineers), retrieved 2016-09-02 *http://www.sae.org*

[Saltzer1974] Saltzer, Jerome H and Schroeder, Michael D.: The Protection of Information in Computer Systems. Manuscript received October 11, 1974; revised April 17. 1975. Fourth ACM Symposium on Operating System Principles (October 1973). Revised version in Communications of the ACM 17, 7 (July 1974). *http://www.cs.virginia.edu/~evans/cs551/saltzer/*

[SANS-SSCS] Escal Institute of Advanced Technologies (SANS Institute): The State of Security in Control Systems Today, 2015-June *http://www.sans.org/reading-room/whitepapers/analyst/state-security-control-systems-today-36042*

[SarOxl] Sarbanes-Oxley Act, U.S. Securities and Exchange Commission, 2002 *https://www.sec.gov/about/laws.shtml#sox2002*

[Schneider1998] Schneider, Fred B. (Editor), Committee on Information Systems: Trustworthiness: Trust in Cyberspace, National Academic Press, Washington D.C., 1999 *http://www.nap.edu/catalog/6161/trust-in-cyberspace*

[SGIP-CoS] Smart Grid Interoperability Panel (SGIP): Catalog of Standards (CoS) *http://www.sgip.org/wp-content/uploads/SGIPs_Catalog_of_Standards.pdf* from *http://www.sgip.org/catalog-of-standards/*

[SNL2005] Framework for SCADA Security Policy, Sandia National Laboratories, 2015 *http://energy.gov/sites/prod/files/Framework%20for%20SCADA%20Security%20Policy.pdf*

[Sym-ECC] Symantec Corporation: Elliptic Curve Cryptography (ECC), Certificates Performance Analysis, whitepaper, May 2013, retrieved 2016-09-02 *https://www.symantec.com/content/en/us/enterprise/white_papers/b-wp_ecc.pdf*

[TCG] Trusted Computing Group (TCG), retrieved 2016-09-02 *http://www.trustedcomputinggroup.org*

[TCG-AG-ICS] Trusted Computing Group (TCG): Architects Guide: ICS Security Using TNC Technology, retrieved 2016-09-05 *http://www.trustedcomputinggroup.org/wp-content/uploads/ICS-Security-Using-TNC-Technology-Architects-Guide.pdf* from *http://www.trustedcomputinggroup.org/architects-guide-ics-security-using-tnc-technology/*

[TCG-AG-IoT] Trusted Computing Group (TCG): Architects Guide: IoT Security, retrieved 2016-09-05 *http://www.trustedcomputinggroup.org/wp-content/uploads/IOT_Security_Architects_Guide_TCG.pdf* from *http://www.trustedcomputinggroup.org/architects-guide-iot-security/*

[TCG-GS-IoT] Trusted Computing Group (TCG): Guidance for Securing IoT Using TCG Technology Reference Document, retrieved 2016-09-05 *http://www.trustedcomputinggroup.org/wp-content/uploads/TCG_Guidance_for_Securing_IoT_1_0r21.pdf* from *http://www.trustedcomputinggroup.org/guidance-securing-iot-using-tcg-technology-reference-document/*

[TCG-Spec] TCG Specification. Architecture Overview, Revision 1.4, 2nd August 2007, retrieved 2016-09-02 *https://www.trustedcomputinggroup.org/wp-content/uploads/TCG_1_4_Architecture_Overview.pdf*

[TCG-TPM] Trusted Computing Group (TCG): Trusted Platform Module (TPM), retrieved 2016-09-02 *http://www.trustedcomputinggroup.org/work-groups/trusted-platform-module*

[TCG-TPM-Spec] Trusted Computing Group (TCG): TPM Library Specification (TPM 2.0), retrieved 2016-09-02 *http://www.trustedcomputinggroup.org/tpm-library-specification/*

[UEFI] Unified Extensible Firmware Interface Forum (UEFI), specifications, retrieved 2016-09-02
http://www.uefi.org/specifications

[US-EU-Prv-Sh] United States, Department of Commerce: EU-U.S. Privacy Shield, retrieved 2016-09-05
https://www.commerce.gov/page/eu-us-privacy-shield

[UWAT-QC] University of Waterloo: Quantum computing 101, retrieved 2016-09-02
https://uwaterloo.ca/institute-for-quantum-computing/quantum-computing-101

[WASC] Web Application Security Consortium (WASC), retrieved 2016-09-02
http://www.webappsec.org

[XMPP] Extensible Messaging and Presence Protocol (XMPP), retrieved 2016-09-05
http://xmpp.org

INDEX

R

S

T