ICSA-500

ISA Security Compliance Institute — IIoT Component Security Assurance –

Selected commonly accepted security practices

Version 1.1

January 2023

A. DISCLAIMER

ASCI and all related entities, including the International Society of Automation (collectively, “ASCI”) provide all materials, work products and, information (‘SPECIFICATION’) AS IS, WITHOUT WARRANTY AND WITH ALL FAULTS, and hereby disclaim all warranties and conditions, whether express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of reliability or availability, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses, and of lack of negligence, all with regard to the SPECIFICATION, and the provision of or failure to provide support or other services, information, software, and related content through the SPECIFICATION or otherwise arising out of the use of the SPECIFICATION. Also, there is no warranty or condition of title, quiet enjoyment, quiet possession, correspondence to description, or non-infringement with regard to the SPECIFICATION.

Without limiting the foregoing, ASCI disclaims all liability for harm to persons or property, and users of this SPECIFICATION assume all risks of such harm.

In issuing and making the SPECIFICATION available, ASCI is not undertaking to render professional or other services for or on behalf of any person or entity, nor is ASCI undertaking to perform any duty owed by any person or entity to someone else. Anyone using this SPECIFICATION should rely on his or her own independent judgment or, as appropriate, seek the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

B. EXCLUSION OF INCIDENTAL, CONSEQUENTIAL AND CERTAIN OTHER DAMAGES

To the maximum extent permitted by applicable law, in no event shall ASCI or its suppliers be liable for any special, incidental, punitive, indirect, or consequential damages whatsoever (including, but not limited to, damages for loss of profits or confidential or other information, for business interruption, for personal injury, for loss of privacy, for failure to meet any duty including of good faith or of reasonable care, for negligence, and for any other pecuniary or other loss whatsoever) arising out of or in any way related to the use of or inability to use the SPECIFICATION, the provision of or failure to provide support or other services, information, software, and related content through the SPECIFICATION or otherwise arising out of the use of the SPECIFICATION, or otherwise under or in connection with any provision of this SPECIFICATION, even in the event of the fault, tort (including negligence), misrepresentation, strict liability, breach of contract of ASCI or any supplier, and even if ASCI or any supplier has been advised of the possibility of such damages.

C. OTHER TERMS OF USE

Except as expressly authorized by prior written consent from the Automation Standards Compliance Institute, no material from this document owned, licensed, or controlled by the Automation Standards Compliance Institute may be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, except for non-commercial use only, provided that you keep intact all copyright and other proprietary notices. Modification of the materials or use of the materials for any other purpose, such as creating derivative works for commercial use, is a violation of the Automation Standards Compliance Institute’s copyright and other proprietary rights.

Revision history

version	date	changes
1.1	2023.01.24	Initial version published to https://www.isasecure.org/

Contents

Practices

FOREWORD

This is one of a series of documents that defines the ISASecure® ICSA (Industrial Internet of Things Component Security Assurance) certification program for IIoT devices and gateways. ISASecure ICSA is developed and managed by the industry consortium ISA Security Compliance Institute (ISCI). This document provides guidance for interpretation of selected functional security requirements for certification found in other ICSA specifications, that require conformance to “commonly accepted practices.” A description of the program and the current list of documents related to ISASecure ICSA, as well as other ISASecure certification programs, can be found on the web site https://www.isasecure.org/.

1 Scope

1.1 Background

The ISASecure® certification program has been developed by an industry consortium called the ISA Security Compliance Institute (ISCI) with a goal to accelerate industry wide improvement of cyber security for Industrial Automation and Control Systems (IACS), including IIoT systems (Industrial Internet of Things). The ISCI ISASecure ICSA certification program (IIoT Component Security Assurance) achieves this goal by offering a common standards-based, industry-recognized set of component and process requirements that drive IIoT component security, simplifying procurement for asset owners, and component assurance for product suppliers. Components in scope for ICSA are IIoT devices and IIoT gateways as defined in Section 3.1. An IIoT component that is certified to meet the ICSA requirements can display the ISASecure “Certified IIoT Component” symbol.

The ICSA certification scheme ICSA-100 [ICSA-100] provides an overview of the ICSA certification, and describes the relationship between ICSA and the standard IEC 62443-4-2 [IEC 62443-4-2]. IEC 62443-4-2 requires functional capabilities as well as conformance with lifecycle process requirements in IEC 62443-4-1 [IEC 62443-4-1], including handling of security updates after product release. The reader will find it helpful to read ICSA-100 as background context for the present document. In summary, ICSA incorporates a relatively small number of exceptions and extensions to IEC 62443-4-2 to address the IIoT component environment. The document ISASecure-119 [ISASecure-119] provides a detailed comparison of ICSA to the ISASecure CSA program, which assesses conformance to IEC 62443-4-2. ISASecure-119 summarizes the ICSA exceptions and extensions to IEC 62443-4-2.

1.2 Purpose

Figure 1 below illustrates the context of the present document. Practices are provided in ICSA-500, as guidance for interpreting selected ICSA certification requirements found in the ICSA specifications [ICSA-311] and [ISDLA-312] (one requirement, not shown in the figure). ICSA certification criteria include all requirements found in those specifications, whether or not the present document provides further guidance for their interpretation.

In particular, in the ICSA specifications, a number of requirements and validation activities performed by the certifier to assess compliance to requirements, use the phrase “commonly accepted practices,” or “commonly accepted practices for IIoT.” The purpose of the present document is to provide guidance on how those phrases may be interpreted for these requirements, in the context of an ICSA certification, for both assessors and product suppliers. If a requirement for ICSA refers to conformance with commonly accepted practices, then the present document provides supporting lists of examples of such practices, as well as related references. Related references listed for a practice, either serve to support the assertion that the practice described is “commonly accepted,” or provide further technical information about possible implementations of the practice. A reference to a specific implementation or supplier in this document does not indicate that ISCI has approved or endorsed that implementation or supplier.

At the time of publication of this document, the practices described here are judged by ISCI to be commonly accepted to use for the types of IIoT components eligible for ICSA certification. However, commonly accepted practices are expected to evolve rapidly in the IIoT space. The information in this document is provided as a resource to improve the security posture of ICSA certified components, by offering guidance more granular than would typically be appropriate in a standard. It is provided separately from the other specifications for ICSA, to support innovation and the potential for corresponding updates to this document, while not affecting other ICSA specifications. The most current version of this document can be found at https://www.ISASecure.org.

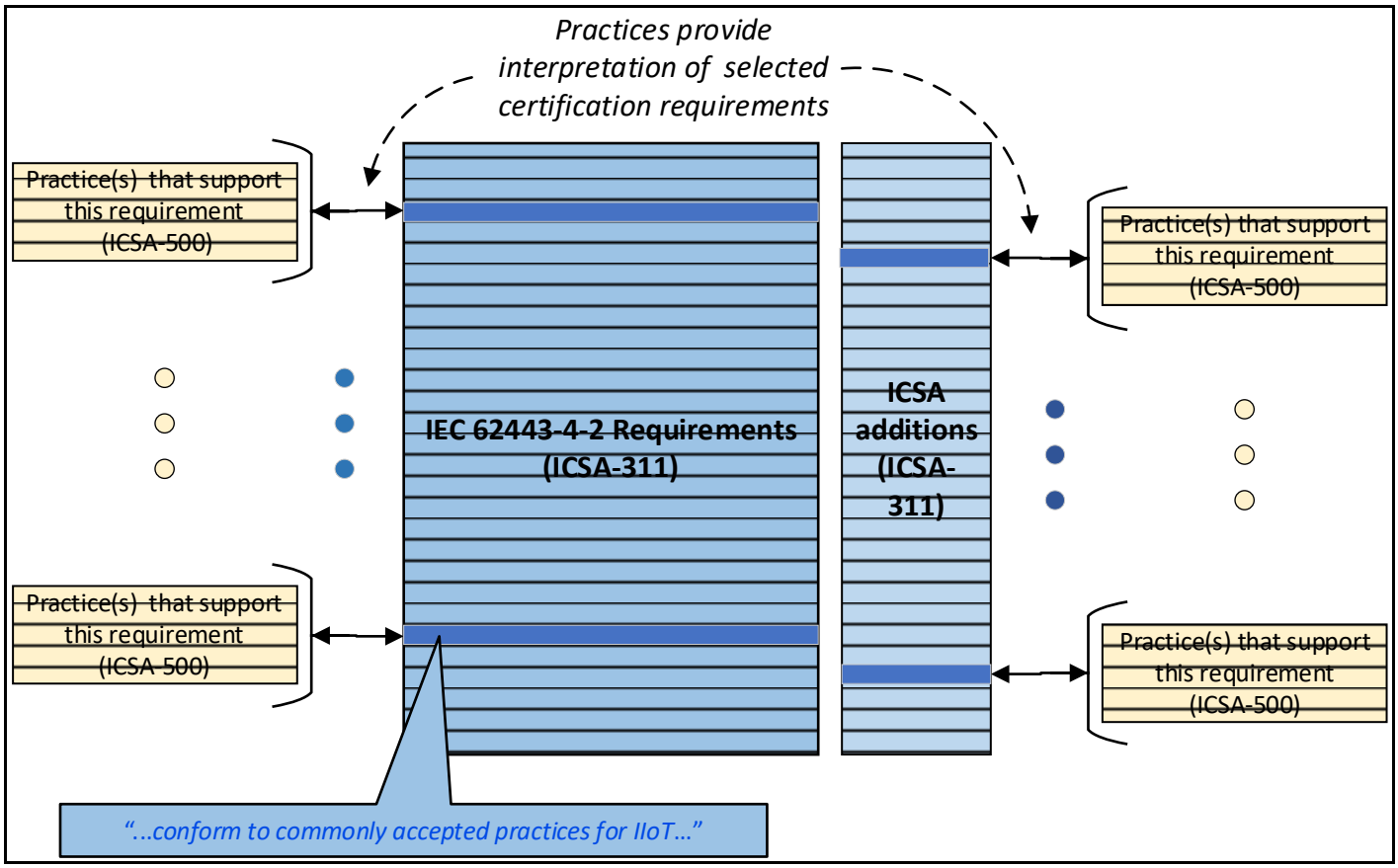

Figure 1. Relationship of ICSA-500 commonly accepted practices to ICSA-311 certification requirements

Where specific implementations are provided in the present document as examples of commonly accepted practices, management of any vulnerabilities in these implementations (as for any part of a product) is subject to defect management requirements in IEC 62443-4-1. Conformance of the product development process to these IEC 62443-4-1 requirements is verified for initial ICSA certification under the Security Development Artifacts (SDA) audit for requirements under the Defect Management practice (DM) in [ISDLA-312]. After initial certification, conformance is verified under the ongoing Security Management Audit (SMA) process defined in [ICSA-301].

1.3 Selection of requirements and associated practices

The following paragraphs describe how topics were selected for coverage in the present document.

The topics addressed in the present document are topics addressed by a subset of the certification requirements in the ICSA specifications ICSA-311 and ISDLA-312, that specifically use the phrase "commonly accepted practices" or "commonly accepted practices for IIoT." The phrase may be used in either the expression of the requirement or in the description of its associated validation activity. The present document does not cover all potential topics that might be considered commonly accepted practices for IIoT components.

ICSA requirements were selected to discuss in this document, where offering additional information was judged to ultimately provide significant value for the users of ICSA certified products. Some ICSA certification requirements that use one of the "commonly accepted" phrases do not have associated practices in this document. The appendix in Section 5 provides a list of those ICSA requirements that have associated practices in the present document, and a mapping to those practices. This mapping is represented by the example double-headed arrows in Figure 1.

This document is intended as a supplement to the ICSA specifications, and as such is not a comprehensive stand-alone reference for IIoT component security. In particular, requirements already found in IEC 62443-4-2 and incorporated in the ICSA-311 specification, are not duplicated in the present document. For example, IEC 62443-4-2 has requirements about auditable events (Requirement CR 2.8). The present document in 4.9 and 4.10 adds practices about auditing specific types of events, and protocols for reporting events, adding detail to related requirements found in IEC 62443-4-2. However, the present document does not repeat requirements related to auditable events found in IEC 62443-4-2 and ICSA-311.

In summary, the present document addresses a subset of topics related to commonly accepted practices for IIoT security, selected due to the value provided for ICSA certified components. For those topics addressed, the document describes practices in a level of detail not already provided in other ICSA specifications.

1.4 How to apply this document

1.4.1 Document structure

This document first describes practices by topic area, followed by the list of ICSA certification requirements associated with those practices, then followed by references. Links to the references provided were tested as of January 3, 2023. A mapping from ICSA certification requirements to commonly accepted practices, is found in the Appendix Section 5.

1.4.2 Impact of conformance to practices on certification

This document provides an interpretation of “commonly accepted” in the context of specific ICSA requirements, which will be acceptable for certification. While these practices are *acceptable* in the context of ICSA certification, they are not mandatory. The present document is a *source* for commonly accepted practices, and does not rule out use of other sources. Also as stated in [ICSA-311], “Practices may be used that are demonstrated as effective or more effective than those conforming to commonly accepted IIoT practice. Examples of such "demonstration" are: proven use in other domains, or recommendation by a recognized authority.”

In summary, to obtain ICSA certification it is not required to conform to the particular commonly accepted practices described here. Other practices may be used if they can be either shown to be commonly accepted, or demonstrated as effective or more effective than those commonly accepted.

2 References

This section contains general references. References associated with specific ICSA requirements and related practices are presented in Section 4 together with those requirements and practices.

2.1 ISASecure specifications

2.1.1 ICSA certification program

[ICSA-100] *ISCI IIoT Component Security Assurance – ISASecure Certification Scheme*, as specified at https://www.isasecure.org/

[CSA-100] *ISCI Component Security Assurance – ISASecure Certification Scheme*, as specified at https://www.isasecure.org/

[ISASecure-119] *ISA Security Compliance Institute – Comparison of ICSA and CSA certifications*, available at https://www.ISASecure.org

2.1.2 Technical specifications

NOTE 1 The following document is the overarching technical specification for ISASecure ICSA certification.

[ICSA-300] *ISCI IIoT Component Security Assurance – ISASecure Certification Requirements,* as specified at https://www.isasecure.org/

NOTE 2 As explained in 1.2-1.4, some requirements in the following two ICSA technical specifications specify the use of “commonly accepted” practices.

[ICSA-311] *ISCI IIoT Component Security Assurance – Functional security assessment for IIoT components,* as specified at https://www.isasecure.org/

[ISDLA-312] *ISCI Security Development Lifecycle Assurance – Security development lifecycle assessment for ICSA*, as specified at https://www.ISASecure.org

2.2 IACS security standards

NOTE 1 [ICSA-100] describes the relationship of ISASecure ICSA to the ANSI/ISA/IEC 62443 series of standards.

NOTE 2 The following pairs of references that have the same document number 62443-m-n, provide the same technical standard, as published by the organizations ANSI/ISA and IEC.

[ANSI/ISA-62443-1-1] ANSI/ISA-62443-1-1 *(99.01.01)-2007 Security for industrial automation and control systems Part 1-1: Terminology, concepts and models*

[IEC 62443-1-1] IEC TS 62443-1-1:2009 *Industrial communication networks – Network and system security - Part 1-1: Terminology, concepts and models*

[ANSI/ISA-62443-4-1] ANSI/ISA-62443-4-1-2018 *Security for industrial automation and control systems Part 4-1: Secure product development lifecycle requirements*

[IEC 62443-4-1] IEC 62443-4-1:*2018 Security for industrial automation and control systems Part 4-1: Secure product development lifecycle requirements*

[ANSI/ISA-62443-4-2] ANSI/ISA-62443-4-2-2018 *Security for industrial automation and control systems Part 4-2: Technical security requirements for IACS components*

[IEC 62443-4-2] IEC 62443-4-2:2019 *Security for industrial automation and control systems Part 4-2: Technical security requirements for IACS components*

3 Definitions and abbreviations

3.1 Definitions

3.1.1
artifact
tangible output from the application of a specified method that provides evidence of its application

NOTE Examples of artifacts for secure product development methods are a threat model document, a security requirements document, meeting minutes, internal test results.

3.1.2
asset owner
individual or company responsible for one or more IACS

NOTE 1 Used in place of the generic term end user to provide differentiation.

NOTE 2 This includes the components that are part of the IACS.

NOTE 3 In the context of this document, an asset owner also includes the operator of the IACS.

3.1.3
bare metal hypervisor
hypervisor that runs directly on component hardware with no hosting OS

NOTE 1 Source NIST SP 800-125.

NOTE 2 Another term for this concept is “Type 1 hypervisor,” as seen for example at https://techterms.com/definition/hypervisor.

3.1.4
capability security level
level that indicates capability of meeting a security level natively without additional compensating countermeasures when properly configured and integrated

3.1.5
certifier
chartered laboratory, which is an organization that is qualified to certify products or supplier development processes as ISASecure

NOTE This term is used when a simpler term that indicates the role of a “chartered laboratory” is clearer in a particular context.

3.1.6
certificate
set of data that uniquely identifies an entity, contains the entity’s public key and possibly other information, and is digitally signed by a trusted party, thereby binding the public key to the entity. Additional information in the certificate could specify how the key is used and its validity period.

NOTE Source NIST SP 800-57.

3.1.7
certification
third party attestation related to products, processes, or persons that conveys assurance that specified requirements have been demonstrated

NOTE Here, this refers to either a successful authorized evaluation of a product or a process to ISASecure criteria. This outcome permits the product supplier or organization performing the process to advertise this achievement in accordance with certification program guidelines.

3.1.8
certification scheme
overall definition of and process for operating a certification program

3.1.9
certified component
component that has undergone an evaluation by a chartered laboratory, has met the ISASecure ICSA criteria and has been granted certified status by the chartered laboratory

3.1.10
chartered laboratory
organization chartered by ASCI to evaluate products or development processes under one or more ISASecure certification programs and to grant certifications under one or more of these programs

NOTE A chartered laboratory is the conformity assessment body for the ISASecure certification programs.

3.1.11
compartmentalization
use of any method or technology to separate multiple functions during execution, where separation limits their interactions to those intended

NOTE Examples of compartmentalization methods are containerization, virtual machines, hardware separation (by chip or board), enforced memory allocation, software-based microsegmentation.

3.1.12
conformity assessment
demonstration that specified requirements relating to a product, process, system, person, or body are fulfilled

3.1.13
component
entity belonging to an IACS that exhibits the characteristics of one or more of a host device, network device, software application, or embedded device

NOTE Source [IEC 62443-4-2].

3.1.14
conformity assessment body
body that performs conformity assessment services and that can be the object of accreditation

NOTE This is an ISO/IEC term and concept. For ISASecure ICSA, the conformity assessment body is a chartered laboratory.

3.1.15
container
method to package an application and its dependencies so it can be run in many environments

3.1.16
control function
capability to change the operation of a process and/or equipment associated with a process, which may be in response to input signals from the process, its associated equipment, other programmable systems and/or an operator

NOTE This definition is adapted from the definition of basic process control system, in IEC 62443-1-2.

3.1.17
data in use
data in non-persistent storage in a physical or logical location in the device architecture, when the data is being used by the component or its use is imminent

NOTE Typically this location is in RAM, or inside the CPU.

3.1.18
embedded device
special purpose device running embedded software designed to directly monitor, control or actuate an industrial process

NOTE 1 Attributes of an embedded device are: no rotating media, limited number of exposed services, programmed through an external interface, embedded OS or firmware equivalent, real-time scheduler, may have an attached control panel, may have a communications interface. Examples are: PLC, field sensor devices, SIS controller, DCS controller.

NOTE 2 Source [IEC 62443-4-2].

3.1.19
essential function
function or capability that is required to maintain health, safety, the environment, and availability for the equipment under control

NOTE 1 Essential functions include but are not limited to the safety instrumented function (SIF), the control function, and the ability of the operator to view and manipulate the equipment under control. The loss of essential functions is commonly termed loss of protection, loss of control, and loss of view respectively. In some industries additional functions such as history may be considered essential.

NOTE 2 Source [IEC 62443-4-2].

3.1.20
end user
organization that purchases, uses, or is impacted by the security of IACS products

3.1.21
full virtualization
form of virtualization where one or more operating systems and the applications they contain are run on top of virtualized hardware

NOTE Source NIST SP 800-125.

3.1.22
functional security assessment
assessment of a defined list of security features for a control system, or for a component of a control system

3.1.23
host device
general purpose device running an operating system (for example Microsoft Windows OS or Linux) capable of hosting one or more software applications, data stores or functions from one or more suppliers

NOTE 1 Typical attributes include filesystem(s), programmable services, no real time scheduler and full HMI (keyboard, mouse, etc.).

NOTE 2 Source [IEC 62443-4-2].

3.1.24
hosted virtualization
form of full virtualization where the hypervisor runs on top of a host OS

NOTE 1 Source NIST SP 800-125.

NOTE 2 Another term for such a hypervisor is “Type 2 hypervisor,” as seen for example at https://techterms.com/definition/hypervisor.

3.1.25
hypervisor
virtualization component that manages the guest OSs on a host and controls the flow of instructions between the guest OSs and the physical hardware

NOTE Source NIST SP 800-125.

3.1.26
industrial automation and control system
collection of personnel, hardware, software and policies involved in the operation of the industrial process and that can affect or influence its safe, secure and reliable operation

NOTE Source [IEC 62443-4-2].

3.1.27
IIoT device
entity that is a sensor or actuator for a physical process, or communicates with sensors or actuators for a physical process, that directly connects to an untrusted network to support and/or use data collection and analytic functions accessible via that network

NOTE 1 This definition adds detail for the purposes of the present document, to the definition from ISO/IEC FDIS 20924, 3.2.4 for IoT, which reads “entity of an IoT system that interacts and communicates with the physical world through sensing or actuating.” The 20924 definition does not specify connection to an untrusted network.

NOTE 2 Examples of IIoT devices that communicate with sensors or actuators are a PLC with an internet connection, and an IIoT integrated edge computing device (see 3.1.29).

3.1.28
IIoT gateway
entity of an IIoT system that connects one or more proximity networks and the IIoT devices on those networks to each other and directly connects to one or more untrusted access networks

NOTE 1 This definition is from ISO/IEC FDIS 20924, except that IoT is replaced by IIoT, and the qualification “untrusted” has been added for the purposes of this document.

NOTE 2 From Industrial Internet Consortium Reference Architecture: “The proximity network connects the sensors, actuators, devices, control systems and assets, collectively called edge nodes.”

NOTE 3 An IIoT gateway device is a type of network device (see 3.1.30).

3.1.29
IIoT integrated edge computing device
IIoT device that communicates with other IIoT devices and includes either or both of: environment for hosting application software or pre-defined application software

NOTE 1 The reader is advised that terminology usage in the IoT arena is not standardized at this time, so that other sources may use other terms for this concept.

NOTE 2 Examples of application software are analytics and data filtering. Device may include IIoT gateway functionality to transmit sensor information or derivative information to the cloud, may provide instructions to sensors, actuators, controllers, or other IIoT integrated edge computing devices, application environment may consist of virtual machines and/or a container environment, may use wired communication, or cellular or other wireless communication.

NOTE 3. An example IIoT integrated edge computing device might include sensor connections providing data for a "local" processing capability on the device, and a connection to the cloud for "remote" processing of some version of that data. In this example, the IIoT integrated edge computing device would meet IEC 62443 definitions for network device and host (if it includes an environment for hosting application software) or software application (if it includes pre-defined applications).

3.1.30
network device
device that facilitates data flow between devices, or restricts the flow of data, but may not directly interact with a control process

NOTE 1 Typical attributes include embedded OS or firmware, no HMI, no real-time scheduler and configured through an external interface.

NOTE 2 Source [IEC 62443-4-2].

3.1.31
product supplier
organization that is responsible for compliance of a product with ISASecure requirements

3.1.32
security development artifacts (SDA)
assessment of artifacts that demonstrates that secure product development and maintenance methods have been applied to a particular product

NOTE In some cases these artifacts will be created during an organization's transition to a secure product development process, for products which predate that process, but will be maintained under it going forward.

3.1.33
security level
measure of confidence that the IACS is free from vulnerabilities and functions in the intended manner

3.1.34
symbol
graphic or text affixed or displayed to designate that ISASecure certification has been achieved

NOTE An earlier term for symbol is "mark."

3.1.35
syslog
protocol that specifies a general log entry format and a log entry transport mechanism

NOTE Source NIST CSRC glossary.

3.1.36
tier
designation to identify a set of certification criteria, where any two tiers are comparable under some ordering scheme

NOTE As described in [ICSA-100], ISASecure ICSA offers certification to Core tier or Advanced tier. Advanced is the higher tier, as it encompasses more requirements than Core tier.

3.2 Abbreviations

The following abbreviations are used in this document.

AIDE	Advanced Intrusion Detection Environment
ANSI	American National Standards Institute
ASCI	Automation Standards Compliance Institute
CIS	Center for Internet Security

CPU	central processing unit
CR	component requirement
CSA	Component Security Assurance
CTIA	Cellular Telecommunications Industry Association
DCS	distributed control system
DDS	Data Distribution Service
DTLS	Datagram Transport Layer Security
EDR	embedded device requirement
ENISA	European Network and Information Security Agency
ETSI	European Telecommunications Standards Institute
ETSI EN	ETSI European Standard
FDIS	Final Draft International Standard
FSA	functional security assessment
HDR	host device requirement
HMI	human-machine interface
IACS	industrial automation and control system(s)
ICS	Industrial control system
ICSA	IIoT Component Security Assurance
IEC	International Electrotechnical Commission
IIoT	Industrial Internet of Things
IMA	Integrity Measurement Architecture
IoT	Internet of Things
IPsec	Internet Protocol Security
ISA	International Society of Automation
ISCI	ISA Security Compliance Institute
ISDLA	Security Development Lifecycle Assurance for ICSA
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
MMU	memory management unit
MPU	memory protection unit
NDR	network device requirement
NIST	National Institute of Standards and Technology
NISTIR	NIST Interagency Report
NX	no-execute
OS	operating system
OT	operational technology
PLC	programmable logic controller
PR	practice
RAM	random access memory
RE	requirement enhancement
SD	secure by design
SDA	security development artifacts

SDLA	security development lifecycle assurance
SESIP	Security Evaluation Standard for IoT Platforms
SIEM	security information and event management
SIF	safety instrumented function
SIS	safety instrumented system
SP	special publication
SR	specification of security requirements, system requirement
SSH	Secure Shell
TCG	Trusted Computing Group
TEE	trusted execution environment
TLS	Transport Layer Security
TPM	Trusted Platform Module
TS	technical specification
US	United States

4 Guide to selected commonly accepted practices for IIoT

4.1 Cryptographic techniques

Many commonly accepted practices described in this document use cryptographic techniques. In accordance with IEC 62443-4-2, all use of cryptography by a component is subject to the requirement CR 4.3 in that standard, quoted below. Therefore, other practices in this document incorporate the following practice by reference.

4.1.1 Practices

IIoT PR 4.1.1-1 Cryptographic techniques
Cryptographic algorithms, including key lengths selected and random number generation methods used, conform to ISO/IEC 19790, or conform to an approved national or regional modification to Annex C of ISO/IEC 19790 (noting that such modifications are permitted by ISO/IEC 19790). There should be no reliance on proprietary or modified cryptographic algorithms. The following are examples of documents that provide conforming methods:

- For the United States, methods referenced in FIPS-140-3 “Security Requirements for Cryptographic Modules” fall under this practice. FIPS-140-3 references a list of algorithms in NIST SP 800-140C revision 1 "CMVP Validation Authority Updates to ISO/IEC 24759" and the status of their acceptance in NIST SP 800-131A revision 2 “Transitioning the Use of Cryptographic Algorithms and Key Lengths."

- For recommendations developed for the European Union, from ENISA, “Algorithms, Key Size and Parameters Report. 2014 Recommendations”

- From Germany Federal Office for Information Security, BSI TR-02102-1: "Cryptographic Mechanisms: Recommendations and Key Lengths" Version: 2022-1

4.1.2 ICSA requirements supported

ICSA-311 FSA-CR 4.3 Use of cryptography *If cryptography is required, the component shall use cryptographic security mechanisms according to internationally recognized and proven security practices and recommendations.*

This ICSA requirement is requirement CR 4.3 from [IEC 62443-4-2].

4.1.3 External references

Example reference that supports use of the above standards and recommendations as commonly accepted practices:

The status of the following document as an international standard, supports the status of the documents named in the practice as commonly accepted for IIoT:

ISO/IEC 19790 Information technology — Security techniques — Security requirements for cryptographic modules

Example references regarding implementation of this practice:

Documents provided in statement of practice.

Additional examples:

ETSI TS 119 312 Electronic Signatures and Infrastructures (ESI); Cryptographic Suites V1.4.1 (2021-08)

The following references point out that public key technology is expected to become vulnerable with the advent of quantum computing, and suggest preparatory steps. This development will have significant future impact on the status of references now listed under Practice IIoT PR 4.1.1-1 as “commonly accepted practices.”

US National Security Memorandum regarding risks to vulnerable cryptographic systems

ENISA Report Post Quantum Cryptography: Current state and quantum mitigation

4.2 Compartmentalization

The following sub sections describe commonly accepted practices for applying several common compartmentalization methods to IIoT components. Practices in 4.2.1.1 apply to all compartmentalization methods, where the term “compartmentalization” is used as defined in Section 3.1. A component may use more than one of the specific compartmentalization methods addressed here, in which case all applicable practices apply.

4.2.1 Practices

4.2.1.1 General compartmentalization practices

The following practices apply to IIoT devices and IIoT gateways, when using any method of compartmentalization. Section references shown in parentheses after each practice are from NIST SP 800-190 and/or NIST SP 800-125, which contain guidance regarding containerization and full virtualization, respectively.

4.2.1.1.1 Essential functions

IIoT PR 4.2.1.1.1-1 Compartmentalize control functions
The component architecture has a dedicated compartment or compartments for any essential functions and associated input and output.

IIoT PR 4.2.1.1.1-2 Resource allocation to essential function compartments
If a component hosts both essential functions and non-essential functions using a compartmentalization technology, the component uses the resource allocation capabilities of that technology to allocate, monitor, and limit computing, storage and I/O resources available to non-essential functions of the component. (NIST SP 800-190 2.2, NIST SP 800-125 3.1)

4.2.1.1.2 Security functions

IIoT PR 4.2.1.1.2-1 Separated security functions

For all IIoT components, the following security functions are separated from other component functions

- Setting and storage of credentials for component users
- Verifying user credentials, such as certificate chain validation
- Generation, changing and storage of keys
- Encryption of data for communication or storage
- Perform hash functions used for integrity checking under IIoT PR 4.5.1-2.

IIoT PR 4.2.1.1.2-2 Mechanism for security function separation

Separation of the security functions as listed in IIoT PR 4.2.1.1.2-1 is achieved by either:

- Implementation on certified TPM 2.0 hardware
- Use of a trusted execution environment (TEE) comprised of hardware and/or software that separates all persistent and non-persistent data storage and processing for the set of functions listed in IIoT PR 4.2.1.1.2-1 from other component functions.

IIoT PR 4.2.1.1.2-3 Advanced tier mechanism for security function separation
For high security IIoT components (represented in ICSA as Advanced tier), the security functions described in IIoT PR 4.2.1.1.2-1 are implemented in certified TPM hardware.

NOTE The definition of *tier* is in Section 3.1.

4.2.1.1.3 Threats to separation of compartments

IIoT PR 4.2.1.1.3-1 Least privilege compartment interaction
Policies enforceable by the compartmentalization technology limit interactions between compartments for least privilege.

IIoT PR 4.2.1.1.3-2 Residual threats due to compartment interactions
The threat model for the component considers any threats due to inter-compartment interactions that remain after applying the policy enforcement provided by the compartmentalization technology, as described in IIoT PR 4.2.1.1.3-1. The threat model also considers any threats that could result in bypassing that policy enforcement. Where indicated by the threat model, controls in addition to those provided by the compartmentalization technology are used to limit interactions between compartments. Examples are application of POSIX directory and file access control lists, or mandatory access control capabilities provided by the host operating system, such as SELinux or AppArmor (NIST SP 800-190 4.4.3).

NOTE For example, additional controls might be added in the case where a component is unable to detect anomalous behavior as described in IIoT PR 4.2.1.3-6.

IIoT PR 4.2.1.1.3-3 Security benchmarking for compartmentalization
If a commonly accepted security benchmark is available for a compartmentalization technology used by the component, conformance with the benchmark is evaluated by the supplier to identify and address any vulnerabilities indicated by nonconformance. Examples of such benchmarks are the CIS Docker, Kubernetes, LXD, and VMware Benchmarks. (NIST SP 800-190 4.4.3)

4.2.1.1.4 Security Event Detection

IIoT PR 4.2.1.1.4-1 Identify compartment source for communication
Monitoring of network traffic outbound from the component or outbound communication internal to the component from a compartment, can identify the originating compartment, to support detection of anomalous traffic and incident handling. (NIST SP 800-190 4.4.2)

IIoT PR 4.2.1.1.4-2 Detect compartment infrastructure attack
A component using compartmentalization methods has the capability to detect and log user activity that may indicate unauthorized actions using the compartmentalization infrastructure that can affect the compartments, for example by gaining privileged access to orchestration capabilities or the hypervisor. (NIST SP 800-190 4.5.4)

4.2.1.2 Virtual machines (Full virtualization)

In addition to the general practices in 4.2.1.1, these practices apply when a component includes applications running on a virtualized operating system, where the entire operating system has been virtualized. Section references shown in parentheses after each practice are from NIST SP 800-125.

IIoT PR 4.2.1.2-1 Use bare metal hypervisor
The component uses a bare metal hypervisor and not a hosted hypervisor. (2.2)

NOTE Related definitions are in 3.1.

IIoT PR 4.2.1.2-2 Hypervisor physical protection
Physical access to the component alone does not enable the ability to access the hypervisor. (4.1)

IIoT PR 4.2.1.2-3 Access control for hypervisor
User access to the hypervisor is protected at a minimum with the same strength authentication as any application that will run on the device. (4.1)

4.2.1.3 Containers

In addition to the general practices in 4.2.1.1, the practices in this section apply when a component uses containerization technology for compartmentalization. Section references shown in parentheses after each practice are from NIST SP 800-190.

IIoT PR 4.2.1.3-1 Use container-specific OS
Where feasible, a component uses a container-specific OS minimized to run containerized software, to host its containers. If a component using containerization uses a full OS as a host, the supplier has a documented process they use for hardening the OS prior to product delivery. Further hardening guidance for the user is provided in cases where completion of hardening by the supplier prior to delivery is technically infeasible (such as when user-supplied software may be added to the product and may entail further hardening steps). (4.5.1)

IIoT PR 4.2.1.3-2 Containers non-privileged
Containers not used for orchestration run as non-privileged users on their host OS. (4.1.2)

IIoT PR 4.2.1.3-3 No remote shells
SSH or other remote administration shells are not enabled for communication to a container. (4.1.2)

IIoT PR 4.2.1.3-4 No data values in container
A container does not include specific data values, rather it includes configuration information that allows the container to locate data it creates or uses. (2.1, 3.1.4).

IIoT PR 4.2.1.3-5 No host file system mounted by container
A container does not allow a host file system mount function. (4.5.5)

IIoT PR 4.2.1.3-6 Detect anomalous container behavior
Containers are profiled, with automated tools where possible, so that anomalous container behavior may be detected and logged, such as: (4.4.4)

- Invalid or unexpected process execution
- Invalid or unexpected system calls
- Changes to protected configuration files and binaries

- Writes to unexpected locations and file types
- Creation of unexpected network listeners
- Traffic sent to unexpected network destinations
- Presence or execution of files with unexpected types or sources
- Malware storage or execution.

4.2.2 ICSA requirements supported

ISDLA-312 SDLA-SD-4-ICSA1 Secure design best practices – compartmentalization *Verify that the supplier has a documented secure design practice for compartmentalization. The scope of this practice is when and how to partition critical from less critical functions to facilitate creating a zoning model internal to a component, using commonly accepted practices for IIoT. One source for these practices is the ISASecure document ICSA-500.*

Note: **ICSA-311 FSA-ICSA-12 Component application partitioning** does not explicitly refer to commonly accepted practices, but does so indirectly since it refers to ISDLA-312 requirement SDLA-SD-4-ICSA1. In particular the validation activity for FSA-ICSA-12 reads "Verify that the set of partitions (a.k.a. zones) within the component (where there may be one or more) is in accordance with the suppliers secure design practice for compartmentalization, which is required for certification by specification ISDLA-312 in requirement SDLA-SD-4-ICSA1."

ICSA-311 FSA-ICSA-16 Zone separation methods *Zones within a component shall provide logical or physical separation using approaches and methods in accordance with commonly accepted practices for IIoT.*

ICSA-311 FSA-ICSA-20 Hardware compartmentalization of security functions (Advanced tier) *The component shall use hardware separation to separate those security functions for which employing hardware separation is commonly accepted IIoT practice, from other component functions.*

NOTE The definition of *tier* is in Section 3.1

4.2.3 External references

Example references that support use of the above standards and recommendations as commonly accepted practices:

CTIA Cybersecurity Test Plan for IoT Devices v1.2, available at https://www.ctia.org/certification-resources, Section 5.17 Use separation and segmentation to isolate critical functions

ENISA Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures (2017). Countermeasure GP-PS-05 Design architecture by compartments

Industrial Internet Consortium Security Framework. Section 3.5 design to compartmentalize failures, Section 8 discussion of isolation, 8.12 Isolation techniques

The Seven Properties of Highly Secure Devices, Galen Hunt, George Letey, and Edmund B. Nightingale, Microsoft Research NExT Operating Systems Technologies Group. Compartmentalization is one of the seven properties.

How to Use the TPM: A Guide to Hardware-Based Endpoint Security, lists functions normally supported in a TPM

NIST SP 800-57 Recommendation for Key Management, 6.2.2.3 protection of keys, 8.2.1.1 storage of keys in cryptographic module

For ICSA-311 FSA-ICSA-20:

Industrial Internet Consortium Security Framework. Section 8.2.2 on hardware vs. software security

ENISA Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures (2017) Countermeasure GP-TM-02 “Use hardware that incorporates security features …for example, specialised security chips / coprocessors…providing, among other things, a trusted storage of device identity and authentication means, protection of keys at rest and in use, and preventing unprivileged from accessing to security sensitive code.”

Example references regarding implementation of these practices:

Industrial Internet Consortium Security Framework. Section 3.5, Section 8 discussion of isolation, 8.12 Isolation techniques

NIST SP 800-190 Application Container Security Guide

NIST SP 800-125 Guide to Security for Full Virtualization Technologies

https://linuxcontainers.org/lxc/security/

Global Platform Technology TEE specifications, documents from organization with goal to define common TEE environment

4.3 Human user identification and authentication

4.3.1 Practices

IIoT PR 4.3.1-1 Remote human user authentication methods
Multifactor or certificate-based authentication is used for remote human access to the component.

4.3.2 ICSA requirements supported

ICSA-311 FSA-CR 1.1 Human user identification and authentication *Components shall provide the capability to identify and authenticate all human users according to IEC 62443-3-3 SR 1.1 on all interfaces capable of human user access. This capability shall enforce such identification and authentication on all interfaces that provide human user access to the component to support segregation of duties and least privilege in accordance with applicable security policies and procedures. This capability may be provided locally by the component or by integration into a system level identification and authentication system.*

NOTE 1 In many cases it is expected that an IIoT device or IIoT gateway will support authentication for remote human access by integration into a system, and not necessarily support it directly.

NOTE 2 Multi-factor capability for all human users is required by IEC 62443-4-2 CR 1.1 RE(2) for capability security levels 3 and 4. Multi-factor authentication is therefore required by ICSA for all human access to IIoT devices or gateways at the Advanced tier, separately from the consideration of the ICSA certification criterion in the certifier validation activity for conformance to commonly accepted practices for human user identification and authentication, for this requirement. The definition of *tier* is in Section 3.1.

4.3.3 External references

Example references that support use of authentication methods described by IIoT PR 4.3.1-1 as commonly accepted practices

CTIA Cybersecurity Test Plan for IoT Devices v1.2, available at https://www.ctia.org/certification-resources, Section 4.9, requires multi-factor authentication for human users (at certification level 2)

ENISA Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures (2017) Countermeasure GP-TM-23 consider two-factor authentication, multi-factor authentication and certificates

NERC CIP-005-5 Cyber Security – Electronic Security Perimeter(s) Part 2.3 multi-factor authentication for all interactive Remote Access sessions for applicable systems

4.4 Software processes and device identification and authentication

4.4.1 Practices

IIoT PR 4.4.1-1 Non-human user authentication methods
Software processes and devices that access the component via an untrusted network use certificate-based credentials, to uniquely identify and authenticate themselves to the component.

4.4.2 ICSA requirements supported

ICSA-311 FSA-ICSA-8 Authentication of non-human users from untrusted networks *Components shall provide the capability, either locally or by integration into a system, to uniquely identify and authenticate devices and applications communicating with the component over untrusted networks, using commonly accepted practices for IIoT.*

4.4.3 External references

Example references that support use of authentication methods described by IIoT PR 4.4.1-1 as commonly accepted practices

The Seven Properties of Highly Secure Devices, Galen Hunt, George Letey, and Edmund B. Nightingale, Microsoft Research NExT Operating Systems Technologies Group. Certificate-based mutual authentication with other devices is one of the seven properties.

Industrial Internet Consortium Security Framework. Section 8.6.1 mutual authentication using strong cryptographic credentials for endpoints, where endpoints may be remote (e.g., cloud) or local components

ENISA Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures (2017). Countermeasure GP-TM-23 consider two-factor authentication, multi-factor authentication and certificates

4.5 Software and data at rest integrity checking – ongoing

4.5.1 Practices

IIoT PR 4.5.1-1 Store initial file hash values
A facility is provided which is triggered at first bootup of the component, to calculate and store initial “good” values of a cryptographic hash function applied to each of a baseline set of component files configured by the supplier as selected for integrity checking. At a minimum the baseline set of files will include all critical files.

IIoT PR 4.5.1-2 Compare current to stored file hash values
An integrity check function is provided locally or remotely, to calculate hash values for the current versions of selected files from the baseline set that are present on the component, and compare them to stored values, during component runtime.

IIoT PR 4.5.1-3 Report differences current to stored file hash values
Integrity function output that states whether or not differences are found between the current and stored hash values for selected files, is provided to a remote human interface.

IIoT PR 4.5.1-4 Triggers for integrity checking
The integrity check function is triggered in at least one of the following ways:

- By remote request, on-demand, for all selected files
- According to a remote scheduling capability, for all selected files
- Automatically, for files as they are used.

IIoT PR 4.5.1-5 Hash function for integrity check
The hash function used is approved by a national or regional authority, as described in IIoT PR 4.1.1-1.

IIoT PR 4.5.1-6 Changing stored file hash values
A facility is provided to update stored hash values for files that have authorized modifications, that does not require local access to the component.

IIoT PR 4.5.1-7 Add or delete stored hash values
If any of the critical files of the component can be added or deleted, or change file identifiers, then a facility is provided that allows an authorized user to add or delete files configured for integrity checking that does not require local access to the component.

NOTE Authentication and authorization for human or non-human users to perform these actions is required under ICSA-311 requirements from IEC 62443-4-2: FSA-CR 1.1, FSA-CR 1.1 RE(1) FSA-CR 1.2, FSA-CR 1.2 RE(1), FSA-CR 2.1, FSA-CR 2.1 RE(1) and ICSA requirement addition FSA-ICSA-8. These requirements are supported by practices IIoT PR 4.3.1-1 and IIoT PR 4.4.1-1 in the present document.

4.5.2 ICSA requirements supported

ICSA-311 FSA-CR 3.4 Software and information integrity *Components shall provide the capability to perform or support integrity checks on software, configuration and other information as well as the recording and reporting of the results of these checks or be integrated into a system that can perform or support integrity checks.*

4.5.3 External references

Example references that support above requirements as commonly accepted practices:

TCG Guidance for Securing Industrial Control Systems Using TCG Technology states in 4.2: “Secure and Measured Boot can be extended to run-time software through mechanisms such as the Linux Integrity Measurement Architecture.”

Example references regarding implementation of these practices:

Linux Integrity Measurement Architecture (IMA) and Advanced Intrusion Detection Environment (AIDE) are examples of utilities that can be used to support the above practices.

Linux IMA is a kernel solution, described in the first four references below. Linux IMA natively provides automatic triggering of integrity checking for files as they are used. AIDE is a user space solution, described in the last reference. AIDE natively provides integrity checking triggered on-demand.

An Overview of The Linux Integrity Subsystem

https://www.kernel.org/doc/html/latest/security/IMA-templates.html

https://www.redhat.com/ja/blog/how-use-linux-kernels-integrity-measurement-architecture

TCG Guidance for Securing Industrial Control Systems Using TCG Technology discusses IMA in 6.12.1

https://aide.github.io/

Section 6.12 in the TCG reference also includes this list of types of files appropriate for integrity checking:

- Executables
- Shared libraries
- Policy or configuration files
- Scripts
- Cryptographic and key material used by the OS.

4.6 Integrity protection for software and data in use

4.6.1 Practices

See 3.1 for the definition of software and data “in use.”

IIoT PR 4.6.1-1 Non-persistent memory overflow protection
Kernel and user space non-persistent memory structures including the call stack and counters are protected from overflow attacks. An example method for protecting the call stack is a stack canary. Structures used to track memory in use and free are protected from unauthorized manipulation.

IIoT PR 4.6.1-2 Protect from unauthorized writes to non-persistent memory
The component protects non-persistent memory from unauthorized changes using memory-region protection and monitoring, or RAM encryption.

NOTE The following is an important special case of the previous practice.

IIoT PR 4.6.1-3 Protect from unauthorized writes to security software and data in non-persistent memory
The component protects the integrity of security software and/or related data in non-persistent memory using the generic memory protection methods described in IIoT PR 4.6.1-2, dedicated hardware protection for keys, and/or a virtual or physical CPU or execution environment dedicated to security functions (Trusted Execution Environment, TEE).

4.6.2 ICSA requirements supported

ICSA-311 FSA-ICSA-3 Integrity of software and data in use *Components shall provide the capability to protect while in use by the component, the integrity of software and data that affects the security of the component.*

4.6.3 External references

Example references that support use of methods to protect software and data in use described by IIoT PR 4.6.1-3, as commonly accepted practices:

Industrial Internet Consortium Security Framework Section 8.7.2 memory region protection controls, Section 8.2.2 TEE

ENISA Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures (2017) Countermeasure GP-TM-02 “specialised security chips/coprocessors…providing protection of keys at rest and in use”

ETSI EN 303 645 Cyber Security for Consumer Internet of Things: Baseline Requirements V2.1.1 Provision 5.6-7 example stack canary, Provision 5.6-8 examples memory access control mechanisms include “MMU’s or MPU’s, executable space protection (e.g., NX bits), memory tagging, and trusted execution environments”

Example references regarding implementation:

CWE-121 Stack-based Buffer overflow describes potential mitigations

The Linux Kernel Archives, Kernel Self Protection, Memory integrity

gcc manual page references to stack-protector

The Linux Kernel Archives Memory Protection Keys

The Linux Kernel Archives TEE subsystem, generic interface to TEE

Built in TEE hardware support in ARM: TrustZone

Built in TEE hardware support in RISC V: MultiZone™ Security Trusted Execution Environment

Keystone Customizable TEE Framework

4.7 Communication confidentiality, integrity and authenticity – incoming and outgoing

4.7.1 Practices

IIoT PR 4.7.1-1 Communication protocols
One or more of the following protocols is used for protecting the confidentiality, integrity and authenticity of data transmitted and received by the component over an untrusted network: TLS; DTLS; IPsec; SSH, DDS.

NOTE See also IIoT PR 4.1.1-1 regarding cryptographic techniques to be used in implementation of these protocols.

4.7.2 ICSA requirements supported

ICSA-311 FSA-CR 3.1 Communication integrity *Components shall provide the capability to protect integrity of transmitted information.*

ICSA-311 FSA-CR 3.1 RE(1) Communication authentication *Components shall provide the capability to verify the authenticity of received information during communication.*

ICSA-311 FSA-CR 4.1B Information confidentiality – in transit *Components shall support the protection of the confidentiality of information in transit as defined in IEC 62443-3-3 SR 4.1.*

4.7.3 External references

Example references that support use of the above protocols as commonly accepted practices:

The Industrial Internet of Things Connectivity Framework, references to TLS, DTLS, DDS throughout

Industrial Internet Consortium Security Framework, Section 9 references to TLS, DTLS, DDS, IPsec

ENISA Baseline Security Recommendations for IoT in the Context of Critical Information Infrastructures (2017), Requirement GP-TM-39 TLS as example of state-of-the-art standardized security protocol

CTIA Cybersecurity Test Plan for IoT Devices v1.2, available at https://www.ctia.org/certification-resources, Section 4.8 references to SSH, IPsec, TLS, or DTLS

4.8 Confidentiality protection for data at rest

4.8.1 Practices

IIoT PR 4.8.1-1 Encrypt confidential data at rest
For any data for which explicit read authorization is supported, the component stores only encrypted versions of this data in persistent storage. Examples of possible implementations are:

- Data is encrypted before storing as individual files in flash memory.
- Data is stored in an encrypted filesystem container.
- Data is stored in a self-encrypting drive.

In the following additional practices, such data is referred to as “confidential data.”

IIoT PR 4.8.1-2 Approved cryptography for data at rest encryption
Encryption of confidential data placed in persistent storage uses an encryption function approved by a national or regional authority, as described in IIoT PR 4.1.1-1.

IIoT PR 4.8.1-3 Unique keys for data at rest encryption
Keys used for encryption of confidential data for persistent storage are unique to each instance of the component.

NOTE Protection of keys is addressed under IIoT PR 4.2.1.1.2-1, IIoT PR 4.2.1.1.2-2, and IIoT PR 4.2.1.1.2-3.

4.8.2 ICSA requirements supported

ICSA-311 FSA-CR 4.1A Information confidentiality – at rest *Components shall provide the capability to protect the confidentiality of information at rest for which explicit read authorization is supported.*

NOTE Security parameters such as keys may not have configurable read authorization. The confidentiality of this information is addressed by other IEC 62443-4-2 requirements that are ICSA certification criteria.

4.8.3 External References

Example references that support the above methods for protecting confidential data as commonly accepted practices:

ENISA Baseline Security Recommendations for IoT in the Context of Critical Information Infrastructures (2017), Requirement GP-TM-32 recommends encrypted storage medium

Industrial Internet Consortium Data Protection Best Practices recommends to use encrypted containers (here meaning e.g., database or storage container) and to use widely approved cryptography algorithms

NISTIR 8259A IoT Device Cybersecurity Capability Core Baseline, Data protection capability has associated "common element" which refers to use of cryptographic modules to protect the confidentiality (and integrity) of an IoT device's stored data from compromise

Security Evaluation Standard for IoT Platforms (SESIP) 3.6.2 use of key unique to product instance

Example references regarding implementation of these practices:

TCG Guidance for Securing Industrial Control Systems Using TCG Technology regarding use of self-encrypting drive Section 6.6

4.9 Elements for continuous security monitoring

4.9.1 Practices

IIoT PR 4.9.1-1 IIoT component log events
The component logs security events that include but are not limited to the following, or provides data to another IIoT system element or SIEM to support this logging:

- Flooding attack at network layer on interface to untrusted network
- Flooding attack at application layer on interface to untrusted network
- Account locked out manually or automatically for any reason
- Attempt by any human or non-human user to perform action not authorized (special case of IEC 62443-4-2 CR 2.8 access control event)
- Unusual or unexpected input data rejected by component
- Disallowed attempted data flow between zones internal to the component
- Related cloud application unreachable
- Time since last device reboot
- Failure of integrity check
- Events defined by applications added to the device, where applications added are consistent with supplier recommendations in user documentation.

NOTE 1 Section 4.2 of this document includes additional events and specific instances of the above events, that apply when employing compartmentalization technologies.

NOTE 2 ICSA incorporates IEC 62443-4-2 requirements on auditable events that are not duplicated here, as in CR 2.8 for Core tier, and EDR|HDR|NDR 3.11 RE(1) for Advanced tier. The definition of *tier* is in Section 3.1

4.9.2 ICSA requirements supported

ICSA-311 FSA-CR 6.2 Continuous monitoring *Components shall provide the capability to be continuously monitored using commonly accepted security industry practices and recommendations to detect, characterize and report security breaches in a timely manner.*

4.9.3 External references

Example references that support logging of the events listed above as commonly accepted practices:

Industrial Internet Consortium Security Framework, Section 7.3 references to detecting malicious usage patterns, denial of service activity, enforcement of security policies

4.10 Interfaces for reporting monitored elements

4.10.1 Practices

IIoT PR 4.10.1-1 Event reporting format
Components report results of continuous security monitoring over the network using syslog-ng, together with a method for structuring message content in a human and machine-readable format such as JSON.

IIoT PR 4.10.1-2 SIEM integration
Components are able to demonstrate the capability to report to at least one well-known Security Information and Event Management system (SIEM), for example including but not limited to: Splunk, Verve OT/ICS SIEM, Wazuh and LogESP.

4.10.2 ICSA requirements supported

ICSA-311 FSA-CR 6.2 Continuous monitoring *Components shall provide the capability to be continuously monitored using commonly accepted security industry practices and recommendations to detect, characterize and report security breaches in a timely manner.*

4.10.3 External references

Example references that support the above event reporting features as commonly accepted practices:

NIST SP 800-92 Guide to Computer Security Log Management sections 3.3 and 3.4 discuss use of syslog and the SIEM concept

CTIA Cybersecurity Test Plan for IoT Devices v1.2, available at https://www.ctia.org/certification-resources, Section 4.7 requires syslog format

Example administrative guide for syslog-ng has section on JSON parser

Example references for leading SIEM capabilities in the IACS space:

Protecting Operational Technology with Splunk

Verve OT/ICS SIEM

Wazuh

LogESP

4.11 Values for supplier-assigned unique passwords and keys

4.11.1 Practices

IIoT PR 4.11.1-1 Randomness for generated security parameters
Supplier-assigned passwords and keys for IIoT devices and gateways are generated in accordance with requirements on random number generators defined under IIoT PR 4.1.1-1 regarding cryptographic techniques.

4.11.2 ICSA requirements supported

ICSA-311 FSA-ICSA-2 Unique initial passwords and keys *If the component when initially placed in operational state has default passwords or keys, either (1) these shall have individual unique values per component delivered, that meet commonly accepted guidelines or (2) the component shall require changing the password or key before operational use.*

4.11.3 External references

See IIoT PR 4.1.1-1.

5 Appendix – Mapping from ICSA certification requirements to ICSA-500 practices

The following table lists the ICSA certification requirements in ICSA-311 and ISDLA-312 that have corresponding practices in the present document. In Table 1 below, requirement IDs that start with "FSA-CR" are IEC 62443-4-2 requirements incorporated in ICSA-311. The requirement IDs that start with "FSA-ICSA" are ICSA additions defined in ICSA-311 that are not found in IEC 62443-4-2.

Table 1. Mapping from ICSA certification requirements to ICSA-500 practices

ICSA specification	Requirement ID	Requirement Title	Practice ID	Practice Title	ICSA-500 Section
ICSA-311	FSA-CR 1.1	Human user identification and authentication	IIoT PR 4.3.1-1	Remote human user authentication methods	4.3.1
ICSA-311	FSA-CR 3.1	Communication integrity	IIoT PR 4.7.1-1	Communication protocols	4.7.1
ICSA-311	FSA-CR 3.1 RE(1)	Communication authentication	IIoT PR 4.7.1-1	Communication protocols	4.7.1
ICSA-311	FSA-CR 4.1B	Information confidentiality – in transit	IIoT PR 4.7.1-1	Communication protocols	4.7.1
ICSA-311	FSA-CR 3.4	Software and information integrity	IIoT PR 4.5.1-1	Store initial file hash values	4.5.1
ICSA-311	FSA-CR 3.4	Software and information integrity	IIoT PR 4.5.1-2	Compare current to stored file hash values	4.5.1
ICSA-311	FSA-CR 3.4	Software and information integrity	IIoT PR 4.5.1-3	Report differences current to stored file hash values	4.5.1
ICSA-311	FSA-CR 3.4	Software and information integrity	IIoT PR 4.5.1-4	Triggers for integrity checking	4.5.1
ICSA-311	FSA-CR 3.4	Software and information integrity	IIoT PR 4.5.1-5	Hash function for integrity check	4.5.1
ICSA-311	FSA-CR 3.4	Software and information integrity	IIoT PR 4.5.1-6	Changing stored file hash values	4.5.1
ICSA-311	FSA-CR 3.4	Software and information integrity	IIoT PR 4.5.1-7	Add or delete stored hash values	4.5.1
ICSA-311	FSA-CR 4.1A	Information confidentiality – at rest	IIoT PR 4.8.1-1	Encrypt confidential data at rest	4.8.1

ICSA specification	Requirement ID	Requirement Title	Practice ID	Practice Title	ICSA-500 Section
ICSA-311	FSA-CR 4.1A	Information confidentiality – at rest	IIoT PR 4.8.1-2	Approved cryptography for data at rest encryption	4.8.1
ICSA-311	FSA-CR 4.1A	Information confidentiality – at rest	IIoT PR 4.8.1-3	Unique keys for data at rest encryption	4.8.1
ICSA-311	FSA-CR 4.1B	Information confidentiality – in transit	IIoT PR 4.7.1-1	Communication protocols	4.7.1
ICSA-311	FSA-CR 4.3	Use of cryptography	IIoT PR 4.1.1-1	Cryptographic techniques	4.1.1
ICSA-311	FSA-CR 6.2	Continuous monitoring	IIoT PR 4.9.1-1	IIoT component log events	4.9.1
ICSA-311	FSA-CR 6.2	Continuous monitoring	IIoT PR 4.10.1-1	Event reporting format	4.10.1
ICSA-311	FSA-CR 6.2	Continuous monitoring	IIoT PR 4.10.1-2	SIEM integration	4.10.1
ICSA-311	FSA-ICSA-2	Unique initial passwords and keys	IIoT PR 4.11.1-1	Randomness for generated security parameters	4.11.1
ICSA-311	FSA-ICSA-3	Integrity of software and data in use	IIoT PR 4.6.1-1	Non-persistent memory overflow protection	4.6.1
ICSA-311	FSA-ICSA-3	Integrity of software and data in use	IIoT PR 4.6.1-2	Protect from unauthorized writes to non-persistent memory	4.6.1
ICSA-311	FSA-ICSA-3	Integrity of software and data in use	IIoT PR 4.6.1-3	Protect from unauthorized writes to security software and data in non-persistent memory	4.6.1
ICSA-311	FSA-ICSA-8	Authentication of non-human users from untrusted networks	IIoT PR 4.4.1-1	Non-human user authentication methods	4.4.1

ICSA specification	Requirement ID	Requirement Title	Practice ID	Practice Title	ICSA-500 Section
ICSA-311	FSA-ICSA-12	Component application partitioning	IIoT PR 4.2.1.1.1-1	Compartmentalize control functions	4.2.1.1.1
ICSA-311	FSA-ICSA-12	Component application partitioning	IIoT PR 4.2.1.1.2-1	Security functions	4.2.1.1.2
ISCA-311	FSA-ICSA-16	Zone separation methods	All practice IDs in section 4.2.1	All practices in section 4.2.1	4.2.1
ICSA-311	FSA-ICSA-20	Hardware compartmentalization of security functions (Advanced tier)	IIoT PR 4.2.1.1.2-3	Advanced tier mechanism for security function separation	4.2.1.1.2
ISDLA-312	SDLA-SD-4-ICSA1	Secure design best practices - compartmentalization	All practice IDs in section 4.2.1	All practices in section 4.2.1	4.2.1