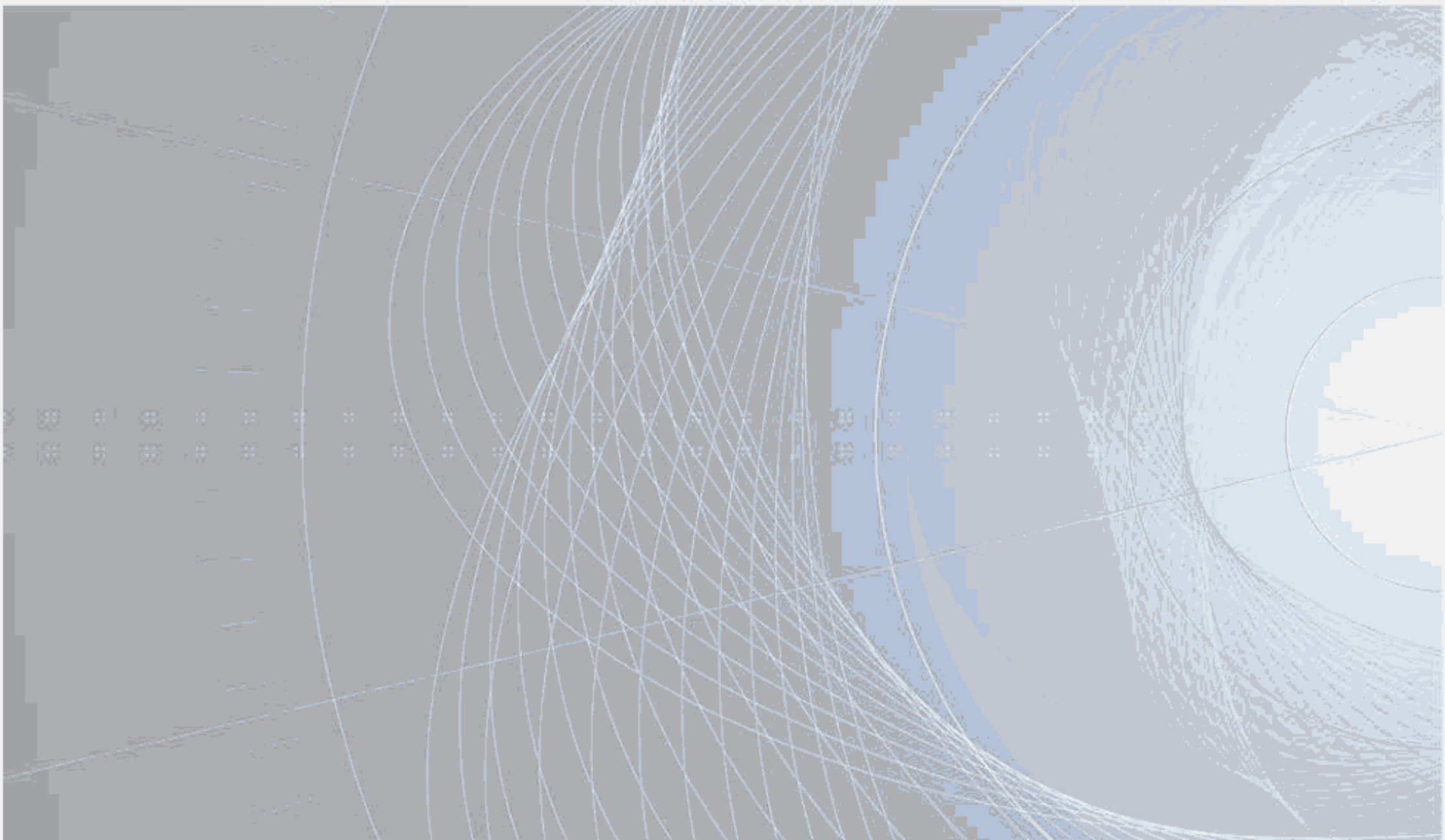
IEC 62443-4-1

Edition 1.0 2018-01

# INTERNATIONAL STANDARD

## Security for industrial automation and control systems – Part 4-1: Secure product development lifecycle requirements



IEC 62443-4-1

Edition 1.0 2018-01

# INTERNATIONAL STANDARD

colour inside

## Security for industrial automation and control systems – Part 4-1: Secure product development lifecycle requirements

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ICS 25.040.40; 35.030

ISBN 978-2-8322-5239-0

# CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

# SECURITY FOR INDUSTRIAL AUTOMATION AND CONTROL SYSTEMS –

## Part 4-1: Secure product development lifecycle requirements

## FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62443-4-1 has been prepared by IEC technical committee 65: Industrial-process measurement, control and automation.

The text of this International Standard is based on the following documents:

| FDIS | Report on voting |
|---|---|
| 65/685/FDIS | 65/688/RVD |

Full information on the voting for the approval of this International Standard can be found in the report on voting indicated in the above table.

This document has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 62443 series, published under the general title *Security for industrial automation and control systems*, can be found on the IEC website.

Future standards in this series will carry the new general title as cited above. Titles of existing standards in this series will be updated at the time of the next edition.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under "http://webstore.iec.ch" in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

**IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.**

# INTRODUCTION

This document is part of a series of standards that addresses the issue of security for industrial automation and control systems (IACS). This document describes product development life-cycle requirements related to cyber security for products intended for use in the industrial automation and control systems environment and provides guidance on how to meet the requirements described for each element.

This document has been developed in large part from the Secure Development Life-cycle Assessment (SDLA) Certification Requirements [26] [1] from the ISA Security Compliance Institute (ISCI). Note that the SDLA procedure was based on the following sources:

– ISO/IEC 15408-3 (Common Criteria) [18];
– Open Web Application Security Project (OWASP) Comprehensive, Lightweight Application Security Process (CLASP) [36];
– The Security Development Life-cycle by Michael Howard and Steve Lipner [43];
– IEC 61508 Functional safety of electrical/electronic/ programmable electronic safety-related systems [24], and
– RCTA DO-178B Software Considerations in Airborne Systems and Equipment Certification [28].

Therefore, all these sources can be considered contributing sources to this document.

This document is the part of the IEC 62443 series that contains security requirements for developers of any automation and control products where security is a concern.

Figure 1 illustrates the relationship of the different parts of IEC 62443 that were in existence or planned as of the date of circulation of this document. Those that are normatively referenced are included in the list of normative references in Clause 2, and those that are referenced for informational purposes or that are in development are listed in the Bibliography.

---

1 Figures in square brackets refer to the bibliography.

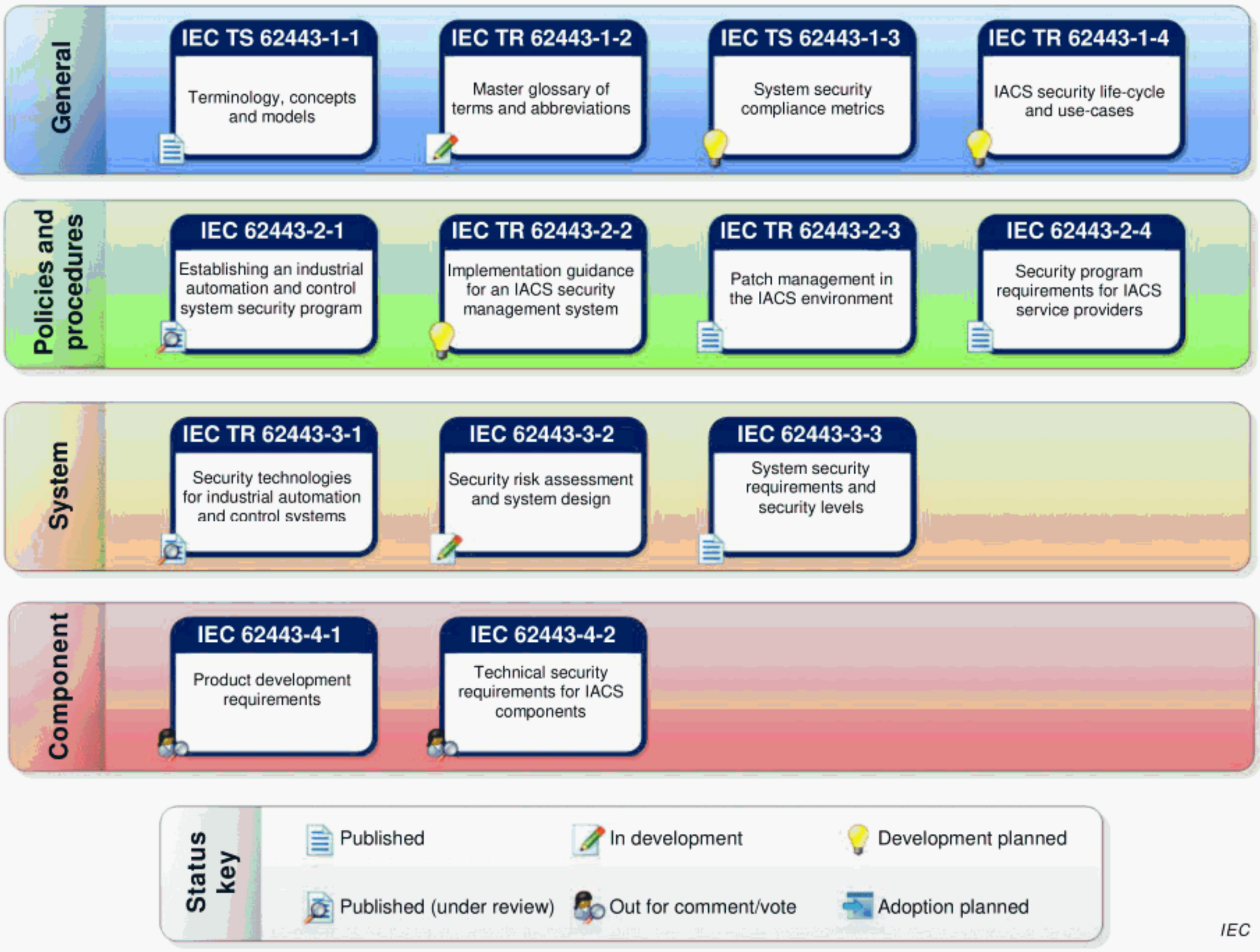



**Figure 1 – Parts of the IEC 62443 series**

Figure 2 illustrates how the developed product relates to maintenance and integration capabilities defined in IEC 62443-2-4 and to its operation by the asset owner. The product supplier develops products using a process compliant with this document. Those products may be a single component, such as an embedded controller, or a group of components working together as a system or subsystem. The products are then integrated together, usually by a system integrator, into an Automation Solution using a process compliant with IEC 62443-2-4. The Automation Solution is then installed at a particular site and becomes part of the industrial automation and control system (IACS). Some of these capabilities reference security measures defined in IEC 62443-3-3 [10] that the service provider ensures are supported in the Automation Solution (either as product features or compensating mechanisms). This document only addresses the process used for the development of the product; it does not address design, installation or operation of the Automation Solution or IACS.

In Figure 2, the Automation Solution is illustrated to contain one or more subsystems and optional supporting components such as advanced control. The dashed boxes indicate that these components are “optional”.

NOTE 1 Automation Solutions typically have a single product, but they are not restricted to do so. In some industries, there may be a hierarchical product structure. In general, the Automation Solution is the set of hardware and software, independent of product packaging, that is used to control a physical process (for example, continuous or manufacturing) as defined by the asset owner.

NOTE 2 If a service provider provides products used in the Automation Solution, then the service provider is fulfilling the role of product supplier in this diagram.

NOTE 3 If a service provider provides products used in the Automation Solution, then the service provider is fulfilling the role of product supplier in this diagram.

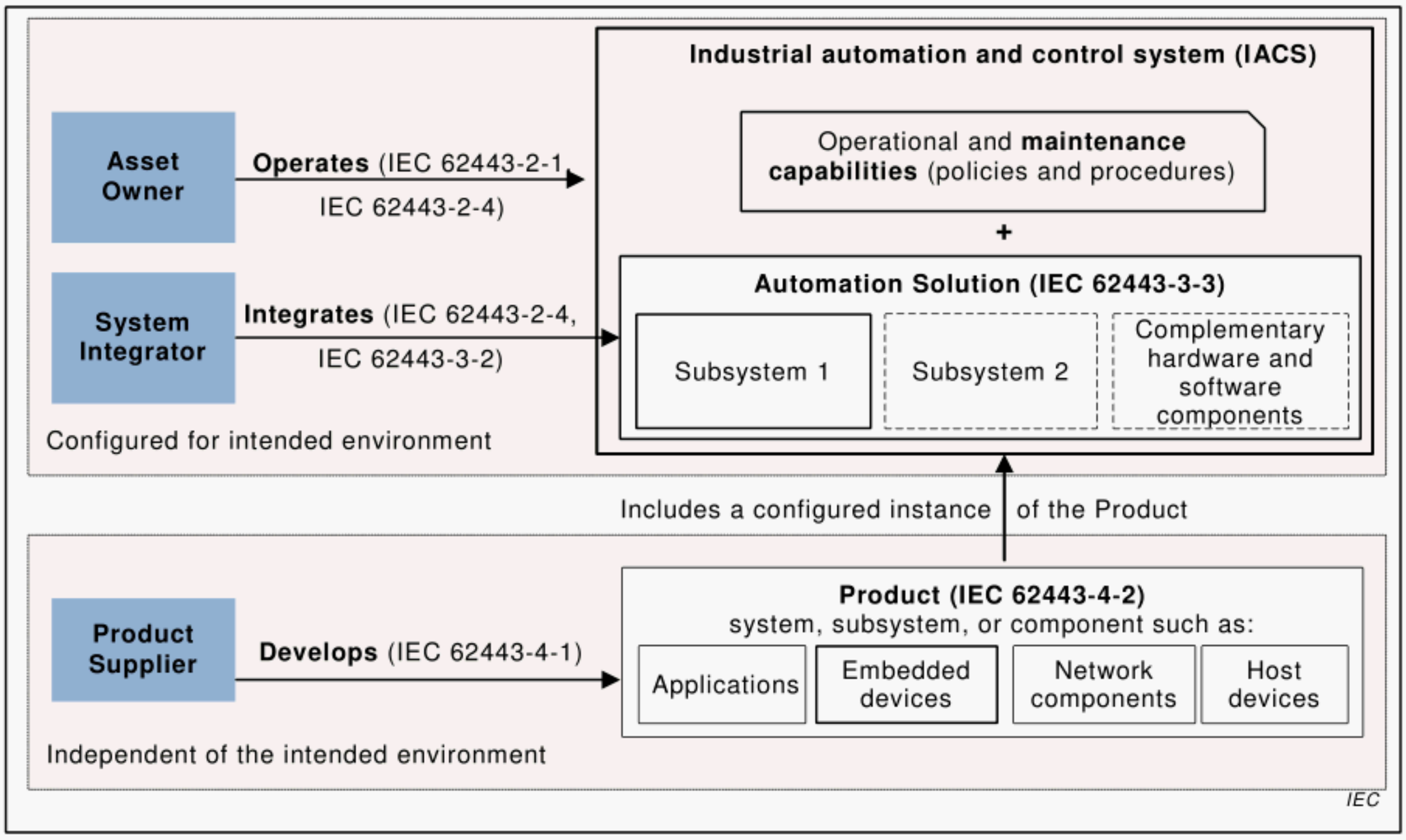



**Figure 2 – Example scope of product life-cycle**

# SECURITY FOR INDUSTRIAL AUTOMATION AND CONTROL SYSTEMS –

# Part 4-1: Secure product development lifecycle requirements

## 1 Scope

This part of IEC 62443 specifies process requirements for the secure development of products used in industrial automation and control systems. It defines a secure development life-cycle (SDL) for the purpose of developing and maintaining secure products. This life-cycle includes security requirements definition, secure design, secure implementation (including coding guidelines), verification and validation, defect management, patch management and product end-of-life. These requirements can be applied to new or existing processes for developing, maintaining and retiring hardware, software or firmware for new or existing products. These requirements apply to the developer and maintainer of the product, but not to the integrator or user of the product. A summary list of the requirements in this document can be found in Annex B.

## 2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 62443-2-4:2015, *Security for industrial automation and control systems – Part 2-4: Security program requirements for IACS service providers*
IEC 62443-2-4:2015/AMD1:2017

## 3 Terms, definitions, abbreviated terms, acronyms and conventions

### 3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC TR 62443-1-2[2] and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

**3.1.1**
**abuse case**
test case used to perform negative operations of a use case

Note 1 to entry: Abuse case tests are simulated attacks often based on the threat model. An abuse case is a type of complete interaction between a system and one or more actors where the results of the interaction are intentionally intended to be harmful to the system, one of the actors or one of the stakeholders in the system.

---

2 Under consideration.

**3.1.2**
**access control** <protection>
protection of system resources against unauthorized access

**3.1.3**
**access control** <process>
process by which use of system resources is regulated according to a security policy and is permitted by only authorized users according to that policy

Note 1 to entry: Access control includes identification and authentication requirements specified in other parts of the IEC 62443 series.

**3.1.4**
**administrator**
user who has been authorized to manage security policies/capabilities for a product or system

**3.1.5**
**asset**
physical or logical object owned by or under the custodial duties of an organization, having either a perceived or actual value to the organization

Note 1 to entry: In this specific case, an asset is an object that is part of an IACS.

**3.1.6**
**asset owner**
individual or organization responsible for one or more IACSs

**3.1.7**
**attack surface**
physical and functional interfaces of a system that can be accessed and, therefore, potentially exploited by an attacker

**3.1.8**
**audit log**
event log that requires a higher level of integrity protection than provided by typical event logs

Note 1 to entry: Audit logs are used to protect against claims that repudiate responsibility for an action.

**3.1.9**
**authentication**
provision of assurance that a claimed characteristic of an identity is correct

Note 1 to entry: Not all credentials used to authenticate an identity are created equally. The trustworthiness of the credential is determined by the configured authentication mechanism. Hardware or software-based mechanisms can force users to prove their identity before accessing data on a device. A typical example is proving the identity of a user usually through an identity provider.

Note 2 to entry: Authentication includes verifying human users as well as non-human users such as devices or processes.

**3.1.10**
**automation solution**
control system and any complementary hardware and software components that have been installed and configured to operate in an IACS

Note 1 to entry: Automation Solution is used as a proper noun in this part of the IEC 62443 series.

Note 2 to entry: The difference between the control system and the Automation Solution is that the control system is incorporated into the Automation Solution design (for example, a specific number of workstations, controllers and devices in a specific configuration), which is then implemented. The resulting configuration is referred to as the Automation Solution.

Note 3 to entry: The Automation Solution can be comprised of components from multiple suppliers including the product supplier of the control system.

**3.1.11**
**banned function**
software method that is no longer recommended to be used in software because more secure versions exist with less propensity for misuse

Note 1 to entry: Banned functions are sometimes called banned methods or banned Application Programming Interfaces (APIs).

**3.1.12**
**best practices**
guidelines for securely designing, developing, testing, maintaining or retiring products that the supplier has determined are commonly recommended by both the security and industrial automation communities

EXAMPLE Least privilege, economy of mechanism and least common mechanism.

**3.1.13**
**component**
one of the parts that make up a product or system

Note 1 to entry: A component may be hardware or software and may be subdivided into other components.

**3.1.14**
**configuration management**
discipline of identifying the components of an evolving system for the purposes of controlling changes to those components and maintaining continuity and traceability throughout the life-cycle

**3.1.15**
**defense in depth**
approach to defend the system against any particular attack using several independent methods

Note 1 to entry: Defense in depth implies layers of security and detection, even on single systems, and provides the following features:

- is based on the idea that any one layer of protection, may and probably will be defeated;
- attackers are faced with breaking through or bypassing each layer without being detected;
- a flaw in one layer can be mitigated by capabilities in other layers;
- system security becomes a set of layers within the overall network security; and
- each layer should be autonomous and not rely on the same functionality nor have the same failure modes as the other layers.

**3.1.16**
**dependent component**
component external to the product on which the product depends

EXAMPLE Java run time environment or a driver

Note 1 to entry: This includes both hardware and software.

**3.1.17**
**deprecated function**
software method that is supported but whose use is no longer recommended

Note 1 to entry: Methods are generally deprecated before becoming obsolete (deleted from the set of functions provided by the supplier of the function). Deprecated functions are sometimes called deprecated methods or deprecated APIs.

**3.1.18**
**externally provided component**
component included in a product that is developed by an external organization and is not developed specifically for one supplier

Note 1 to entry: Examples include purchased software and open source software.

**3.1.19**
**fuzz testing**
process of creating malformed or unexpected data or call sequences to be consumed by the entity under test to verify that they are handled appropriately

**3.1.20**
**industrial automation and control system**
collection of personnel, hardware, software, procedures and policies involved in the operation of the industrial process and that can affect or influence its safe, secure and reliable operation

Note 1 to entry: The IACS can include components that are not installed at the asset owner's site.

Note 2 to entry: The definition of IACS was taken from in IEC 62443-3-3 [10] and is illustrated in Figure 2.

**3.1.21**
**patch management**
area of systems management that involves acquiring, testing and installing software patches (code changes) to a product

Note 1 to entry: See IEC TR 62443-2-3 [7] for additional information.

Note 2 to entry: Patch management also applies to the process of keeping included 3rd party libraries current before releasing a product.

**3.1.22**
**product**
system, subsystem or component that is manufactured, developed or refined for use by other products

Note 1 to entry: The processes required by the practices defined in this document apply iteratively to all levels of product design (for example, from the system level to the component level).

**3.1.23**
**product security context**
security provided to the product by the environment (asset owner deployment) in which the product is intended to be used

Note 1 to entry: The security provided to the product by its intended environment can effectively restrict the threats that are applicable to the product.

**3.1.24**
**product supplier**
manufacturer of hardware and/or software product

Note 1 to entry: The product supplier includes the entity responsible for developing and maintaining a product which can include more than just the manufacturer (for example, integrator).

**3.1.25**
**product users**
users of the hardware and/or software product including asset owners, integrators and maintenance personnel, vendors of other components or products that reuse or contain this product

**3.1.26**
**record**
document stating results achieved or providing evidence of activities performed

Note 1 to entry: The term artefact is often used to have the same meaning.

**3.1.27**
**regression**
change to a system component that has adversely affected functionality, reliability or performance or has introduced additional defects

**3.1.28**
**root cause**
initiating cause of either a condition or a causal chain that leads to an outcome or effect of interest

Note 1 to entry: These weaknesses often result from misapplication of best practices.

**3.1.29**
**security defect**
design or implementation deficiency that can be exploited to compromise an asset or resource

**3.1.30**
**security advisor**
organizational role to guide team in the process of the SDL (Security Development Life-cycle)

Note 1 to entry: Security advisor may be part of the project team or may be consultant to the team to provide guidance and assistance where required.

**3.1.31**
**security incident**
security compromise that is of some significance to the asset owner or failed attempt to compromise the system whose result could have been of some significance to the asset owner

Note 1 to entry: The term “near miss” is sometimes used to describe an event that could have been an incident under slightly different circumstances.

**3.1.32**
**security-related issue**
characteristic of the design or implementation of the product that can potentially affect the security of the product

**3.1.33**
**security verification and validation testing**
testing performed to assess the overall security of a component, product or system when used in its intended product security context and to determine if a component, product or system satisfies the product security requirements and satisfies its designed security purpose

Note 1 to entry: Security verification testing supplements security validation testing with additional testing focused on the product security context and defense in depth strategy.

**3.1.34**
**system integrator**
person or company that specializes in bringing together component subsystems into a whole and ensuring that those subsystems perform in accordance with project specifications

**3.1.35**
**third party supplier**
organization independent of the product supplier organization

**3.1.36**
**threat**
circumstance or event with the potential to adversely affect operations (including mission, functions, image or reputation), assets, control systems or individuals via unauthorized access, destruction, disclosure, modification of data and/or denial of service

**3.1.37**
**threat modelling**
security design analysis technique that identifies potential security issues

Note 1 to entry: Threat models are often synonymous with attack trees and are used by software and hardware architects to identify and mitigate potential security issues early, when they are relatively easy and cost-effective to resolve.

**3.1.38**
**trust boundary**
element of a threat model that depicts a boundary where authentication is required or a change in trust level occurs (higher to lower or vice versa)

Note 1 to entry: Trust boundary enforcement mechanisms for product users typically include authentication (for example, challenge/response, passwords, biometrics or digital signatures) and associated authorization (for example, access control rules).

Note 2 to entry: Trust boundary enforcement mechanisms for data typically include source authentication (for example, message authentication codes and digital signatures) and/or content validation.

**3.1.39**
**unit testing**
verification that an individual unit of computer software or hardware performs as intended

Note 1 to entry: Automated verification, or testing, is generally performed by computer test software.

Note 2 to entry: What constitutes a unit of source code is a design decision. A unit is often designed as the smallest testable part of an application. It may include one or more computer program modules and may also include associated control data, usage procedures and operating procedures. In procedural programming, a unit could be an entire module, but is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method.

**3.1.40**
**user**
person, organization entity, or automated process that accesses a system, whether authorized to do so or not

**3.1.41**
**zone**
collection of entities that represents partitioning of a System under Consideration on the basis their functional, logical and physical (including location) relationship

Note 1 to entry: Zones are often created on the basis of common security requirements, criticality (e.g., high financial, health, safety, or environmental impact), functionality, logical or physical (including location) relationship

## 3.2 Abbreviated terms and acronyms

The following abbreviated terms and acronyms are used in this document.

| | |
|---|---|
| ACL | Access control list |
| CMMI | Capability maturity model integration |
| CMMI-DEV | Capability maturity model integration for development |
| CMU | Carnegie Mellon University |
| COTS | Commercial off the shelf |
| CVSS | Common vulnerability scoring system |

| | |
|---|---|
| DM | Defect management |
| e.g. | exempli gratia |
| FDIS | Final Draft International Standard |
| HTTP | Hypertext transfer protocol |
| IACS | Industrial automation and control system(s) |
| IEC | International Electrotechnical Commission |
| ISA | International Society of Automation |
| ISO | International Organization for Standardization |
| MIN | Minimum |
| OWASP | Open Web Application Security Project |
| SL-C | Capability Security Level |
| SCA | Static code analysis |
| SD | Secure Design |
| SDL | Security development life-cycle |
| SDLA | Secure Development Life-Cycle Assessment |
| SEI | Software Engineering Institute |
| SG | Security guidelines |
| SI | Secure implementation |
| SM | Security management |
| SR | Security requirements |
| STRIDE | Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege |
| SVV | Security verification and validation |
| TCP | Transmission control protocol |
| TCP/IP | Transmission control protocol/Internet protocol |
| TR | Technical report |
| USB | Universal serial bus |

### 3.3 Conventions

Requirements defined in this document generally begin with the phrase “A process shall be employed…”. This terminology is used to specify that the required processes have to be part of the product suppliers documented product development life-cycle processes. The practice of these requirements is dependent on the product supplier having product development projects that require the use of these processes.

According to 5.5, products requiring the use of these processes shall be identified.

## 4 General principles

### 4.1 Concepts

The primary goal of these requirements is to provide a framework to address a secure by design, defense in depth approach to designing, building, maintaining and retiring products used in industrial automation and control products and systems. Application of the framework is intended to provide confidence that the component, product or system has security commensurate with its expected level of risk throughout the product’s life-cycle. While the concept of security levels is not discussed in this document, complying with this document

will help ensure that the security capabilities implemented in the product (see IEC 62443-4-2[3] [11]) will be implemented correctly and that any known security vulnerabilities in the product are eliminated or mitigated. Therefore, compliance with this document supports meeting the overall capability security level (SL-C) of the product.

The secondary goal of these requirements is to align the development process with the elevated security needs of product users of Industrial Automation and Control Systems (for example, providers of IEC 62443-2-4 capabilities such as integrators and maintenance contractors). This means that the process needs to generate items such as well-documented security configurations and patch management policies and procedures, as well as providing clear and succinct communications about security vulnerabilities uncovered in the product.

NOTE 1 For IACS, IEC 62443-3-2[4] [9] describes requirements for determining the expected level of risk associated with the system's zones and conduits.

Figure 3 illustrates how secure by design principles in this document contribute to a defense in depth strategy for the product. The security management practice is shown on the outermost circle because it is applied throughout all the other practices to ensure that the practices are being followed and managed. The other practices, shown on the second circle are applied throughout the development life-cycle, often in an iterative pattern. These practices each contribute to the overall defense in depth strategy which is shown as the center of the circle because it represents the key result of following the security development life-cycle. The defect management and security update management provide verified repairs to the secure implementation, and fall under the category of overall security management in the diagram.

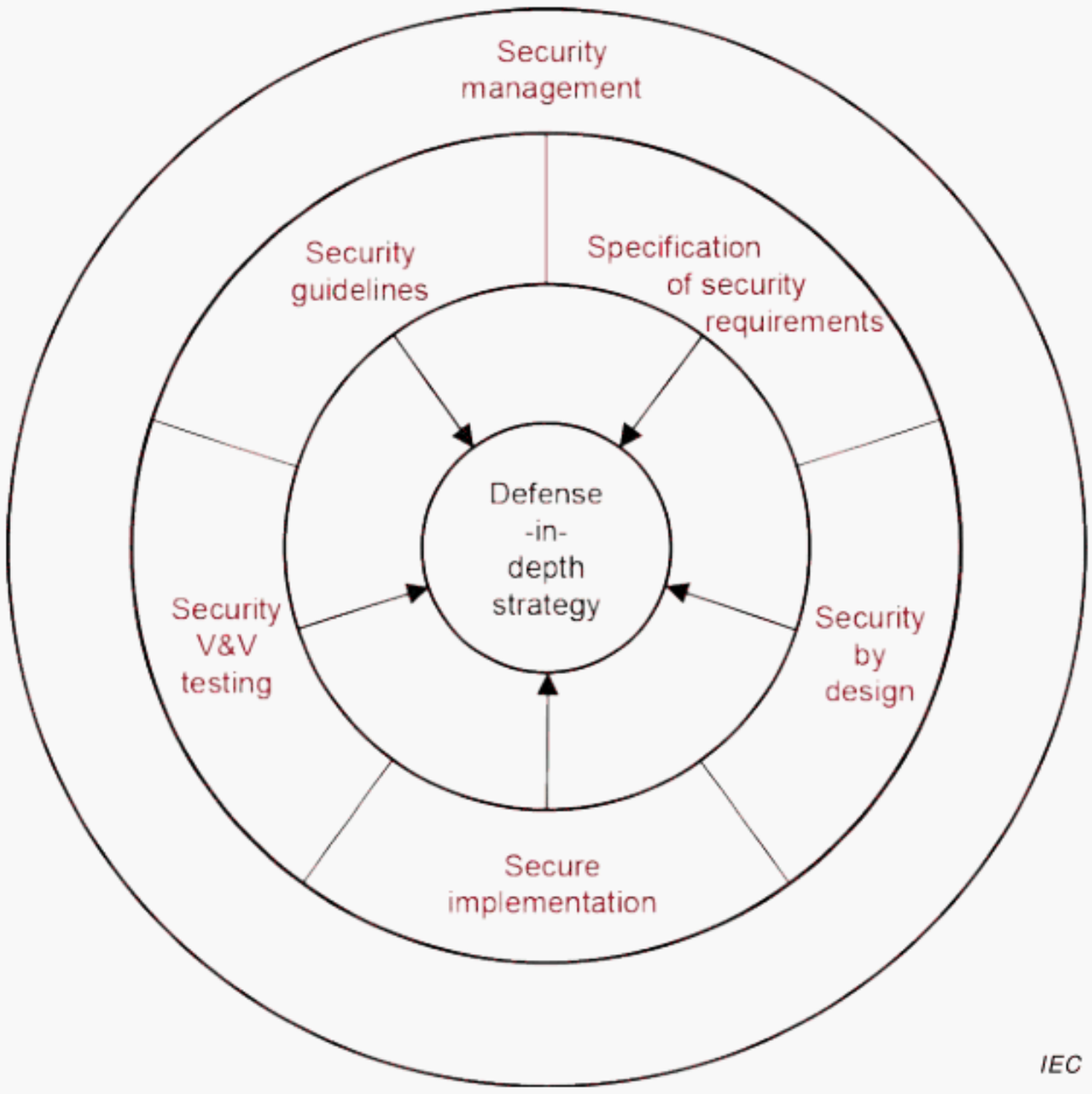



**Figure 3 – Defence in depth strategy is a key philosophy of the secure product life-cycle**

A key concept used throughout this document is the use of threat modelling. Design and implementation reviews to refine work products improve the security posture of a product. Reviews of any work product (for example, requirements, design records, implemented

---

3 Under preparation. Stage at the time of publication: IEC/CDV 62443-4-2:2017.

4 Under preparation. Stage at the time of publication: IEC/CDV 62443-3-2:2017.

modules and verification/validation testing) through any means (for example, manual, automated or a combination) are used to discover security-related issues in the product. An impact analysis of each discovered issue assesses its security severity based on its potential compromise (for example, availability, integrity and confidentiality) and its associated losses (for example, control of the process, view of the process and intellectual property). Then, the resolution process determines the most appropriate course of action based on the issue's severity and the suitability of various response options.

## 4.2 Maturity model

There is a range of methods by which a product supplier could comply with the requirements specified in this document. The maturity model sets benchmarks for meeting these requirements.

These benchmarks are defined by maturity levels as shown in Table 1. The maturity levels are based on the Capability Maturity Model Integration (CMMI) for Development (CMMI-DEV) model [42]. Table 1 shows the relationship to the CMMI-DEV in the description column.

Maturity levels provide more details on how thoroughly a supplier has met these requirements. Therefore, these levels can be used by system integrators and asset owners to assess the level of rigor used to develop products.

The purpose of the maturity levels described in this subclause is to provide an organization a benchmark to define their readiness to use their processes and procedures to design and implement a secure product. Using these benchmarks, it is possible that an organization will discover that it is not ready to implement all requirements to the same level of maturity.

When designing, and implementing a secure product according to this document (see IEC 62443-4-2 and IEC 62443-3-3), all applicable requirements as defined by SM-5 in this document shall be followed and used in the development lifecycle of that product regardless of the maturity level of the organization. SM-5 provides for case-by-case exceptions for applicability of requirements.

NOTE 1 Industry groups/sectors identify/select those maturity levels that best meet their individual needs.

NOTE 2 It is intended that over time and for a specific requirement, a product supplier's development capabilities will evolve to higher levels as it gains proficiency in meeting the requirement. The rate of evolution will often vary for each requirement. For example, a product supplier might reach Level 4 for Practice 6 months or years before they reach Level 4 for Practice 5.

NOTE 3 These maturity levels have been defined as a way for organizations to measure and report their compliance to this document. This model will allow organizations to evolve to higher (more mature) levels of capabilities for their processes.

NOTE 4 Measurement driven continuous improvement is vital for improving the maturity level of a product supplier for each practice in this document. In addition, since best practices for secure product development are evolving, product suppliers need to seek out and implement new best practices.

**Table 1 – Maturity levels**

| Level | CMMI-DEV | IEC 62443-4-1 | IEC 62443-4-1 Description |
|---|---|---|---|
| 1 | Initial | Initial | Product suppliers typically perform product development in an ad-hoc and often undocumented (or not fully documented) manner. As a result, consistency across projects and repeatability of processes may not be possible. |
| 2 | Managed | Managed | At this level, the product supplier has the capability to manage the development of a product according to written policies (including objectives). The product supplier also has evidence to show that personnel who will perform the process have the expertise, are trained and/or follow written procedures to perform it.<br><br>However, at this level, the organization does not have experience developing products to all of the written policies. This would be the case when the organization has updated its procedures to conform to this document, but has not yet put all of the procedures into actual practice, yet.<br><br>The development discipline reflected by maturity level 2 helps to ensure that development practices are repeatable, even during times of stress. When these practices are in place, their execution will be performed and managed according to their documented plans.<br><br>NOTE At this level, the CMMI and IEC 62443-4-1 maturity models are fundamentally the same, with the exception that IEC 62443-4-1 recognizes that there may be a significant delay between defining/formalizing a process and executing (practicing) it. Therefore, the execution related aspects of the CMMI-DEV Level 2 are deferred to Level 3. |
| 3 | Defined | Defined (Practiced) | The performance of a level 3 product supplier can be shown to be repeatable across the supplier's organization. The processes have been practiced, and evidence exists to demonstrate that this has occurred.<br><br>NOTE At this level, the CMMI and IEC 62443-4-1 maturity models are fundamentally the same, with the exception that the execution related aspects of the CMMI-DEV level 2 are included here. Therefore, a process at level 3 is a level 2 process that the supplier has practiced for at least one product. |
| 4 | Quantitatively Managed | Improving | At this level, Part 4-1 combines CMMI-DEV levels 4 and 5. Using suitable process metrics, product suppliers control the effectiveness and performance of the product and demonstrate continuous improvement in these areas. |
| 5 | Optimizing | | |

# 5 Practice 1 – Security management

## 5.1 Purpose

The purpose of the security management practice is to ensure that the security-related activities are adequately planned, documented and executed throughout the product's life-cycle.

If care is not taken in planning and supporting the activities related to security, then those activities can be rendered ineffective due to inadequate resources, insufficient time or process inefficiencies. Similarly, misalignment of the product's security needs with related organizational processes such as configuration management, information technology policies and procedures and supply chain management can jeopardize the effectiveness of the secure product development life-cycle.

## 5.2 SM-1: Development process

### 5.2.1 Requirement

A general product development/maintenance/support process shall be documented and enforced that is consistent and integrated with commonly accepted product development processes that include, but are not limited to:

a) configuration management with change controls and audit logging;

b) product description and requirements definition with requirements traceability;

c) software or hardware design and implementation practices, such as modular design;

d) repeatable testing verification and validation process;

e) review and approval of all development process records; and

f) life-cycle support.

## 5.3 Rationale and supplemental guidance

This process is required to ensure that the product supplier has well-defined and proven product development processes in place that can be extended to support the requirements specified by this document. The required processes defined by this document assume the existence of a mature product development life-cycle. Secure product development life-cycles cannot be effective without these processes and rely upon them being in place. Examples of commonly accepted product development processes include ISO 9001 [13] and ISO/IEC 27034 [34] compliant processes.

Having this process means that the product supplier uses techniques during the product development life-cycle that support, as a minimum, configuration management, requirements definition, design, implementation and testing.

## 5.4 SM-2: Identification of responsibilities

### 5.4.1 Requirement

A process shall be employed that identifies the organizational roles and personnel responsible for each of the processes required by this document.

### 5.4.2 Rationale and supplemental guidance

This process is required to ensure that responsibilities are assigned to elements of the product supplier's organization for performing and completing the processes required by this document.

Having this process means that the product supplier's development, maintenance and product support processes required by this document each identify the organizational roles and personnel that are responsible for performing and completing them. The organization and personnel can be within the developer’s organization or external to it.

NOTE A responsible, accountable, consulted and informed (RACI) matrix is an example of a tool that could be used to meet this requirement.

## 5.5 SM-3: Identification of applicability

### 5.5.1 Requirement

A process shall be employed for identifying products (or parts of products) to which this document applies.

### 5.5.2 Rationale and supplemental guidance

This process is required to ensure that the processes in scope as part of this document are applied to the appropriate products as needed and that the correct level of detail is applied.

Having this process means that the product supplier has criteria for identifying which of its products are to be developed, maintained and supported using the processes required by this document. It is envisioned that a product supplier may apply this specification to selected products based on a number of factors, including the marketplace for which a product is intended and whether or not the product requires security to be built into the product and fully evaluated. As an example, certain products or components may not have a security context or provide anonymous access and therefore may not require security to be built into the product. An organization may also base the criteria on the particular features being developed to enhance a product for target markets as long as the common features for all markets remain subject to this standard. Organizations may also use criteria such as applicable security requirements or security risk.

These requirements may be applied to externally provided components or custom developed components from third party suppliers. See 5.11 and 5.12 for more details.

## 5.6 SM-4: Security expertise

### 5.6.1 Requirement

A process shall be employed for identifying and providing security training and assessment programs to ensure that personnel assigned to the organizational roles and duties specified in 5.4, have demonstrated security expertise appropriate for those processes.

### 5.6.2 Rationale and supplemental guidance

This process is required to ensure that personnel involved in security-related processes have adequate expertise for the specific tasks to which they are assigned. Expertise can have been gained by training, experience, seminars, conferences, certifications, etc. This includes technical expertise in defense in depth strategies and related security techniques, and also in the practices, including best practices, required to develop and maintain the product.

Having this process means that personnel assigned to security-related processes have evidence that shows their relevant qualifications. This includes knowledge not only of security, but also for the use of any security-related standards (for example, coding standards), techniques (for example, best practices), and tools (for example, static analysis tools). While security awareness training is vital for everyone involved in the secure product life-cycle, it is generally insufficient for personnel involved in security requirements analysis, design reviews, etc. The security training is role-specific and can vary in formality from informal to formal. Similarly, the personnel assigned to security-related processes have experience (for example, past projects and number of years) that matches the specific security tasks and their specific role.

## 5.7 SM-5: Process scoping

### 5.7.1 Requirement

A process, that includes justification by documented security analysis, shall be employed to identify the parts of this document that are applicable to a selected product development project. Justification for scoping the level of compliance of a project to this document shall be subject to review and approval by personnel with the appropriate security expertise (see 5.6).

### 5.7.2 Rationale and supplemental guidance

Examples include:

a) The product does not include software therefore process requirements applicable to software are out of scope.

b) The threat model indicates that the product does not have any external interfaces or sources of untrusted input (for example, a product with no external connections that can only be accessed in a room with high physical security). In this case, for example, the requirement for fuzz testing external interfaces would not apply.

## 5.8 SM-6: File integrity

### 5.8.1 Requirement

A process shall be employed to provide an integrity verification mechanism for all scripts, executables and other important files included in a product.

### 5.8.2 Rationale and supplemental guidance

This process is required to ensure that product users can verify that executables, scripts, and other important files received from the supplier have not been altered. Common methods of meeting this requirement include cryptographic hashes and digital signatures (which also provide proof of origin).

## 5.9 SM-7: Development environment security

### 5.9.1 Requirement

A process that includes procedural and technical controls shall be employed for protecting the product during development, production and delivery. This includes protecting the product or product update (patch) during design, implementation, testing and release.

### 5.9.2 Rationale and supplemental guidance

This process is required to ensure that the product has not been altered or disclosed in any way during the development process, unless allowed by policy. Loss of integrity of any aspect of the development environment (e.g., the product design and implementation, code signing infrastructure, and software build environment) can negatively affect fielded versions of the product without the knowledge of the organization or its customers. For example, the ability of an attacker to insert an infection in the binary code of a product could lead to that infection being distributed as part of the released product.

Having this process means that the product supplier has mechanisms in place to protect the integrity of design documents, the product implementation (for example, code and user manuals), configuration settings and private keys used for signing software images. For example, application of ISO/IEC 27001 [20] and ISO/IEC 27002 [19] policies and controls can reduce the likelihood of unauthorized access to source code or corruption of source code. They can also reduce the likelihood of unauthorized disclosure of product designs and test results that could be used to compromise fielded versions of the product. Items to specially safeguard include authenticators (e.g., passwords, access control lists, code signing certificates and exploit records collected during defect management).

## 5.10 SM-8: Controls for private keys

### 5.10.1 Requirement

The supplier shall have procedural and technical controls in place to protect private keys used for code signing from unauthorized access or modification.

### 5.10.2 Rationale and supplemental guidance

Private keys are the root of trust, so they require extra protection to ensure that they are not stolen or modified.

## 5.11 SM-9: Security requirements for externally provided components

### 5.11.1 Requirement

A process shall be employed to identify and manage the security risks of all externally provided components used within the product.

### 5.11.2 Rationale and supplemental guidance

This process is required to ensure that supply chain security is addressed for equivalent security practices, latest security updates, security deployment guides and the supplier's ability to respond if a vulnerability is discovered. Supply chain security applies to components which are included within the product and are provided external to the development team responsible for a given product, but do not meet the definition described in 5.12. The security provided by such third-party components is directly related to their role in the product's secure design and defense in depth strategy (see Clause 7).

Having this process means that the product supplier is able to identify when one or more of the following characteristics apply to the use of third-party components in the product:

a) the degree to which the component aligns with the product's security context (see Clause 6) and defense in depth strategy (see Clause 7);

b) the degree of rigor applied to the component's implementation (see Clause 8);

c) the degree of security verification and validation performed on the component by the product supplier or the component supplier (see Clause 9);

d) how to receive and/or monitor notifications about security-related issues from the component supplier (see Clause 10) and patches (see Clause 11); and

e) the sufficiency of security documentation for the component (see Clause 12).

f) the degree that the software is currently supported by the supplier or open source community.

Examples of work items that would satisfy some elements of this requirement include:

– identifying known vulnerabilities in specific versions of open source software components and updating the version of the open source components to the version that fixes the vulnerability;

– evaluating the compliance of vendors of commercial off the shelf (COTS) components to this document or a similar SDL standard; and

– employing compensating mechanisms for known vulnerabilities on COTS or open source components (such as static code analysis).

It is recommended that there be an inventory of components from third party suppliers in order to facilitate defect management (see Clause 6).

For related supply chain requirements, see ISO/IEC 27036-3 [21].

## 5.12 SM-10: Custom developed components from third-party suppliers

### 5.12.1 Requirement

A process shall be employed to ensure that product development life-cycle processes for components from a third-party supplier conform to the requirements used in this document when they meet the following criteria:

a) the components are developed specifically for a single supplier for a specific purpose; and

b) the components can have an impact on security.

### 5.12.2 Rationale and supplemental guidance

This requirement applies when a supplier subcontracts a third-party to specifically develop a component for them which can have security implications. Threat modelling is usually used to determine which components will have security implications.

## 5.13 SM-11: Assessing and addressing security-related issues

### 5.13.1 Requirement

A process shall be employed for verifying that a product or a patch is not released until its security-related issues have been addressed and tracked to closure (see 10.5). This includes issues associated with:

a) requirements (see Clause 6);

b) secure by design (see Clause 7);

c) implementation (see Clause 8);

d) verification/validation (see Clause 9); and

e) defect management (see Clause 10).

### 5.13.2 Rationale and supplemental guidance

This process is required to ensure that the product is not released with security-related issues that have been discovered and whose resolution is not complete and whose severity as defined by a vulnerability scoring system, such as the Common Vulnerability Scoring System (CVSS), is calculated as above the residual risk acceptable within the product security context.

Having this process means that any security-related issue identified during the development and support of a product is documented and addressed to allow the effective security of the product to be determined prior to product release. This would include issues found in all phases such as design review, code review, verification and validation testing, use of static analysis tools, etc.

## 5.14 SM-12: Process verification

### 5.14.1 Requirement

A process shall be employed for verifying that, prior to product release, all applicable security-related processes required by this specification (see 5.7) have been completed with records documenting the completion of each process.

### 5.14.2 Rationale and supplemental guidance

This process is required to ensure that key security practices are being executed.

## 5.15 SM-13: Continuous improvement

### 5.15.1 Requirement

A process shall be employed for continuously improving the SDL. This process shall include the analysis of security defects in component/subsystem/system technologies that escape to the field.

### 5.15.2 Rationale and supplemental guidance

This process is required to ensure that product suppliers improve the rigor of their SDL over time. New security threats are constantly being identified and exploited by attackers so it is important product suppliers help compensate for this by continuously improving their SDL.

Continuous improvement is a well-established and proven method of improving product quality. Since product security issues are a type of quality issue, continuous improvement methodologies are applicable to an SDL. See Annex A for potential metrics related to SDL effectiveness and improvement.

Having this process means that the supplier has a procedure in place to review the process and security defects that escape to the field on a periodic basis and that this procedure includes making improvements to the process as a result of these reviews.

Some examples of activities that would help improve a product supplier's SDL are included in Table 2. Ultimately it is up to suppliers to implement their own means of continuously improving their SDL.

**Table 2 – Example SDL continuous improvement activities**

| Activity | SDL / Security benefits |
|---|---|
| Use a known security vulnerability database to help improve the threat model. For example, if the threat model indicates that the product uses the TLS protocol for transport security, review known vulnerabilities in TLS implementations and ensure these are mitigated in the design. | Improves the threat model by keeping it current with actual security issues observed in the field. |
| Attend external security / SDL conferences or participate in industry SDL groups such as OWASP | Helps a product supplier stay current with emerging threats and SDL best practices. |
| Conduct internal SDL conferences or sessions for sharing of SDL expertise and best practices within the product supplier's organization. | Improve the overall SDL expertise of the product supplier's employees and help them stay current with emerging security threats and SDL best practices. |
| Perform SDL root cause analysis for security vulnerabilities found externally in a supplier's product and identify plus implement corrective action. All SDL practices should be in scope for this analysis. | Root cause analysis and corrective action is a well-established method for improving product quality. Since security issues are quality issues it works well for an SDL too. |
| Combine manual penetration testing with automated tool base testing or use multiple similar security testing tools for SVV-3 Vulnerability testing. | Improves test coverage relative to using a single automated tool. This becomes especially valuable after the existing automated tool stops finding new vulnerabilities. |
| Create fuzzing tools for any protocols for which tools are not available. | Help avoid the scenario where an attacker develops its own fuzzing tool and uses it to find and exploit security vulnerabilities in a product. |
| Train and use dedicated security testing experts for SVV-3 Vulnerability testing. | Since security vulnerability requires extensive and constantly growing expertise, developing and using dedicated experts will improve security test coverage. |

## 6 Practice 2 – Specification of security requirements

### 6.1 Purpose

The processes specified by this practice are used to document the security capabilities that are required for a product along with the expected product security context. Security capabilities can include such items as authentication, authorization, encryption, auditing and other security capabilities a product needs to include. The product security context can include items such as physical security level, protection of external interfaces via a firewall, etc. See [10] for more information on security capability requirements.

These security requirements can be defined at the product-level or they may supplement product-level requirements.

## 6.2 SR-1: Product security context

### 6.2.1 Requirement

A process shall be employed to ensure that the intended product security context is documented.

### 6.2.2 Rationale and supplemental guidance

This process is required to ensure that the minimum requirements of the environment and the assumptions about that environment are documented in order to achieve the security level for which the product was designed. The purpose of defining this information is so that both the developers of the product and the product users have the same understanding about how the product is intended to be used. This will help the developers make appropriate design decisions and the users to use the product as it was intended. Security context could include:

a) location in the network;

b) physical or cyber security provided by the environment where the product will be deployed;

c) isolation (from a network perspective); and

d) if known, potential impact to the environment (for example, loss of life, injury, loss of production, etc.).

For example, it is important to document whether physical security is required. If no physical security is expected to be present, then that may add a number of related requirements such as not allowing pushbutton configuration on the product. Another example is if the product is expected to be protected by a user supplied firewall that connects it to the plant network, the product would typically not require a firewall of its own. Documenting these external security features for the product (its security context) allows developers to design a defense in depth strategy that complements this security context and testers to validate and verify the security of a product in an environment similar to how it should be deployed.

Having this process means that the deployment environment in which the product is intended to be used is correctly represented in all processes involved in the development and testing of this product and are documented.

## 6.3 SR-2: Threat model

### 6.3.1 Requirement

A process shall be employed to ensure that all products shall have a threat model specific to the current development scope of the product with the following characteristics (where applicable):

a) correct flow of categorized information throughout the system;

b) trust boundaries;

c) processes;

d) data stores;

e) interacting external entities;

f) internal and external communication protocols implemented in the product;

g) externally accessible physical ports including debug ports;

h) circuit board connections such as Joint Test Action Group (JTAG) connections or debug headers which might be used to attack the hardware;

i) potential attack vectors including attacks on the hardware, if applicable;

j) potential threats and their severity as defined by a vulnerability scoring system (for example, CVSS);

k) mitigations and/or dispositions for each threat;

l) security-related issues identified; and

m) external dependencies in the form of drivers or third-party applications (code that is not developed by the supplier) that are linked into the application.

The threat model shall be reviewed and verified by the development team to ensure that it is correct and understood.

The threat model shall be reviewed periodically (at least once a year) for released products and updated if required in response to the emergence of new threats to the product even if the design does not change.

Any issues identified in the threat model shall be addressed as defined in 10.4 and 10.5.

### 6.3.2 Rationale and supplemental guidance

This process is required to ensure that security threats for the product are identified, validated, documented, addressed and tested by the product's project team according to the defense in depth strategy.

Having this process means that a threat model for the product is defined and maintained throughout the product life-cycle (for example, as a result of changing threats or updates to the defense in depth strategy) that identifies and describes threats that can occur within the product security context, and against which product is expected to defend itself.

External dependencies are external components or systems that the product depends upon for security. As an example, a product could depend on power for physical security. Or a product could depend on the session management of a web server to be secure. In these examples, failure of the external dependency could lead to a security vulnerability in the product, so mitigations need to be put in place to minimize the chances of such failures. So for the power example, the mitigation could be the installation of an uninterruptable power supply (UPS). In the example of a web server, security should be considered when choosing a web server, and if a secure web server cannot be found, then other compensating measures need to be considered.

Third-party code is an external dependency that can present significant challenges in determining where the threats can occur. If deeply embedded there might not be access to/from this code that crosses a trust boundary. If there is access to the trust boundary, a deeper inspection of the third-party code can be needed.

## 6.4 SR-3: Product security requirements

### 6.4.1 Requirement

A process shall be employed for ensuring that security requirements are documented for the product/feature under development including requirements for security capabilities related to installation, operation, maintenance and decommissioning.

### 6.4.2 Rationale and supplemental guidance

This process is required to ensure that security requirements specific to the product are defined. This includes both technical security requirements (for example, password complexity) and business-oriented security requirements (for example, sensitive data, user authorizations and separation of duties).

Having this process means that the product supplier defines and documents all product security requirements that apply to the life-cycle of the product, including:

a) security privileges required to install, operate, and maintain the product;

b) security options, including removal of default passwords, used to install, configure, operate and maintain the product; and

c) security considerations/actions associated with removing the product from use (for example, removing sensitive data).

NOTE For different capability security levels (SL-C 1 through SL-C 4), IEC 62443-3-3 and IEC 62443-4-2 [11] define the security capability requirements for control systems and components, respectively. These security capabilities are then included as product security requirements for products that are to include these capabilities.

## 6.5 SR-4: Product security requirements content

### 6.5.1 Requirement

A process shall be employed for ensuring that security requirements include the following information:

a) the scope and boundaries of the component or system, in general terms in both a physical and a logical way; and

b) the required capability security level (SL-C) of the product.

### 6.5.2 Rationale and supplemental guidance

If the product is targeted to meet a certain security capability level, it is important to document this as a requirement because it implies that certain security capabilities need to be included in the product. Note that capability security levels and required security capabilities for products are defined in IEC 62443-4-2 [11] and IEC 62443-3-3 [10].

## 6.6 SR-5: Security requirements review

### 6.6.1 Requirement

A process shall be employed to ensure that security requirements are reviewed, updated as necessary and approved to ensure clarity, validity, alignment with the threat model (discussed in 6.3), and their ability to be verified. Each of the following representative disciplines shall participate in this process. Personnel may be assigned to more than one discipline except for testers, who shall remain independent:

a) architects/developers (those who will implement the requirements);

b) testers (those who will validate that the requirements have been met);

c) customer advocate (such as sales, marketing, product management or customer support); and

d) security advisor.

### 6.6.2 Rationale and supplemental guidance

This process is required to ensure that security requirements are valid, understood and testable (or otherwise verifiable).

Having this process means that the product supplier conducts reviews of all security requirements and revises/deletes those that are invalid or that are untestable/unverifiable.

# 7 Practice 3 – Secure by design

## 7.1 Purpose

The processes specified by this practice are used to ensure that the product is secure by design including defense in depth. Defense in depth provides one or more layers of security to thwart security threats. Each layer of the defense in depth strategy is designed to protect the assets from attack in the case that all other layers have been compromised.

The processes required by this practice are required to be applied to all stages of product design, from conceptual design to detailed design, and to all levels of product design from the overall architecture to the design of individual components.

## 7.2 SD-1: Secure design principles

### 7.2.1 Requirement

A process shall be employed for developing and documenting a secure design that identifies and characterizes each interface of the product, including physical and logical interfaces, to include:

a) an indication of whether the interface is externally accessible (by other products), or internally accessible (by other components of the product), or both;

b) security implications of the product security context (see Clause 6) on the external interface;

c) potential users of the interface and the assets that can be accessed through it (directly or indirectly);

d) a determination of whether access to the interface crosses a trust boundary;

e) security considerations, assumptions and/or constraints associated with the use of the interface within the product security context, including applicable threats;

f) the security roles, privileges/rights and access control permissions needed to use the interface and to access the assets defined in c) above;

g) the security capabilities and/or compensating mechanisms used to safeguard the interface and the assets defined in c) above, including input validation as well as output and error handling;

h) the use of third-party products to implement the interface and their security capabilities;

i) documentation that describes how to use the interface if it is externally accessible; and

j) description of how the design mitigates the threats identified in the threat model.

### 7.2.2 Rationale and supplemental guidance

This process is required to ensure that security for access to assets is comprehensively addressed from the perspective of external and internal interfaces of the product through which attacks can be mounted.

Having this process means that interfaces of the product are identified and characterized by the interactions that take place over them (for example, data and control flows), the security mechanisms designed to protect them and the assets that can be compromised if not adequately protected. Interfaces include physical and wireless connections to networks (for example, Ethernet) and devices (for example, keyboards, monitors and USB/compact disc [CD]/digital versatile disc [DVD] media). Logical interfaces support data control flows (for example, application messaging) between product components and include mechanisms such as application programming interfaces (for example, structured query language [SQL]) and communications protocols (for example, the transmission control protocol [TCP]). Protection mechanisms include general hardening capabilities (for example, security policy settings), user access controls (for example, account management), and security event detection and reporting, among others.

Viewing interfaces within the setting provided by the product security context allows the secure design to focus on the specific environment in which the product is expected to operate, including both protections offered by the product security context and vulnerabilities resulting from it (for example, where it can be open to attack). For an internal component of the design, the concept of the product security context is extended to include the security context provided by surrounding product components. For example, the product security context of an application program running on a workstation that is part of an industrial control system product includes the network(s) to which the workstation connects and the software environment of the workstation in which the application runs.

Identifying threats, users, assets and trust boundaries associated with interfaces specifies who is expected to use the interfaces, and indicates where threats and unknown subjects potentially can gain access to the interface and the assets that can be accessed through it. This allows the reduction of the number of interfaces where possible and to provide the appropriate safeguards for the remaining interfaces and the assets that can be accessed through them. Identifying trust boundaries also supports future definition of zones and conduits (see IEC 62443-3-2 [9]), and thus is a primary component in the definition of the security architecture of the product. Sample data assets (resources) include:

a) databases and database tables;

b) configuration files;

c) cryptographic key stores;

d) access control lists (ACLs);

e) registry keys;

f) web pages (static and dynamic);

g) audit logs;

h) network sockets / network media;

i) inter-process communications (IPC), services and remote procedure call (RPC) resources;

j) any other files and directories; and

k) any other memory resource.

Based on analysis of the product security requirements, the product security context and trust boundary considerations, the design can be developed for interfaces that include the definition of user roles, privileges and authorization/access permissions required to use the interfaces as well as specific security capabilities (for example, authentication, encryption and logging) that provide additional safeguards. As part of the design of the defense in depth strategy, the expected use of compensating mechanisms and third-party hardware or software components will aid in the assessment of the adequacy of the defense in depth strategy (see 7.4).

Finally, preparing documentation for the use of externally accessible interfaces (for example, by users and third-parties) reduces the potential for accidental misuse.

## 7.3 SD-2: Defense in depth design

### 7.3.1 Requirement

A process shall be employed to implement multiple layers of defense using a risk based approach based on the threat model. This process shall be employed for assigning responsibilities to each layer of defense.

NOTE 1 Each layer provides additional defense mechanisms.

NOTE 2 It is possible for any layer to be compromised; therefore, secure design principles (see 7.2) are applied to each layer.

NOTE 3 The objective is to reduce the attack surface of the subsequent layers.

### 7.3.2 Rationale and supplemental guidance

For example, the TCP/IP stack could check for invalid packets, an HTTP server could authenticate input and then another layer could validate that the input and audit logs are produced for administrative changes. Each layer provides an additional defense mechanism, has a responsibility and provides attack surface reduction for the next layer. Each layer assumes that the layer in front of it can be compromised.

## 7.4 SD-3: Security design review

### 7.4.1 Requirement

A process shall be employed for conducting design reviews to identify, characterize and track to closure security-related issues associated with each significant revision of the secure design including but not limited to:

a) security requirements (see Clause 6) that were not adequately addressed by the design;

NOTE 1 Requirements allocation, including security requirements, is part of typical design processes.

b) threats and their ability to exploit product interfaces, trust boundaries, and assets (see 7.2); and

c) identification of secure design practices (see 7.5) that were not followed (for example, failure to apply principle of least privilege).

NOTE 2 Characterizing threats and their ability to exploit interfaces is often referred to as threat modelling.

### 7.4.2 Rationale and supplemental guidance

This process is required to ensure that the secure design addresses the requirements and threats (see Clause 6) defined for the product, and that design best practices have been followed (see 7.5). All discovered security-related issues are to be documented and tracked through the processes defined by 7.4 and 10.5.

Having this process means that each version of the design is reviewed to determine:

a) whether any product security requirements have not been adequately addressed by the defense in depth strategy; and

b) whether there are threat vectors (paths for threats to follow) that bypass the defense in depth strategy or that are otherwise capable of breaching the defense in depth strategy.

In either case, the threat model is to be updated to reflect security-related issues discovered as a result of the review process.

## 7.5 SD-4: Secure design best practices

### 7.5.1 Requirement

A process shall be employed to ensure that secure design best practices are documented and applied to the design process. These practices shall be periodically reviewed and updated. Secure design practices include but are not limited to:

a) least privilege (granting only the privileges to users/software necessary to perform intended operations);

b) using proven secure components/designs where possible;

c) economy of mechanism (striving for simple designs);

d) using secure design patterns;

e) attack surface reduction;

f) documenting all trust boundaries as part of the design; and

g) removing debug ports, headers and traces from circuit boards used during development from production hardware or documenting their presence and the need to protect them from unauthorized access.

### 7.5.2 Rationale and supplemental guidance

This process is required to ensure that guidance is provided to developers to help them avoid common pitfalls during design that could lead to later security issues.

Having this process means that the product supplier has a list of security best practices that is maintained and followed during the development of the secure design for the product. These best practices should be based commonly accepted security best practices in industry for the type of product being developed. It is completely up to the supplier to determine which practices they consider to be most appropriate for their design practices. These practices are kept current as a result of both changes in the industry and the application of lessons learned by the product supplier.

Note that these practices apply to both hardware and software design.

## 8 Practice 4 – Secure implementation

### 8.1 Purpose

The processes specified by this practice are used to ensure that the product features are implemented securely.

### 8.2 Applicability

Requirements in this practice apply to all hardware and software components in the product with the exception of externally provided components. For externally provided components, requirement 5.11 applies instead.

### 8.3 SI-1: Security implementation review

#### 8.3.1 Requirement

A process shall be employed to ensure that implementation reviews are performed for identifying, characterizing and tracking to closure security-related issues associated with the implementation of the secure design including:

a) identification of security requirements (see Clause 6) that were not adequately addressed by the implementation;

NOTE Requirements allocation, including security requirements, is part of typical design processes.

b) identification of secure coding standards (see 8.4) that were not followed (for example, use of banned functions or failure to apply principle of least privilege);

c) Static Code Analysis (SCA) for source code to determine security coding errors such as buffer overflows, null pointer dereferencing, etc. using the secure coding standard for the supported programming language. SCA shall be done using a tool if one is available for the language used. In addition, static code analysis shall be done on all source code changes including new source code.

d) review of the implementation and its traceability to the security capabilities defined to support the security design (see Clause 7); and

e) examination of threats and their ability to exploit implementation interfaces, trust boundaries and assets (see 7.2 and 7.3).

### 8.3.2 Rationale and supplemental guidance

This process is required to ensure that the implementation properly addresses (implements) the secure design and its associated security requirements and follows implementation best practices.

Having this process means that the product supplier conducts a comprehensive set of security reviews of the implementation and its design. Different types of reviews will typically be used to address different objectives. For example, manual reviews are typically conducted against the implementation design to verify that requirements are being met and that the implementation will adequately protect against threats expected to be present. In addition, manual source code reviews may be used to examine source code for adherence to best practices (see 8.4), and automated static source code analysis may be used to identify anomalies, including security vulnerabilities in the code as well as non-conformities with given programming rules.

## 8.4 SI-2: Secure coding standards

### 8.4.1 Requirement

The implementation processes shall incorporate security coding standards that are periodically reviewed and updated and include at a minimum:

a) avoidance of potentially exploitable implementation constructs – implementation design patterns that are known to have security weaknesses;

b) avoidance of banned functions and coding constructs/design patterns – software functions and design patterns that should not be used because they have known security weaknesses;

c) automated tool use and settings (for example, for static analysis tools);

d) secure coding practices;

e) validation of all inputs that cross trust boundary.

f) error handling.

### 8.4.2 Rationale and supplemental guidance

This process is required to ensure that guidance is provided to developers to help them avoid common pitfalls during implementation that could lead to later security issues.

Having this process means that the product supplier has a list of security best practices that it maintains and follows during the implementation of a product. These best practices should be based on commonly accepted security best practices in industry for the type of product being developed. It is completely up to the supplier to determine which practices they consider to be most appropriate for their design and coding standard. These practices are kept current as a result of both changes in the industry and the application of lessons learned by the product supplier.

# 9 Practice 5 – Security verification and validation testing

## 9.1 Purpose

The processes specified by this practice are used to document the security testing required to ensure that all the security requirements have been met for the product and that security of the product is maintained when it is used in its product security context and configured to employ its defense in depth strategy.

Security testing can be performed at various times by various personnel during the SDL based on the type of testing and the development model used by the vendor. For example,

fuzz testing could be performed during software development by the software development team and later in the cycle by a test team.

Issues uncovered by testing will be addressed as per “Practice 6 – Security defect management”.

## 9.2 SVV-1: Security requirements testing

### 9.2.1 Requirement

A process shall be employed for verifying that the product security functions meet the security requirements and that the product handles error scenarios and invalid input correctly. Types of testing shall include:

a) functional testing of security requirements;

b) performance and scalability testing; and

c) boundary/edge condition, stress and malformed or unexpected input tests not specifically targeted at security.

### 9.2.2 Rationale and supplemental guidance

This process is required to ensure that the product meets the security requirements defined for it (see Clause 6).

Having this process means that the product supplier verifies through testing that the product meets its documented security requirements.

Examples of the types of functionality in scope for security requirements include:

a) general security capabilities (features);

b) API (application programming interface);

c) permission delegation;

d) anti-tampering and integrity functionality;

e) signed image verification; and

f) secure storage of secrets.

## 9.3 SVV-2: Threat mitigation testing

### 9.3.1 Requirement

A process shall be employed for testing the effectiveness of the mitigation for the threats identified and validated in the threat model. Activities shall include:

a) creating and executing plans to ensure that each mitigation implemented to address a specific threat has been adequately tested to ensure that the mitigation works as designed; and

b) creating and executing plans for attempting to thwart each mitigation.

### 9.3.2 Rationale and supplemental guidance

The effectiveness of mitigations to threats identified by the threat model are tested as part of this practice. Examples of threat mitigation testing include attempts to thwart mitigations identified using the spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege (STRIDE). For example, if STRIDE identified authentication as a mitigation for spoofing threat mitigation tests would focus on bypassing authentication.

If a layered defense strategy is used as a mitigation, then the effectiveness of each layer would be tested. For example, if the product employs the combination of authentication,

authorization and audit logs as a layered defense strategy to thwart tampering, then each layer will be tested for its contribution to this mitigation strategy.

This process is required to ensure that the product's defense in depth and threat mitigation strategies and capabilities are effective.

## 9.4 SVV-3: Vulnerability testing

### 9.4.1 Requirement

A process shall be employed for performing tests that focus on identifying and characterizing potential security vulnerabilities in the product. Known vulnerability testing shall be based upon, at a minimum, recent contents of an established, industry-recognized, public source for known vulnerabilities. Testing shall include:

a) abuse case or malformed or unexpected input testing focused on uncovering security issues. This shall include manual or automated abuse case testing and specialized types of abuse case testing on all external interfaces and protocols for which tools exist. Examples include fuzz testing and network traffic load testing and capacity testing;

b) attack surface analysis to determine all avenues of ingress and egress to and from the system, common vulnerabilities including but not limited to weak ACLs, exposed ports and services running with elevated privileges;

c) black box known vulnerability scanning focused on detecting known vulnerabilities in the product hardware, host or software components. For example, this could be a network based known vulnerability scan;

d) for compiled software, software composition analysis on all binary executable files, including embedded firmware, delivered by the supplier to be installed for a product. This analysis shall detect the following types of problems at a minimum:

   1) known vulnerabilities in the product software components;

   2) linking to vulnerable libraries;

   3) security rule violations; and

   4) compiler settings that can lead to vulnerabilities;

e) dynamic runtime resource management testing that detects flaws not visible under static code analysis, including but not limited to denial of service conditions due to failing to release runtime handles, memory leaks and accesses made to shared memory without authentication. This testing shall be applied if such tools are available.

### 9.4.2 Rationale and supplemental guidance

Void.

## 9.5 SVV-4: Penetration testing

### 9.5.1 Requirement

A process shall be employed to identify and characterize security-related issues via tests that focus on discovering and exploiting security vulnerabilities in the product.

### 9.5.2 Rationale and supplemental guidance

Penetration testing focuses specifically on compromising the confidentiality, integrity or availability of the product. It can involve defeating multiple aspects of the defense in depth design. For example, bypassing authentication to access the product, using elevation of privilege to gain administrative access and then compromising confidentiality by breaking encryption. As this example shows, penetration testing involves approaching testing like an attacker and often involves exploiting chained vulnerabilities in a product.

This process is required to ensure that efforts have been taken to discover security-related issues in the product or product documentation that could allow the product to be exploited.

Having this process means that the product supplier attempts to breach the security of the product through penetration testing. Penetration testing consists of confirming that vulnerabilities in any product capability or the defense in depth strategy can be exploited and used to compromise security of the product. It requires in depth knowledge of the type of product being tested along with security testing tools and techniques. Penetration testing can involve the use of manual techniques, test tools or combinations of the two.

## 9.6 SVV-5: Independence of testers

### 9.6.1 Requirement

A process shall be employed to ensure that individuals performing testing are independent from the developers who designed and implemented the product according to Table 3.

**Table 3 – Required level of independence of testers from developers**

| Test type | Reference | Level of independence |
|---|---|---|
| Security requirements testing | SVV-1 – Security requirements testing | Independent department |
| Threat mitigation testing | SVV-2 – Threat mitigation testing | Independent department |
| Abuse case testing | SVV-3 – Vulnerability testing | Independent person |
| Static code analysis | SI-1 – Security implementation review | None |
| Attack surface analysis | SVV-3 – Vulnerability testing | Independent person |
| Known vulnerability scanning | SVV-3 – Vulnerability testing | Independent person |
| Software composition analysis | SVV-3 – Vulnerability testing | None |
| Penetration testing | SVV-4 – Penetration testing | Independent department or organization |

The levels of independence are defined as follows:

- **None** – no independence required. Developer can perform the testing.
- **Independent person** – the person who performs the testing cannot be one of the developers of the product.
- **Independent department** – the person who performs the testing cannot report to the same first line manager as any developers of the product. Alternatively, they could be a member of a quality assurance (QA) department.
- **Independent organization** – the person who performs the testing cannot be part of the same organization as any developers of the product. An organization can be a separate legal entity, a division of a company or a department of a company that reports to a different executive such as a vice president or similar level.

### 9.6.2 Rationale and supplemental guidance

An independent tester can often find out more, other and different defects than a tester working within a programming team – or a tester who is by profession a programmer. Such a tester brings a different set of assumptions to testing and to reviews, which often helps in exposing the hidden defects and problems. In addition, an independent tester who reports to senior management can report his results honestly and without any concern for reprisal that might result from pointing out problems in co-workers' or, worse yet, the manager's work.

Additional security defects can often be found when a tester's black-box level knowledge of the product is supplemented with white-box level knowledge of a developer acting as an advisor to the tester.

# 10 Practice 6 – Management of security-related issues

## 10.1 Purpose

The processes specified by this practice are used for handling security-related issues of a product that has been configured to employ its defense in depth strategy (see Clause 7) within the product security context (see Clause 6).

## 10.2 DM-1: Receiving notifications of security-related issues

### 10.2.1 Requirement

A process shall exist for receiving and tracking to closure security-related issues in the product reported by internal and external sources including at a minimum:

a) security verification and validation testers;

b) suppliers of third-party components used in the product;

c) product developers and testers; and

d) product users including integrators, asset owners, and maintenance personnel.

NOTE External security verification and validation testers include researchers.

### 10.2.2 Rationale and supplemental guidance

This process is required to ensure that security-related issues/defects discovered by any organization within the product supplier or external organizations (for example, product users and security researchers) can be reported to the product supplier and tracked to closure.

Having this process means that the product supplier defined instructions for reporting security-related issues (see ISO/IEC 30111 [23]) to it. For reports from external entities, the product supplier will have incident response processes such as those identified in ISO/IEC 29147 [22] for receiving vulnerability reports about supported products and interacting with the entity that reported the issue.

Guidelines for reporting security-related issues are to be readily accessible to each of the potential internal and external sources of these reports. Awareness training, product documentation and support websites are all potential ways to communicate this information. These guidelines include:

a) information needed to facilitate validation;

b) how to protect the confidentiality of and access to the information being reported;

c) the degree of communications with the entity that reported the security-related issue;

d) timelines for reporting internally discovered security-related issues in released products; and

e) a strategy for handling third-party component vulnerabilities discovered internally.

## 10.3 DM-2: Reviewing security-related issues

### 10.3.1 Requirement

A process shall exist for ensuring that reported security-related issues are investigated in a timely manner to determine their:

a) applicability to the product;

b) verifiability; and

c) threats that trigger the issue.

NOTE Timeliness is driven by market forces.

### 10.3.2 Rationale and supplemental guidance

This process is required to ensure that security-related issues reported to the product supplier are examined to determine that they are applicable to the product, are verifiable, and that the cause of the issue (such as the threat(s)) is understood.

Having this process means that the product supplier verifies all security issues reported to it. Perceived security-related issues can be unsubstantiated or not applicable to the product, so there needs to be a process to verify and examine reported vulnerabilities (see ISO/IEC 30111).

For security-related issues in components maintained by the product developer, this process can involve such activities as attempting to reproduce the reported vulnerability or examining the third-party embedded source code's usage within the product. For security-related issues in components maintained by a third-party, this process can be as straightforward as comparing the version of the third-party binary with the versions to which the patch applies.

## 10.4 DM-3: Assessing security-related issues

### 10.4.1 Requirement

A process shall be employed for analysing security-related issues in the product to include:

a) assessing their impact with respect to:
   1) the actual security context in which they were discovered;
   2) the product's security context (see Clause 6); and
   3) the product's defense in depth strategy (see Clause 7);
b) severity as defined by a vulnerability scoring system (for example, CVSS);
c) identifying all other products/product versions containing the security-related issue (if any);
d) identifying the root causes of the issue; and
e) identifying related security issues.

For root cause analysis, a methodical approach such as that described in IEC 62740 [25] may be employed.

### 10.4.2 Rationale and supplemental guidance

This process is required to ensure that the potential impact of security-related design issues is examined and understood to support decisions related to how they will be addressed.

Having this process means that the product supplier assesses the potential impact and severity of each security-related issue, determines whether the issues exist in other products or versions (for example, by using the same or similar components) and identifies the root causes of the issue. Completing such an assessment provides the basis for determining how to address the issue (see 10.5), and which development life-cycle processes, such as redesign activities and threat model updates, may be involved in the resolution.

NOTE Risk assessments can be used in this evaluation of security-related issues.

Verifiable security-related issues can vary widely in their security impact and their distribution within the product, so there needs to be a process for characterizing each issue so that an appropriate resolution can be determined.

It is recommended that the process identify conditions that would exit the security defect management process such as duplicate, non-security, and third-party security-related issues (see ISO/IEC 30111). The impact assessment should also take into consideration additional factors such as the scope of affected product users, the potential for collateral damage, the availability of exploits and (for control systems) the potential impact to essential functions (see IEC 62443-3-3).

The impact assessment may be as simple as a qualitative rating (for example, low, medium and high), a more quantitative method based on likelihood and consequence or a standardized method such as the CVSS. A security-related issue that is associated with a widely used design pattern or implementation method can be symptomatic of a larger problem. In such a situation, the impact assessment associated with the vulnerability should address the combined impact of all instances rather than dealing with each instance in isolation.

## 10.5 DM-4: Addressing security-related issues

### 10.5.1 Requirement

A process shall be employed for addressing security-related issues and determining whether to report them based on the results of the impact assessment (see 10.4). The supplier shall establish an acceptable level of residual risk that shall be applied when determining an appropriate way to address each issue. Options include one or more of the following:

a) fixing the issue through one or more of the following:
   1) defense in depth strategy or design change;
   2) addition of one or more security requirements and/or capabilities;
   3) use of compensating mechanisms; and/or
   4) disabling or removing features;
b) creating a remediation plan to fix the problem;
c) deferring the problem for future resolution (reapply this requirement at some time in the future) and specifying the reason(s) and associated risk(s);
d) not fixing the problem if the residual risk is below the established acceptable level of residual risk.

In all cases, the following shall be done as well:

a) inform other processes of the issue or related issue(s), including processes for other products/product revisions; and
b) inform third parties if problems found in included third-party source code.

When security-related issues are resolved recommendations to prevent similar errors from occurring in the future shall be evaluated.

This process shall include a periodic review of open security-related issues to ensure that issues are being addressed appropriately. This periodic review shall at a minimum occur during each release or iteration cycle.

NOTE When the resolution decision is to fix the security-related issue in the product implementation, the timing of the release of the fix can result in a patch (see Clause 12) or the fix can be deferred until the next release.

### 10.5.2 Rationale and supplemental guidance

This process is required to ensure that a determination is made for how to handle (address) each security-related issue and that no security-related issue is inadvertently overlooked or ignored.

Having this process means that the product supplier reviews the potential residual risk of each security-related issue and makes a justifiable decision for how to handle (address) it.

Residual risk can be determined using many different methods. An example would be to start with CVSS score, but then add other security controls and countermeasures not accounted for in CVSS such as whether the issue is applicable to the product's security context.

The process for deciding upon and implementing a resolution to a security-related issue needs to address these considerations (see ISO/IEC 30111) because of their potential impact:

a) a proposed resolution can violate a premise of the secure design that other aspects of the product rely upon;

b) a proposed resolution can be unnecessary because of a mismatch between the reporting entity's security context and the product security context;

c) a proposed resolution can only partially reduce the impact of the security-related issue, may take an unacceptably long time to implement because of its complexity, or may be so unusable that it is likely to be disabled; and

d) a proposed resolution can introduce side-effects that are unacceptable.

Timeliness for determining and implementing a resolution based on the impact of the security-related issue will typically align with market forces and may drive establishment of clear interfaces to related organizational processes (for example, legal, customer service and public relations) to avoid unnecessary delays.

## 10.6 DM-5: Disclosing security-related issues

### 10.6.1 Requirement

A process shall be employed for informing product users about reportable security-related issues (see 10.5) in supported products in a timely manner with content that includes but is not limited to the following information:

a) issue description, vulnerability score as per CVSS or a similar system for ranking vulnerabilities, and affected product version(s); and

b) description of the resolution.

NOTE 1 The description of the resolution can include references to installation of patches (see Clause 12).

NOTE 2 Timeliness is driven by market forces.

The strategy for handling third-party component vulnerabilities discovered by the product developer should take into account the possibility of premature public disclosure by the third-party component supplier.

### 10.6.2 Rationale and supplemental guidance

This process is required to ensure that product users are informed of resolved security-related issues that have been designated as reportable. Reportable resolutions are typically those that are related to released products and whose issue severity is deemed high enough to report by the product supplier. Product users need this information to make informed security assessments about their operations, and service providers use this information to handle vulnerabilities as part of their event management capability (see IEC 62443-2-4).

Having this process means that the product supplier has procedures for determining which security issues are reportable, and reporting resolutions for reportable issues to the users of the product. The disclosure process will typically include provisions for informational alerts in addition to vulnerability notices. For example, informational alerts can be used to notify product users of compensating mechanisms in response to current cyber security activity or to inform product users that the product is not susceptible to a highly publicized vulnerability. See IEC 29147 [22] for information regarding content of notifications.

### 10.7 DM-6: Periodic review of security defect management practice

#### 10.7.1 Requirement

A process shall be employed for conducting periodic reviews of the security-related issue management process. Periodic reviews of the process shall, at a minimum, examine security-related issues managed through the process since the last periodic review to determine if the management process was complete, efficient, and led to the resolution of each security-related issue. Periodic reviews of the security-related issue management process shall be conducted at least annually.

#### 10.7.2 Rationale and supplemental guidance

This process is required for continuous improvement of the issue management practice.

## 11 Practice 7 – Security update management

### 11.1 Purpose

The processes specified by this practice are used to ensure that security updates associated with the product are tested for regressions and made available to product users in a timely manner.

### 11.2 SUM-1: Security update qualification

#### 11.2.1 Requirement

A process shall be employed for verifying that

1) security updates created by the product developer address the intended security vulnerabilities;
2) security updates do not introduce regressions, including but not limited to patches created by:
   a) the product developer;
   b) suppliers of components used in the product; and
   c) suppliers of components or platforms on which the product depends.

The process should include a confirmation that update is not contradicting other operational, safety or legal constraints.

#### 11.2.2 Rationale and supplemental guidance

This process is required to ensure that patches applicable to the product are evaluated to ensure that they do not adversely affect operation of the product.

Having this process means that qualification of patches (typically via testing) is performed to verify that patches applicable to the product do not directly or indirectly (for example, via side effects) compromise the product's operation or defense in depth strategy (see Clause 7). Documentation about this process may be used by the service provider to address the patch management requirements of IEC 62443-2-4.

### 11.3 SUM-2: Security update documentation

#### 11.3.1 Requirement

A process shall be employed to ensure that documentation about product security updates is made available to product users that includes but is not limited to:

a) the product version number(s) to which the security patch applies;

b) instructions on how to apply approved patches manually and via an automated process;

c) description of any impacts that applying the patch to the product can have, including reboot;

d) instructions on how to verify that an approved patch has been applied; and

e) risks of not applying the patch and mediations that can be used for patches that are not approved or deployed by the asset owner.

### 11.3.2 Rationale and supplemental guidance

This process is required to ensure that security patches are documented to allow approved patches to be installed and non-approved patches to be remediated.

Having this process means that the product supplier provides or otherwise makes documentation available that identifies and describes applicable security patches, how to install approved patches, how to determine patch status (whether a patch has been applied) of components and how to mediate non-approved patches. See the patch management requirements of IEC 62443-2-4 for more information.

## 11.4 SUM-3: Dependent component or operating system security update documentation

### 11.4.1 Requirement

A process shall be employed to ensure that documentation about dependent component or operating system security updates is made available to product users that includes but is not limited to:

a) stating whether the product is compatible with the dependent component or operating system security update; and

b) for security updates that are unapproved by the product vendor, the mitigations that can be used in lieu of not applying the update.

### 11.4.2 Rationale and supplemental guidance

End users are hesitant to install software in an IACS that might upset operations. As a result, vendors need to provide information to the users about whether a particular security update of the operating system is compatible with the product.

## 11.5 SUM-4: Security update delivery

### 11.5.1 Requirement

A process shall be employed to ensure that security updates for all supported products and product versions are made available to product users in a manner that facilitates verification that the security patch is authentic.

### 11.5.2 Rationale and supplemental guidance

This process is required to ensure that product users can obtain applicable security patches for the product in a timely manner and to reduce the possibility that the security patches are fraudulent (see Clause 11).

Having this process means that the product supplier provides a mechanism or technique that allows product users to verify the authenticity of patches. Concurrent release of patches for all supported versions can reduce the time window between awareness of the vulnerability and the availability of patches.

### 11.6 SUM-5: Timely delivery of security patches

#### 11.6.1 Requirement

A process shall be employed to define a policy that specifies the timeframes for delivering and qualifying (see 11.2) security updates to product users and to ensure that this policy is followed. At a minimum, this policy shall consider the following factors:

a) the potential impact of the vulnerability;

b) public knowledge of the vulnerability;

c) whether published exploits exist for the vulnerability;

d) the volume of deployed products that are affected; and

e) the availability of an effective mitigation in lieu of the patch.

#### 11.6.2 Rationale and supplemental guidance

Security updates typically have target release timing which is based on the factors listed in this requirement. For example, some companies classify patches as required to be addressed within 30 days, 60 days or 90 days or longer of being found.

## 12 Practice 8 – Security guidelines

### 12.1 Purpose

The processes specified by this practice are used to provide user documentation that describes how to integrate, configure, and maintain the defense in depth strategy of the product in accordance with its product security context (see Clause 6). IEC 62443-2-4 defines complementary hardening requirements for the use of this documentation by IACS service providers.

Applying and maintaining the defense in depth strategy for a specific installation will typically address the following:

a) policies and procedures associated with the product security context, as defined in Clause 6;

b) architectural considerations, such as firewall placement and use of compensating mechanisms including security measures, as defined in Clause 7;

c) configuring security settings/options, such as configuring firewall rules, delegation, certificate management, and managing user accounts (for example, setting their privileges/permissions); and

d) use of tools to assist in the hardening.

NOTE Patching is not included in this list, but is addressed in Clause 11.

The remainder of Clause 12 defines requirements for development processes used to produce and maintain this documentation. Supporting these requirements means that the product supplier has identifiable processes for creating, maintaining and delivering documentation that describes how to harden the product.

### 12.2 SG-1: Product defense in depth

#### 12.2.1 Requirement

A process shall exist to create product user documentation that describes the security defense in depth strategy for the product to support installation, operation and maintenance that includes:

a) security capabilities implemented by the product and their role in the defense in depth strategy;

b) threats addressed by the defense in depth strategy; and

c) product user mitigation strategies for known security risks associated with the product, including risks associated with legacy code.

### 12.2.2 Rationale and supplemental guidance

This process is required to ensure that documentation for the defense in depth strategy is produced to support hardening of the product at the customer site. Such documentation is required by IEC 62443-2-4, that defines security requirements for IACS installation and maintenance service providers.

Having this process means that the product supplier documents various aspects of the defense in depth strategy necessary to harden the product during installation and keep it hardened during its lifetime of use. Aspects of the defense in depth strategy to be documented include the residual threats that are expected to be present and capable of attacking the product, the security capabilities of the product to safeguard it against these threats and any compensating security controls/mitigations that can be used with the product to further protect the product.

## 12.3 SG-2: Defense in depth measures expected in the environment

### 12.3.1 Requirement

A process shall be employed to create product user documentation that describes the security defense in depth measures expected to be provided by the external environment in which the product is to be used (see Clause 6).

NOTE These measures can also come from 10.5.

### 12.3.2 Rationale and supplemental guidance

This process is required to ensure that documentation for the defense in depth strategy is produced to support hardening of the product at the customer site. Such documentation is required by IEC 62443-2-4, that defines security requirements for IACS installation and maintenance service providers.

Having this process means that the product supplier documents various aspects of the defense in depth strategy necessary to harden the product during installation and keep it hardened during its lifetime of use.

## 12.4 SG-3: Security hardening guidelines

### 12.4.1 Requirement

A process shall be employed to create product user documentation that includes guidelines for hardening the product when installing and maintaining the product. The guidelines shall include, but are not limited to, instructions, rationale and recommendations for the following:

a) integration of the product, including third-party components, with its product security context (see Clause 6);

b) integration of the product's application programming interfaces/protocols with user applications;

c) applying and maintaining the product's defense in depth strategy (see Clause 7);

d) configuration and use of security options/capabilities in support of local security policies, and for each security option/capability:

   1) its contribution to the product's defense in depth strategy (see Clause 7);

   2) descriptions of configurable and default values that include how each affects security along with any potential impact each has on work practices; and

3) setting/changing/deleting its value;

e) instructions and recommendations for the use of all security-related tools and utilities that support administration, monitoring, incident handling and evaluation of the security of the product;

f) instructions and recommendations for periodic security maintenance activities;

g) instructions for reporting security incidents for the product to the product supplier; and

h) description of the security best practices for maintenance and administration of the product.

#### 12.4.2 Rationale and supplemental guidance

This process is required to ensure that instructions that describe how to harden the product and keep it hardened are documented. Such documentation is required by IEC 62443-2-4 that defines security requirements for IACS installation and maintenance service providers.

Having this process means that the product supplier creates user documentation that provides directions for hardening the product during installation and for keeping it hardened during the lifetime of the product use. This requirement recognizes that the security policies and requirements for customer sites are often different, and as a result, instructions for securely integrating the product into the customer site, configuring it appropriately and maintaining its security are necessary.

### 12.5 SG-4: Secure disposal guidelines

#### 12.5.1 Requirement

A process shall be employed to create product user documentation that includes guidelines for removing the product from use. The guidelines shall include, but is not limited to, instructions and recommendations for the following:

a) removing the product from its intended environment (see Clause 6);

b) including recommendations for removing references and configuration data stored within the environment;

c) secure removal of data stored in the product; and

d) secure disposal of the product to prevent potential disclosure of data contained in the product that could not be removed as described in c) above.

#### 12.5.2 Rationale and supplemental guidance

This process is required to ensure that instructions that describe how to securely take the product out of use (decommission it) are documented. Such documentation is required by IEC 62443-2-4, that defines security requirements for IACS installation and maintenance service providers.

Having this process means that the product supplier creates user documentation that provides directions for sanitizing the product of sensitive, confidential and/or proprietary data and software.

### 12.6 SG-5: Secure operation guidelines

#### 12.6.1 Requirement

A process shall be employed to create product user documentation that describes:

a) responsibilities and actions necessary for users, including administrators, to securely operate the product; and

b) assumptions regarding the behaviour of the user/administrator and their relationship to the secure operation of the product.

### 12.6.2 Rationale and supplemental guidance

This process is required to ensure that instructions that describe the secure use of the product during its operation and administration are included in the security guidelines.

Having this process means that the product supplier creates user/administrator documentation that provides instructions for using the product securely. In general, this represents a set of best practices for the secure use of the product. For example, this could include guidelines for certificate management, password management and other authentication mechanisms.

## 12.7 SG-6: Account management guidelines

### 12.7.1 Requirement

A process shall be employed to create product user documentation that defines user account requirements and recommendations associated with the use of the product that includes, but is not limited to:

a) user account permissions (access control) and privileges (user rights) needed to use the product, including, but not limited to operating system accounts, control system accounts and data base accounts; and

b) default accounts used by the product (for example, service accounts) and instructions for changing default account names and passwords.

### 12.7.2 Rationale and supplemental guidance

This process is required to ensure that requirements for the user accounts necessary to use the product are defined and documented.

Having this process means that the product supplier creates documentation that defines accounts and their settings, including default accounts, that are needed to use the product.

## 12.8 SG-7: Documentation review

### 12.8.1 Requirement

A process shall be employed to identify, characterize and track to closure errors and omissions in all user manuals including the security guidelines to include:

a) coverage of the product's security capabilities;

b) integration of the product with its intended environment (see Clause 6); and

c) assurance that all documented practices are secure.

### 12.8.2 Rationale and supplemental guidance

This process is required to ensure that the security-related documentation for the product is accurate and complete and that non-secure practices are not documented in other user documentation.

Having this process means that the product's security-related documentation is reviewed to determine whether any product security capabilities have not been correctly or adequately addressed, and whether the documentation adequately describes how the product's defense in depth strategy is to be integrated with the product security context; and if discrepancies are found, that a process exists for addressing them.

# Annex A
(informative)

# Possible metrics

Subclause 5.15 requires that the development organization takes steps to continuously improve its process. Using metrics that show the effectiveness of the development process is helpful to determine if measurable improvements are being made. The specific metrics to be collected (if any) are up to the product developer and may vary significantly, but some examples are included in this Annex A.

As an example, the baseline security posture score (SPS) can be calculated as follows:

NOTE 1 The actual method used by the vendor can vary. Using a scale of 0-100 with 100 being the worst, multiple factors are averaged together to determine the score.

a) **Functional security** – Based on security functions defined in IEC 62443-4-2

EXAMPLE 1 *NSF – NRM* normalized to 100

where

*NSF* is the number of security functions defined in IEC 62443-4-2;

*NRM* is the number of such requirements met.

b) **Implementation security** –

1) based on the results of SCA

EXAMPLE 2 $\left(1-\left(\frac{NSCA}{NSR}\right)\right)\times 100$

where

*NSCA* is the number of SCA security rules enabled returning 0 warnings;

*NSR* is the total number of security rules.

NOTE 2 This can be averaged over multiple modules.

2) based on the results of banned.h

EXAMPLE 3 $\left(1-\left(\frac{P}{TP}\right)\right)\times 100$

where

*P* is the number of projects linking to banned.h;

*TP* is the number of total projects required to link.

c) **Build security** – Based on the compiler options and flags

EXAMPLE 4 $\left(1-\left(\frac{B}{C}\right)\right)\times 100$

where

*B* is the number of clean BinScope component reports;

*C* is the number of components.

d) **Deployment security** – Based on the results of an analysis of the attack surface of the product

EXAMPLE 5 $\left(1-\left(\frac{A}{N}\right)\right)\times 100$

where

*A* is the number of mitigated attack vectors;

*N* is the total number of attack vectors.

e) **Current backlog** – Based on the number of critical or important security defects in the product

EXAMPLE 6 MIN ((50 · *CSI*) + (10 · (*ISI*), 100)

where

*CSI* is the number of critical security issues;

*ISI* is the number of important security issues.

f) **Training** – Based on the assessment scores of the engineers

EXAMPLE 7 $\left(1-\left(\frac{CA}{SE}\right)\right)\times 100$

where

*CA* is the number of completed assessments;

*SE* is the total number of software engineers.

g) **SDL violations** – Based on deviations to the SDL secure coding guidelines

EXAMPLE 8 $\left(1-\left(\frac{CI}{CT}\right)\right)\times 100$

where

*CI* is the number of secure code compliance checklist items in compliance;

*CT* is the total number of secure code compliance items in checklist.

NOTE 3 Item number varies for code type such as Web, C++, Managed, etc.

# Annex B
(informative)

# Table of requirements

Table B.1 summarizes the requirements in order to give an overview of this document.

**Table B.1 – Summary of all requirements**

| Requirement number and name |
|---|
| SM-1: Development process |
| SM-2: Identification of responsibilities |
| SM-3: Identification of applicability |
| SM-4: Security expertise |
| SM-5: Process scoping |
| SM-6: SM-6: File integrity |
| SM-7: Development environment security |
| SM-8: Controls for private keys |
| SM-9: Security requirements for externally provided components |
| SM-10: Custom developed components from third-party |
| SM-11: Assessing and addressing security-related issues |
| SM-12: Process verification |
| SM-13: Continuous improvement |
| SR-1: Product security context |
| SR-2: Threat model |
| SR-3: Product security requirements |
| SR-4: Product security requirements content |
| SR-5: Security requirements review |
| SD-1: Secure design principles |
| SD-2: Defense in depth design |
| SD-3: Security design review |
| SD-4: Secure design best practices |
| SI-1: Security implementation review |
| SI-2: Secure coding standards |
| SVV-1: Security requirements testing |
| SVV-2: Threat mitigation testing |
| SVV-3: Vulnerability testing |
| SVV-4: Penetration testing |
| SVV-5: Independence of testers |
| DM-1: Receiving notifications of security-related issues |
| DM-2: Reviewing security-related issues |
| DM-3: Assessing security-related issues |
| DM-4: Addressing security-related issues |
| DM-5: Disclosing security-related issues |
| DM-6: Periodic review of security defect management practice |
| SUM-1: Security update qualification |

| Requirement number and name |
|---|
| SUM-2: Security update documentation |
| SUM-3: Dependent component or operating system security update documentation |
| SUM-4: Security update delivery |
| SUM-5: Timely delivery of security patches |
| SG-1: Product defense in depth |
| SG-2: Defense in depth measures expected in the environment |
| SG-3: Security hardening guidelines |
| SG-4: Secure disposal guidelines |
| SG-5: Secure operation guidelines |
| SG-6: Account management guidelines |
| SG-7: Documentation review |

# Bibliography

This bibliography includes references to sources used in the creation of this document as well as references to sources that can aid the reader in developing a greater understanding of cyber security as a whole and developing a management system. Not all references in this bibliography are referred to throughout the text of this document. The references have been broken down into different categories depending on the type of source they are.

**References to other parts, both existing and anticipated, of the IEC 62443 series:**

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

[2] IEC TR 62443-1-2[5], *Security for industrial automation and control systems – Part 1-2: Master glossary of terms and abbreviations*

[3] IEC TS 62443-1-3[6], *Security for industrial automation and control systems – Part 1-3: System security compliance metrics*

[4] IEC TR 62443-1-4[7], *Security for industrial automation and control systems – Part 1-4: IACS security life-cycle and use-cases*

[5] IEC 62443-2-1, *Industrial communication networks – Network and system security – Part 2-1: Establishing an industrial automation and control system security program*

[6] IEC TR 62443-2-2[8] , *Security for industrial automation and control systems – Part 2-2: Implementation guidance for an IACS security management system*

[7] IEC TR 62443-2-3, *Security for industrial automation and control systems – Part 2-3: Patch management in the IACS environment*

[8] IEC TR 62443-3-1:2009, *Industrial communication networks – Network and system security – Part 3-1: Security technologies for industrial automation and control systems*

[9] IEC 62443-3-2[9], *Security for industrial automation and control systems – Part 3-2: Security risk assessment and system design*

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

[11] IEC 62443-4-2[10], *Security for industrial automation and control systems – Part 4-2: Technical security requirements for IACS components*

**Other standards references:**

[12] ISO/IEC Directives, Part 2, *Principles and rules for the structure and drafting of ISO and IEC documents*

---

5 Under consideration.

6 Under consideration.

7 Under consideration.

8 Under consideration.

9 Under preparation. Stage at the time of publication: IEC/CDV 62443-3-2:2017.

10 Under preparation. Stage at the time of publication: IEC/CDV 62443-4-2:2017.

[13] ISO 9001, *Quality management systems – Requirements*

[14] ISO/IEC 10746-1, *Information technology – Open distributed processing – Reference model: Overview*

[15] ISO/IEC 10746-2, *Information technology – Open distributed processing – Reference model: Foundations*

[16] ISO/IEC 15408-1, *Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model*

[17] ISO/IEC 15408-2, *Information technology – Security techniques – Evaluation criteria for IT security – Part 2: Security functional components*

[18] ISO/IEC 15408-3, *Information technology – Security techniques – Evaluation criteria for IT security – Part 3: Security assurance components*

[19] ISO/IEC 27002, *Information technology – Security techniques – Code of practice for information security controls*

[20] ISO/IEC 27001, *Information technology – Security techniques – Information security management systems – Requirements*

[21] ISO/IEC 27036-3, *Information technology – Security techniques – Information security for supplier relationships – Part 3: Guidelines for information and communication technology supply chain security*

[22] ISO/IEC 29147, *Information technology – Security techniques – Vulnerability disclosure*

[23] ISO/IEC 30111, *Information technology – Security techniques – Vulnerability handling processes*

[24] IEC 61508 (all parts), *Functional safety of electrical/electronic/programmable electronic safety-related systems*

[25] IEC 62740, *Root cause analysis (RCA)*

[26] ISCI SDLA-300, *Security Development Life-cycle Assurance – Certification Requirement*, Version 1, Revision 3

[27] ISCI EDSA-312, *Embedded Device Security Assurance – Certification Requirements Specifications – Software Development Security Assessment*

[28] ISCI EDSA-310, *Embedded Device Security Assurance – Certification Requirements Specifications – Requirements for embedded device robustness testing*, Version 2.2DO-178B, *Software Considerations in Airborne Systems and Equipment Certification*

[29] NIST Special Publication 800-30, *Guide for Conducting Risk Assessments*

[30] NIST Special Publication 800-37, *Guide for Applying the Risk Management Framework to Federal Information Systems*

[31] NIST Special Publication 800-55, *Performance Measurement Guide for Information Security*

[32] NIST Special Publication 800-61, *Computer Security Incident Handling Guide*

[33] NIST Special Publication 800-82, *Guide to Industrial Control System (ICS) Security*

[34] NIST Process Control Security Requirements Forum (PCSRF) and the Future of Industrial Control System Security

[35] ISO/IEC 27034 (all parts), *Information technology – Security techniques – Application security*

**Industry-specific and sector-specific references:**

[36] OWASP Comprehensive, Lightweight Application Security Process (CLASP), available at <http://www.owasp.org/index.php/Category:OWASP_CLASP_Project> [viewed 2017-08-17]

[37] Report on the Evaluation of Cybersecurity Self-assessment Tools and Methods, November 2004, ChemITC, available at <http://www.chemicalcybersecurity.org/> [viewed 2017-08-17]

[38] U.S. Chemicals Sector Cyber Security Strategy, September 2006, available at <http://www.chemicalcybersecurity.org/> [viewed 2017-08-17]

[39] Building Security In Maturity Model, September 2011, Gary McGraw, Ph.D., Brian Chess, Ph.D., & Sammy Migues, available at <http://www.bsimm.com/> [viewed 2017-08-17]

**Other documents and published resources:**

[40] CARLSON, Tom, Information Security Management: Understanding ISO 17799, 2001

[41] Carnegie Mellon Software Engineering Institute, *CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing,* Version 1.1, Staged Representation, 2002, CMU/SEI-2002-TR-012

[42] Carnegie Mellon Software Engineering Institute, *CMMI for Development (CMMI-DEV),* Version 1.3, 2010, CMU/SEI-2010-TR-033

**Books:**

[43] HOWARD, Michael and LIPNER, Steve, The Security Development Life-cycle; SDL A Process for Developing Demonstrably More Secure Software, 2006, Microsoft Press, Redmond, Washington

[44] McGRAW, Gary, Software Security Building Security In, 2006, Addison-Wesley, Upper Saddle River, NJ

_____

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel:  + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch

This page has been left intentionally blank.