TECHNICAL REPORT

ISO/IEC TR 6114

First edition
2023-10

Cybersecurity — Security considerations throughout the product life cycle

Cybersécurité — Considérations relatives à la sécurité tout au long du cycle de vie du produit

Reference number
ISO/IEC TR 6114:2023(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *Information security, cybersecurity and privacy protection*.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

The globalization of technology design, development, manufacturing, and distribution has created an environment of complicated supply chains with limited transparency. This presents an incredible challenge for the industry and highlights a growing need to ensure product integrity for all stages of the information and communications technology (ICT) product life cycle.

The call for assurance across the supply chain landscape has evolved over several decades. More recently, policy makers around the world have begun to focus on supply chain risks in new ways: from policies considering supply chain security risks for government procurement to various initiatives adding security considerations such as trust and transparency in the supply chain for ICT.

Vendors have been doing their part as well. Over the past several years, ICT suppliers have taken important steps towards increasing supply chain transparency. These steps include sourcing conflict-free minerals,[1] and implementing a set of policies, procedures and tools at factories to improve security consideration throughout the supply chain by validating where and when each component of an ICT product was manufactured.

These are important first steps, however they primarily focus on the production stage, just one stage of the ICT product life cycle. In today’s complex environment, hardware platform providers are expected to enable a full range of tools and solutions that improve security consideration across the entire life cycle, from design and sourcing to secure retirement.

Security considerations throughout the product life cycle (SCLC) establish an end to end framework that can be applied to the multi-year life cycle of ICT products to comprehend and address potential risks for improved transparency and higher levels of security assurances. By enabling transparency and assurances across the ICT product life cycle, supply chain owners can improve platform integrity, resilience and security. The life cycle phases are both iterative and recursive in nature.

Cybersecurity — Security considerations throughout the product life cycle

1 Scope

This document describes security considerations throughout the product life cycle (SCLC), which is a framework that spans the entire information and communications technology (ICT) product life cycle. The aim of the framework is to align the industry and bring greater transparency to customers at every point on the ICT product life cycle.

This document describes the following items for suppliers, end users (consumers), intermediaries of the ICT supply chain, service providers, and regulators:

- definition of phases in the ICT product life cycle from concept to retirement;
- threat vectors possible in each phase of the life cycle;
- potential controls against those threat vectors.

The target audiences of this document are suppliers and consumers of ICT products, including all participants throughout the supply chain such as silicon chip designers, fabricators, product assemblers, logistics providers, service providers, and information security organizations. Clauses 5 to 11 target an organization’s strategic and risk management teams. This document provides an end-to-end view of the threats in each phase to help the organization shape their plans, procedures and policies.

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.

ISO/IEC/IEEE 15288, *Systems and software engineering — System life cycle processes*

ISO/IEC/IEEE 24748-1:2018, *Systems and software engineering — Life cycle management — Part 1: Guidelines for life cycle management*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC/IEEE 15288, ISO/IEC/IEEE 24748-1:2018, and the following apply.

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

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

3.1
digital signature
data appended to, or a cryptographic transformation of, a data unit that allows the recipient of the data unit to verify the source and integrity of the data unit

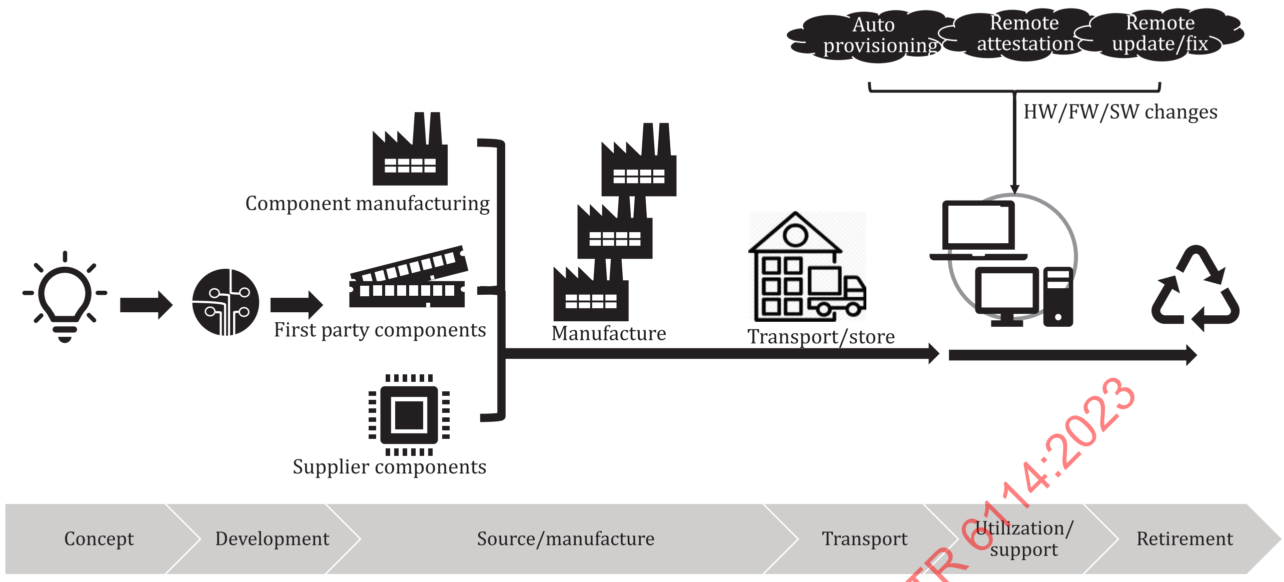
[SOURCE: ISO/IEC 9798-3:2019, 3.3]

3.2
code scanner
program source code and binary file security analysis tool

3.3
hardware trojan
malicious program or hardware that masquerades as a benign application

4 Abbreviated terms

ASIC	application specific integrated circuit
BOM	bill of materials
CPU	central processing unit
DRM	digital rights management
DSP	digital signal processor
ERP	enterprise resource planning
FPGA	field programmable gate array
FW	firmware
HDL	hardware description language
HW	hardware
IC	integrated circuit
ICT	information and communication technology
ISV	independent software vender
OEM	original equipment manufacture
OSAT	outsourced semiconductor assembly and test
OS	operating system
PUF	physically unclonable function
SaaS	software as a service
SDL	security development life cycle
SoC	system on chip
SW	software
VHDL	very high-speed integrated circuit hardware description language



SOURCE ISO/IEC/IEEE 24748-1:2018, reproduced with the permission of the authors.

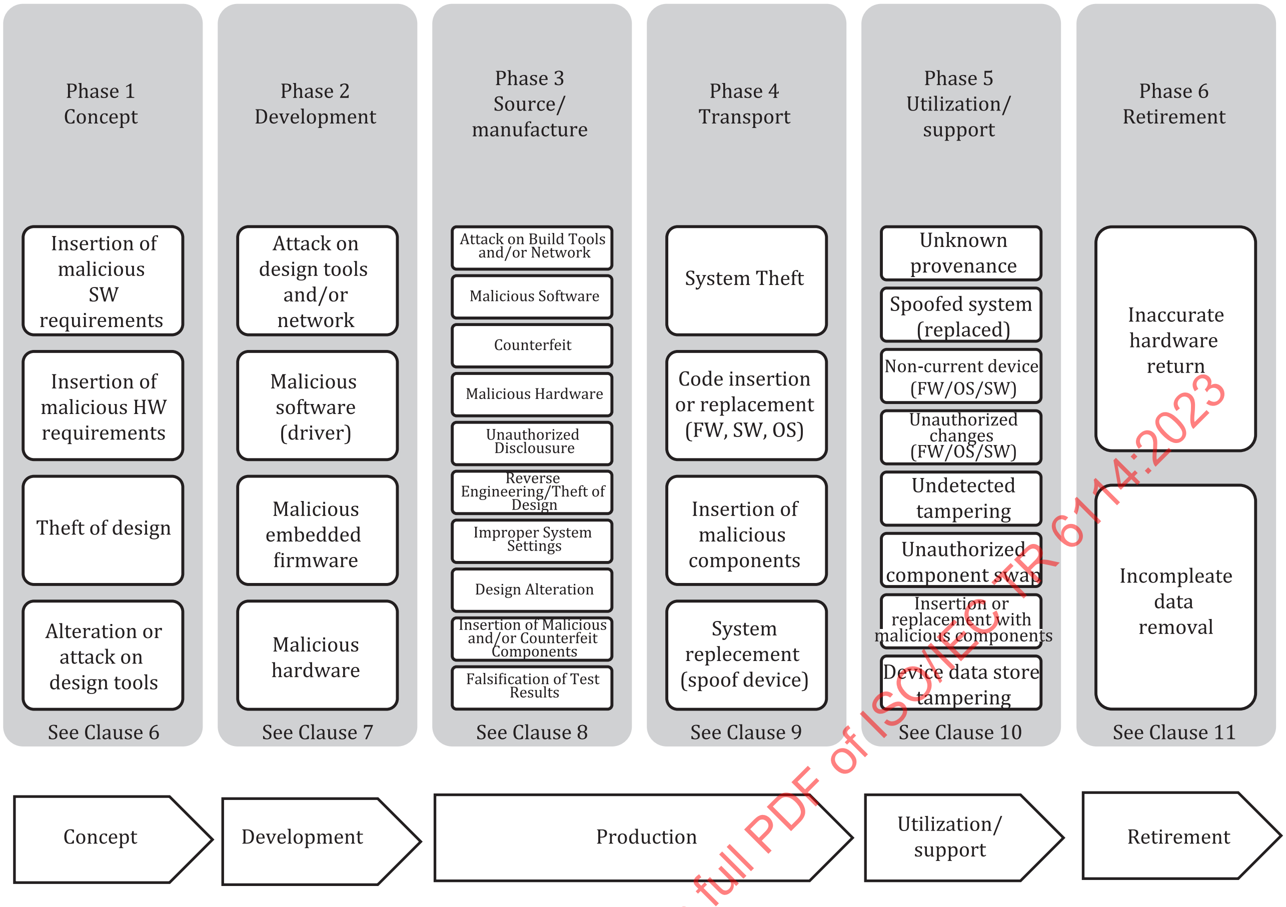
Figure 1 — ICT system life cycle

5 Security considerations throughout the product life cycle

5.1 Security considerations throughout the product life cycle overview

Security considerations throughout the product life cycle (SCLC) is a framework to describe the ICT products life cycle from security assurance and manufacturing perspectives (see Figure 1). The concept of SCLC can be applied to hardware products and software such as microcode, firmware, and other software to support the hardware. SCLC consists of six common product life cycle phases which are shown in Figure 2 with typical threat vectors. The names of the phases are derived from ISO/IEC/IEEE 24748-1 and processes unique to SCLC have been added. It is important to note that each phase exists for individual components, sub-systems and end products. It is both possible and highly likely that some of the phases are executed more than once for an individual component (e.g. integrated circuits go through wafer fabrication, sorting, assembly and final test. Those facilities, if separate, introduce multiple build/transfer phases).

ISO/IEC/IEEE 15288 defines the life cycle of a system by four process groups which are 1) agreement process, 2) organizational project-enabling process, 3) technical management process and 4) technical process. Since this document describes the life cycle from a different perspective, that is, the “product life cycle”, there is no conflict or inconsistency between ISO/IEC/IEEE 15288 and this document. Another systems approach to security is the NIST SP 800-160V1 that aligns with this approach.[3]


Phase 1
Concept
Insertion of malicious SW requirements
Insertion of malicious HW requirements
Theft of design
Alteration or attack on design tools
See Clause 6
Phase 2
Development
Attack on design tools and/or network
Malicious software (driver)
Malicious embedded firmware
Malicious hardware
See Clause 7
Phase 3
Source/ manufacture
Attack on Build Tools and/or Network
Malicious Software
Counterfeit
Malicious Hardware
Unauthorized Disclousure
Reverse Engineering/Theft of Design
Improper System Settings
Design Alteration
Insertion of Malicious and/or Counterfeit Components
Falsification of Test Results
See Clause 8
Phase 4
Transport
System Theft
Code insertion or replacement (FW, SW, OS)
Insertion of malicious components
System replecement (spoof device)
See Clause 9
Phase 5
Utilization/ support
Unknown provenance
Spoofed system (replaced)
Non-current device (FW/OS/SW)
Unauthorized changes (FW/OS/SW)
Undetected tampering
Unauthorized component swap
Insertion or replacement with malicious components
Device data store tampering
See Clause 10
Phase 6
Retirement
Inaccurate hardware return
Incompleate data removal
See Clause 11
Concept
Development
Production
Utilization/ support
Retirement

SOURCE ISO/IEC/IEEE 24748-1:2018, reproduced with the permission of the authors.

Figure 2 — ICT Product life cycle and threat vectors

Figure 3 shows a relationship of technical processes in ISO/IEC/IEEE 15288 and SCLC phases.

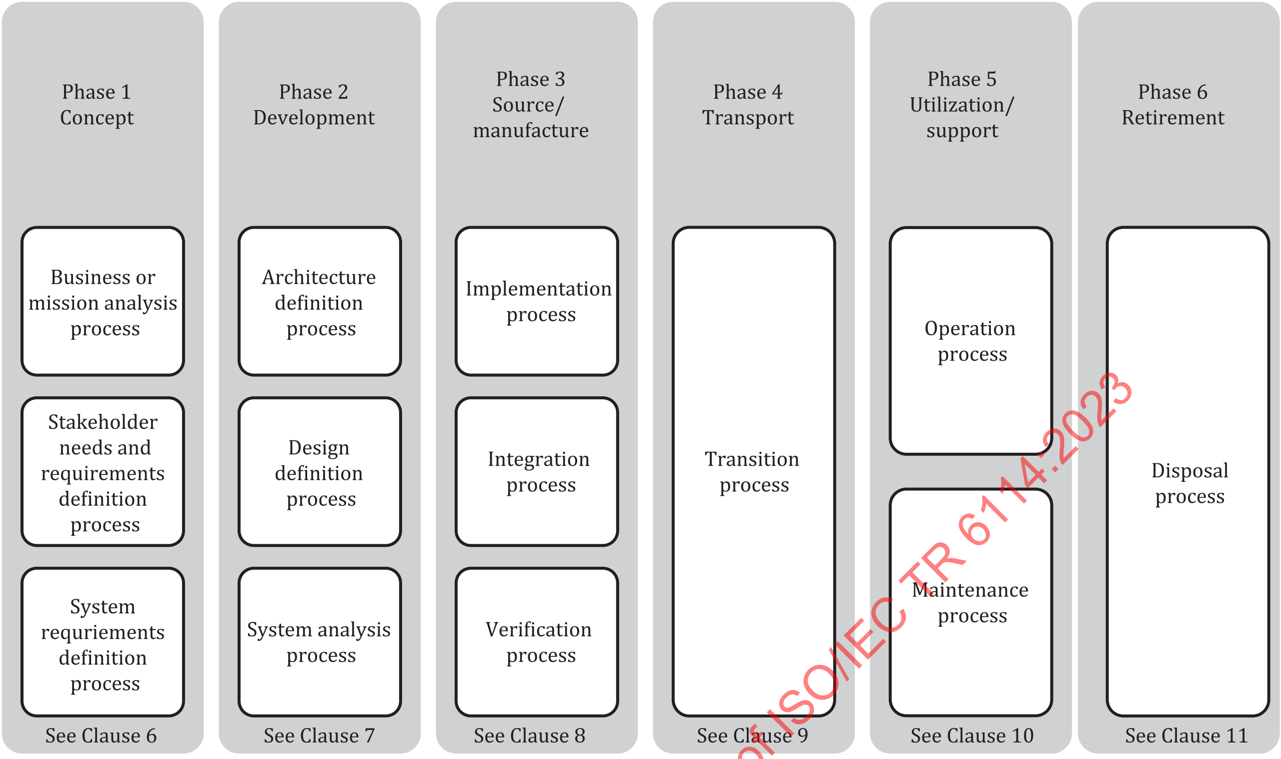


Figure 3 — Mapping technical processes of ISO/IEC/IEEE 15288 and SCLC life cycle phases

5.2 Information and communication technology threat model

Security threats against ICT products exist not only in the utilization/support phase but in all six phases shown in Figure 2.[4] Each phase has different threat vectors, but some vectors such as theft occur in multiple phases. This means responsible parties are expected to apply proper measures in each phase for each threat vector applicable.

From Clause 6 to Clause 11, the key characteristics of each ICT product's life cycle phase and typical threat vectors are described.

5.3 Classes of threats

There are multiple classes of security threats. The first type is an attack that targets unintentional vulnerability, such as ICT product security vulnerability in the manufacturing process or in the physical storage and transportation processes. The second type is an intentional attack by insider personnel who can access the process and change or replace tools, products, or components. The focus of this document is to identify and protect ICT products from the first type of threats. However, the methods described in this document can also be used to mitigate the risks of the second type of threats. In many cases, additional protection measures are necessary, such as employee/personnel management and access control in all phases for all stakeholders who participate in the product life cycle.

5.4 Structure of the report

Clauses 6 to 11 give an overview of each phase, threats, and controls. Additionally, Annexes A, B, C and D provide specific best practice procedures and controls for practitioners that they can implement within their operation to mitigate against potential threats.

6 Phase 1: Concept

6.1 General

In the concept phase, user needs are identified, and system concepts are described and evaluated. In addition to the user needs, companies are expected to consider regulatory controls and security objectives to establish a comprehensive set of requirements to be met in development.

Given the complexity of ICT products, many companies combine internal design, third party design and open source design to deliver a single component on a system that can comprise hundreds of components. For example, the system on a chip (SoC) component can contain hardware blocks sourced from many third parties, as well as blocks designed by the product manufacturer. This complexity introduces additional risks as unneeded features from external components that are incorporated into the end product, possibly creating a vulnerability that can be attacked. For this reason, the product flows back through the concept phase to add additional security control requirements based upon decisions made in development.

It is best practice for ICT product organizations to adopt a security development life-cycle (SDL) process that conforms to ISO/IEC 27034-1. This process ensures the identification of unique threats at the start of the concept phase and turns those into a set of requirements to be met during the development phase. Additionally, SDL requires the training of the engineers, designers and architects that work on how to best maintain a secure system, thereby increasing the security of the product.

SCLC in the concept phase addresses the questions:

- What is in the products function?
- Does the product contain anything additional that can be used against the owner/operator of the ICT product?

In 6.2, possible threat vectors and mitigations in this phase are described.

6.2 Summary of concept threats and controls

6.2.1 Workflow toolchain tampering

Management of the software tools used in support of the concept phase is critical to mitigating attacks to the tool that can alter the design or allow feature requirements to be stolen. These tools often support integration with third-party plugins, such as those from outside tool vendors or even from the original tool provider. If the plugins are not properly verified/certified, they can contain malicious elements. Additionally, installation of these modules can be subject to man-in-the-middle (MITM) attacks during transit over the network or in the build of the tool software by the software provider. The modules require sufficient verification before installation into the concept toolchain. It is essential to implement a practice for managing design tools and keeping them up to date and secure. Best practice ensures these plugins have been digitally signed and licensed by a reputable producer, are regularly scanned by source code scanners (static and dynamic) and have an internal team responsible for securing the design tools. Moving to a SaaS workflow toolchain vendor properly skilled in secure life cycle practices and verified by an independent third party can also be an effective way to improve controls on the toolchain. However, this comes with risks, as the SaaS providers are a likely target of attack due to the multitude of customers in one spot. The ICT product developer is responsible for securing their workflow toolchain and for assessing their own internal security controls against that of a SaaS provider to determine the best way to implement their workflow toolchain.

In cases where the network used in the concept phase is not a closed network, attacks can be launched against these networks from external adversaries in the hopes of obtaining or modifying sensitive design files. Such attacks can be supported by an insider threat that installs malware or a virus on critical security systems, allowing for easier access and attacks against target platforms. Refer to ISO/IEC 27032 for guidance on building and maintaining secure networks, as this issue is the same for any ICT entity that runs a private network.

6.2.2 Unauthorized operations

The unauthorized invocation of data operations for creating, reading, updating/modifying or deleting data in the concept phase can impact the development of the product, its components or sub systems. For example, a security feature developed in concept phase, but which has been deleted/rejected or changed and which influences the product design and build to not include a security feature necessary to protect the product in operation or add a backdoor into the product for future attacks once in operation.

6.2.3 Integrity faults

The integrity faults can occur by unintentional modification or destruction of data due to technical or operational errors, faults or failures occurring within a product, or related to supporting the product's concept phase. Like unauthorized operations, this can lead to a manipulation or exclusion of requirements for the development phase, thereby altering the function of the ICT product. These faults can get caught in development if the appropriate test cases are in place to validate the correct requirements. However, if they are not caught, or caught too late in the life cycle, additional costs are incurred as the product is looped back into the development phase to remove the vulnerability.

6.2.4 Theft or loss

The absence, removal or destruction of a data asset, or unauthorized data access due to actions taken by a malicious actor, or by environmental hazards can occur within the concept phase.

7 Phase 2: Development

7.1 General

This is the second phase of the ICT product life cycle, where engineering design and development are conducted. Example processes are prototype design and evaluation, engineering sample development and manufacturing, threat analysis, manufacturing tool development and production operation planning.

7.2 Summary of development threats and controls

7.2.1 Attacks on development tools and/or network

The same security threats and prevention mechanisms described in 6.2.1 can be applied to this phase.

7.2.2 Malicious embedded firmware

Hardware that has any intelligence requires firmware, so the hardware knows how to operate. Firmware typically sits in non-volatile memory and is not authenticated prior to execution making it vulnerable to attack through alterations. This characteristic of firmware makes firmware susceptible to attack at many stages in the SCLC. This subclause deals with malicious firmware during the design and creation of firmware by the original designer. It is important to note that manufacturers of ICT products can take the firmware from the hardware manufacturer and change it for their unique need or ignore it all together and replace it with their own firmware or that of another third party.

Due to the fluidity of firmware from the original product designer throughout the supply chain, it is important for firmware to be maintained and tracked with good source code management tools and version control processes. Firmware not only changes within the supply chain, prior to provisioning, but it is not uncommon for it to change throughout phase 5 and even into phase 6. Firmware changes in the utilization phase typically occur to address performance and security updates in the hardware that were not discovered in validation.

In design, all firmware can include a process for adding a digital signature to the firmware prior to distribution so the ICT process can verify the signature and only load the firmware if the signature is valid. The signature ensures the firmware has not been tampered with in phase 2. Combining this process with the security design life cycle practices can improve confidence in the validity of the firmware.

7.2.3 Malicious hardware

Malicious hardware (HW) can be introduced by an internal or external source adding additional circuitry into a design that performs a function other than what was specified in the product requirements. Best practices to mitigate the implementation of malicious HW are the implementation of peer code reviews and the utilization of code scanners. Additionally, characterizing the normal behaviour of the HW enables the implementation of test cases that test for abnormal behaviours.

Malicious HW results in additional circuitry added to the design. The hardware performs all the functionality to meet the design requirements but has additional circuitry that can activate only after a specific condition exists (number of executions, amount of time since power on, etc.). It is almost statistically impossible to exhaustively test with every possible input vector. The types of hardware trojans are broad, but it is technically possible to send data back to an unauthorized party. Another hardware trojan can invoke a destructive action or prevent any action during the operation of the product.

7.2.4 Malicious software (driver)

Software drivers and firmware typically come from the hardware supplier, but there are instances where these are provided by a third party. This can be done when a single software driver is controlling components from multiple hardware manufacturers, and it requires the integration of multiple sets of software into a single signed software module. This software is susceptible to the same design threats for software implemented internally, like the insertion of hidden functionality. To mitigate this threat, when selecting suppliers it is important to consider the SDL and supply chain controls followed by the supplier. Additionally, validation results can be made available as part of the product delivery to ensure that there can be independent verification of code scans results and test results and to ensure no abnormal behaviour. Requiring only signed software as a standard practice enables software traceability directly back to the signing organization and possible verification that no manipulation has occurred. Performing static code scans of each version of software provided will give an additional internally verifiable result to the state of the software provided.

Malicious software can also occur from software tampering. Some common types of software tampering mechanisms are described in Annex F.

7.2.5 Counterfeit

Counterfeit components can appear to operate within the requirements established by stakeholders, but can suffer from substandard materials and insufficient quality controls and validation, leading to unpredictable life. This can lead to higher failure rates, inoperability within the required environment (heat, moisture, vibration, etc.) and the inability to hold the product manufacturer accountable.

One best practice which can be implemented in the supply chain is to buy only from distributors that are authorized (by the manufacturer of the component). Another best practice is to perform routine audits of the suppliers on their security practices to prevent counterfeit parts from entering their supply chain. Audits can include observation of supply chain practices, destructive tests on components and verification of validation results.

8 Phase 3: Source and manufacture

8.1 General

This is the phase where the materials and components are manufactured into ICT products. There are many threats in the production process that can impact the performance of the product or expose intellectual property for theft. Like other phases, phase 3 requires an implementation of good practices in information technology processes as well as supply chain processes, to ensure consistency.

8.2 Source

Source is the acquisition of goods or services from a third party, which contributes to the ICT component or product. This is the phase to prepare necessary materials and tools for the production. The materials include not only raw materials, components, software, but also services like storage, building and validation. Companies typically implement a comprehensive sourcing strategy that includes establishing security criteria in addition to their corporate responsibility and quality criteria to be used in the selection of the supplier. The parties responsible for supply chain assurance and information security define the criteria. This means the criteria can be used to align supplier expectations with the goals to prevent threat vectors occurring at earlier points of the life cycle. Additionally, controls through audits or quantifiable assurance (documented proof) can be added as part of the criteria for supplier selection and ongoing monitoring.

8.3 Manufacture

Manufacture is the phase to integrate final products from the materials prepared in the previous source phase. During the integration and at the completion of the manufacturing, a series of tests and evaluation can be applied. In some cases, middle products can be transferred from one facility to the other. In that case, phase 4 protection is applied. At the end of the manufacturing, it is important to configure the products as “end of manufacturing” state, which disables all manufacturing interfaces such as debug and test ports.

8.4 Summary of production threats and controls

8.4.1 Attack on production tools, data exchange tools and/or network

The same security threats and prevention mechanisms described in 6.2.1 can be applied to this phase.

In ICT product manufacturing, most fabrication equipment is very sophisticated, and the equipment is serviced by the equipment manufacturer’s technicians. There are two methods of maintenance. The first method is through the network, in which case access controls are expected to be very tight and typically are opened and closed by the builder rather than leaving a business-to-business connection open for someone to take advantage of the open connection to affect the performance of the equipment. Alternatively, technicians can come into the factory to service equipment. In this case, the technician can be supervised or electronically monitored while in the factory to protect assets from theft, manipulation, or insertion of foreign materials into the build process.

8.4.2 Unauthorized disclosure

Most of the design files used in the creation of components are only shared under the strictest of non-disclosure agreements (NDAs). However, violations of these NDAs, either intentional or unintentional, are not always reported by the violating party. This expands the insider threat potential beyond just the manufacturer to include every supplier or vendor that is given sensitive information about the manufacturer’s products. Methods to minimize accidental or purposeful disclosure include:

- applying digital rights management (DRM) to design files with policy settings that restrict viewing files unless specifically granted by the owner of the file, thereby limiting physical access to the manufacturing floor;

— training technicians and engineers on how to properly handle information and designs they readily use to perform their jobs.

8.4.3 Reverse engineering / theft of design

Any design that can be created can also be reverse engineered. Even if an attacker cannot gain direct access to the design files, extraction of the synthesized files (processed designs) can allow for possible reverse engineering of critical design secrets. This can include cryptographic keys stored within the products. Detection of a stolen design can be enabled by obfuscating the cryptographic key by dispersing it throughout the circuitry (vs. having fuses in sequential placement) or having the algorithm generate a unique hash based upon the unique chip design and including this as part of the key.

8.4.4 Improper system settings

Final validation of finished products can include steps to ensure proper system state before it leaves the manufacturing facility. System settings that are open in manufacturing to enable the correct configuration of an ICT product are expected to be locked prior to leaving the end of the manufacturing line.

8.4.5 Design alteration

Alteration of the design can occur during a number of stages by various actors. It is possible for a supplier to suggest a seemingly innocuous change that actually provides the ability to induce an undefined state of execution in the system. An integrator can also make a covert change to the design files provided as part of an integration effort, and a substitute component can be swapped out due to supply shortages or in an effort to save costs.

To mitigate impact by suppliers, a contractual agreement can be implemented to ensure that every change to a material goes through a formal engineering change order (ECO) process and all parties can weigh in on the change and determine the impact before the change is implemented. All material testing performed by the supplier is expected to be made available for inspection if the designer sees anomalies in product quality or performance.

To mitigate against changes between the designer/OEM and the manufacturer, contract wording is expected to disallow alteration and define a process for the manufacturer to engage the designer/OEM for temporary modification of design due to unforeseen conditions, such as natural disasters impacting design supply of approved components. In addition, the designer/OEM is expected to require as-built BOM information to be provided for each of the ICT products built so that it can be compared against the engineering and manufacturing BOMs to ensure they match. The ability to perform teardown verification of every material in the product is a means to sample check conformance to the design.

8.4.6 Insertion of malicious and/or counterfeit components

A malicious component is a device that has been made to look like and function identically to a valid component but has some level of additional functionality that performs a malicious action when triggered. A malicious component can also be a lower value or functionality component that has been intentionally used to replace higher value or functionality component, after all validation testing has occurred. One example is that attackers can seek to replace original memory components with slower memory, tarnishing the name of the suppliers (given that faster memory costs more than slower memory, but both perform the same functions).

A counterfeit component is a device that is expected to function like the legitimate device but can lack in quality, performance of rigor of validation. These devices can be easily distinguishable when powered on due to power draw or throughput deviations from the norm, yet they are easily inserted during later stages of the phase 4 when only visual validation is performed.

To mitigate these risks, the manufacturer’s sourcing processes are expected to be verified to ensure sourcing from only authorized distributors with rigorous controls in place. Additionally, the manufacturers’ internal processes are expected to place controls on the movement of materials, components and products in and out of the manufacturing floor with warehouse controls like cameras and locked cages to prevent malicious or counterfeit components from entering their facilities. With ICT products, the additional electronic verification of the final platform can be performed to detect anomalies with individual units. An example attack case (use of tagalongs) is described in Annex E.

8.4.7 Falsification of test results

During the test phase, an attacker can modify the results of the testing performed. This can be done to prevent detection of malicious changes made to the component or to mask the presence of trojan circuitry injected into a component.

To mitigate against falsifications, processes similar to software best practices can be applied to individual test cases. It is possible to implement a regular comparison of factory implemented test cases against a set of master test cases to detect modifications (hash compare, code compare). Additionally, test program management systems can be a means to ensure all required test cases are being executed so an attacker does not turn off a test case such that the failure is not recorded. The best practices of separation of duties can also be implemented (e.g. a test engineer does not have authorization to update/delete test results storage, only insertion).

8.4.8 Product theft

Product theft is a common threat in the ICT product life cycle. Given the value of the product compared to their relative size, products can become targets for theft, especially during periods of supply shortages.

The key to mitigate against theft is to minimize the value of the ICT product while in transit. This can be done by cryptographically sealing the product with the key being passed to the end user or business separately from the delivery of the product. Additionally, the ability to “brick” the product (turn permanently non-operational) by the owner will result in the product being worth only the recycle value. Additionally, it is important that the product and key valued components (solid state drive, memory, CPU, etc.) all have the capacity to be turned non-operational or sealed so the product is not harvested for the valuable components inside.

8.4.9 Code insertion or replacement (firmware, operating system, software)

It is also possible for an attacker to replace valid firmware images with malicious images or to make alterations to the existing firmware. This is especially true for firmware that is not signed or integrity-checked by a trusted element on the component.

To mitigate this threat, one practice that can be employed in phase 3 is to take measurements of all firmware, operating system and software that includes a hash of the software and any identity information. The data can be stored in the cloud utilizing a unique product identity like Trusted Computing Group (TCG) Platform Certificate.[7] Then a method can be provided for the owner to retake measurements of the system and compare them against the values stored in the cloud to detect any discrepancies. Another practice employed by large IT organizations is to reinstall all firmware, operating system and software from their secure software vault. This neutralizes the impact by an attack on firmware, operating system and software while the product is in transport.

8.4.10 System replacement (spoof device)

This can occur where the entire product has been cloned and is trying to be passed off as OEM product.

To mitigate against these threats the OEM can implement a platform certificate process and a utility for the end user to run to see if their certificate is valid. This still requires the user to run the validation process, but implementation of a method for the OEM to verify authenticity will help ensure brand reputation is not harmed through the introduction of sub-standard products being passed as original.

9 Phase 4: Transport

9.1 General

To mitigate risks in this phase, the sender can implement physical protection mechanisms such as shrink-wraps, physical seal and electrical protection mechanism such as product and software configuration change detection mechanism. By recording hardware and software configuration digital signature at the end of the manufacturing process, the receiving party can confirm any change on the system by checking the digital signatures.

This is the phase where the products are shipped from the manufacturer to other locations, which can include physical and logical product transportation, storage and export/import customs process as required. Since multiple stakeholders can participate in this phase, multiple threat actors and vectors are expected to be considered. This phase can occur during other phases such as source or manufacture. In some cases, ownership of the products is transferred from a sender (e.g. manufacturer) to a recipient (e.g. end users).

It includes sourcing of materials and services, product validation and delivery to the end user which can include physical and logical products transportation, storage and export/import customs process if it is required.

9.2 Summary of production threats and controls

9.2.1 Product theft

See 8.4.8.

9.2.2 Code insertion or replacement (firmware, operating system, software)

See 8.4.9.

9.2.3 Insertion of malicious components

See 8.4.6.

9.2.4 System replacement (spoof device)

See 8.4.10.

9.2.5 Physical attack in storage and transit

This can occur between product shipment by manufacture and receiving by customer. Usually, the sender is responsible for this phase and delegates to a shipping organization by buying services such as transportation, storage and custom process.

10 Phase 5: Utilization and support

10.1 General

This phase comes after the production of the product, as it is put into operation by the owner of the product or their service provider. Many of the same threats in this phase can occur as part of support, which is another SCLC phase that operates at the same time. Differences exists in who the bad actor would be and the ease by which they can affect change on the ICT product.

10.2 Provision

This is the first phase of utilization where the purchaser of the ICT product or their service provider takes physical control of the ICT product. In this phase, it is important to verify that the product matches what was procured from the producer. Additionally, it is advised to scan the product for any possible malicious software loaded on the product in the phase 3 (which includes packing and shipping to the customer or distributor).

10.3 Utilization

This is the phase in which the product is performing its intended function. This phase can be the longest phase in the SCLC. To keep the product performing and protected from vulnerabilities found after the product shipment, it can be necessary to maintain the product through software or firmware updates (patch). In this phase, end users are responsible for keeping the product secure and stable. Manufacturers are responsible for providing security mitigation and guidance to end users.

Once provisioned, the ICT product is installed and typically connected to a network to perform its intended functions. The same ICT product can be used for different purposes throughout its useful life. If the ICT product is reprovisioned to perform new functions, the provisioning phase is executed again to ensure it is properly set to perform the new function. For example, a cloud data centre can typically be required to change configuration and software settings when reprovisioning a server from previous configuration to identity management for a highly secure entity. However, in many cases, this reprovisioning of a server does minimize possibility of an attack.

10.4 Support

This is the phase in which the system is upgraded, repaired or patched once it has entered phase 4. This phase can be very short and, in some cases, exists at the same time as utilization with updates that run in the background while performing its intended use. Support can include automated down-the-wire updates through a network connection (wired or wireless), or physical touch updates and repairs through loading of software directly onto the system or opening the system and inserting, removing, or replacing components of that system. Repairs of hardware components can be achieved through updating its driver or firmware software. This is more typical of system component repairs dealing with security or logic flaws that can be leveraged to control or alter the intended functions of the system.

10.5 Summary of utilization threats and controls

10.5.1 Unknown provenance

Unknown provenance can occur in a building system, platform, and infrastructure without clear security guideline. Unknown provenance device or products can be installed with or without intentional ignorance of security measures.

To mitigate risk, the party responsible for this phase can establish a clear guideline for devices and products that are connected or integrated into the system, platform and infrastructure to make sure the device is safe to connect to the system. The best practice is to not use products of unknown origin with the system.

10.5.2 Spoofed system (replaced system)

This can occur any time in the utilization phase by replacing the hardware, software, or firmware. There are different types of spoofing, i.e. spoofing at software level or replacing a piece of hardware.

To mitigate the risk, the party responsible for utilization can implement protection mechanisms in physical layer such as access restrictions and in ICT network layers. This can be done by implementing the integrity check of each device utilizing hardware base “Root of Trust”.

10.5.3 Undetected tampering

An attacker gains access to the system and successfully modifies it to perform undesired functions. This can be done by gaining access to the system while unattended by the user. Additionally, this can be done by gaining access to the building that houses the system through a compromised service provider like physical security or cleaning services. It can also be done by an insider within the organization.

10.5.4 Build data store tampering

The digital representation of what is contained in the system at time of manufacture is the build data store. An attacker can modify this file to remove, insert or change a record of the actual components assembled into the product, making the use of this data in quality or security excursion investigation ineffective. This can be done more easily by an insider within the organization, but it can be done after the fact by any third party with access to the system housing the build data.

To mitigate tampering, one can employ encryption at rest, thus adding a layer of security within the data file itself. Additionally, a hash of the file or a copy of the file can be stored outside the storage solution with a verification step against that copy or hash every time the build data are accessed with an alert occurring if the hash or file do not match. An additional countermeasure involves block-chaining the build data at each process step. Block-chaining would permit pre-manufacture integrity checking of the process step order, the data/meta-data associated with each step, and facilitate forensic investigation of a tampering incident to determine the targeted functionality within the product and potentially make attribution to the threat actor.

10.5.5 Non-current device/product (firmware, operation system, application, drivers)

Non-current devices or products are those where the system or component vendor has provided an update to address an unintended operation or close a security issue that was discovered after the product was put into use by the customer. Without maintaining current firmware, software, or OS, an attacker can leverage that vulnerability to compromise the system performance or introduce undesired functions on the system.

To mitigate the impact of an out-of-date product software, an organization can create processes to regularly review system manufacturers site for latest updates and have an established process to verify those updates in a lab and perform a test to ensure the update does not introduce quality or reliability issues. Once an update is verified, the organization records the most up-to-date state of that system, then installs the updates to systems. Additionally, implementation of a system verification check on a regular cadence can verify that the system was successfully updated or drive a forced update based upon the organization’s security policy.

10.5.6 Unauthorized changes (firmware, operating system, software)

An attacker can leverage the support mechanisms to alter microcode, firmware, OS or application software to perform unintended functions or alter the existing functions to do something different than intended or make them non-functional.

To mitigate these types of attacks, manageability software can verify the software size, version, signature to an image stored off the system or in a secured storage within the system itself. If the image is stored on the system itself, then the validation can take place prior to connecting the system to the network, thereby minimizing the ability of the tampered system to infect other systems on the shared network. A best practice can be to run these checks on a periodic basis and not rely on a system start-up, as more and more systems run continuously or have standby modes that would not initiate system start-up.

10.5.7 Unauthorized component swap

An attacker can leverage the support processes to gain access to the system and replace the component that impacts functions. Example of such attacks can be removing high speed memory with lower grade memory or replacing storage with lower capacity.

10.5.8 Insertion or replacement with malicious component

An attacker can insert or replace components after all validation testing has occurred or leverage the support process to release malicious component firmware, operating system and software updates into a system. For example, an integrator can change to design files as part of integration or where an upstream antivirus service update server is compromised and releases a signed and verified malware to deployed downstream end points.

To mitigate against these attacks, it is important to minimize the parties who can access the products, tools and other equipment used in all phases of the product life cycle to protect from insider attacks. It is important to protect system integrity by evaluating all possible attack surfaces throughout the product life cycle and known security vulnerabilities, including software update mechanisms.

Best practices include running integrity verification checks prior to connecting the system to the network, minimizing the tampered system's ability to infect other systems on the shared network and automating checks on a periodic basis to ensure continued security.

10.5.9 Product data store tampering

Product data store is a digital representation of the current device configurations. This can include components originally installed as well as new components inserted to replace existing components that are not functioning properly. An attacker can modify this file to remove, insert or change a record of the actual components currently in the product, thereby making the use of this data in quality or security excursion investigation ineffective. This can be done more easily by a support staff within the organization or contracted to provide product support, such as an outsourced IT service provider. It can be done after the fact by any third party with access to the system housing the product data store.

To mitigate tampering, one can employ encryption at rest, thus adding a layer of security within the data file itself. Additionally, a hash of the file or a copy of the file can be stored outside the storage solution with a verification step against that copy or hash every time the build data are accessed, with an alert occurring if the hash or file do not match. Additionally, occasional audits of devices serviced can be compared to the device data store file. Audits can be in the form of a physical check of the product or running utilities that interrogate what is on the product and compare the product data store file.

11 Phase 6: Retirement

11.1 General

Retirement is the final phase of the SCLC. In this phase the products are cleaned, which means all data are erased to protect the stored information and properly transferred or recycled. The objectives of retirement are not to reclaim the scrap value of the product, but rather extend the life of the product in a secondary market.

11.2 Summary of retirement threats and controls

11.2.1 Inaccurate hardware return

This issue applies to business owned products, not consumer products. It involves cases where the system itself has been changed, either components were removed or replaced with lesser quality components. This is important to identify for multiple reasons. First, because the product value is impacted and can be unacceptable for reusing in a second life. Second, because the component removed can contain valuable data assets belonging to the organization.

To mitigate against inaccurate hardware return, the IT organization is expected to maintain a current manifest on what is contained within the product. A utility can be run against the product and compared to the IT manifest to identify discrepancies and follow up with the user of that product.

11.2.2 Incomplete data removal

As ICT products become increasingly complex, it becomes increasingly difficult for IT shops to know where all the data including keys resides on the product. It is no longer sufficient to wipe the primary storage device of its data and assume all information has been removed. This issue is more relevant than ever as public cloud providers reuse servers across customers and various levels of data classifications.

To mitigate this, designers are expected to design for clean erase or reset to new states of their components flash memory. These utilities are made easily available to authorized owners of those products, so they can reset and reuse the product. These utilities are expected to be able to attest to the state of the product to the owner so that the product can be reused in other ways. Not only does this help increase the value of the product at the end of its first life, but it makes a more positive environmental impact as functioning products at the end of their depreciated life can be used again until product failure, or until they are unable to support a business need.

Annex A
(informative)

Product security threat mapping to SCLC phases

Table A.1 — Product security threat mapping to life cycle phases

Attack subsurface	Threat	Short description	Threat target and technique examples	Threat mapping to SCLC phases					
				Concept	Development	Source/ manufacturing	Transport	Utilization/ support	Retirement
Hardware	Workflow toolchain tampering	Alteration of production and support tool sets and/or insertion of malicious devices, mechanisms, configurations, software or other objects into the product toolchain with the intent to disrupt, damage, corrupt or otherwise cause harm to product life cycle workflows, equipment and process outputs.	Design, simulation, fab, ICs (lithography), ERP, supply chain, inventory	X	X	X		X	X
	Semiconductor IP (SIP) tampering	Insertion of malicious semiconductor intellectual property (SIP) into the product design with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into the product or a product component.	IP Blocks, HDL, VHDL, Crypto		X				
	Semiconductor fabrication tampering	Alteration of semiconductor fabrication resources and/or production processes at the chip-level with the intent to insert unauthorized, flawed, disabled or otherwise untrustworthy functionality into a product or product component.	Doping, masks, place and route, PUF logic			X			
	Unpackaged semiconductor tampering	Insertion at the discrete part level of malicious unpackaged semiconductor components and/or packaging technology into the product implementation with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into the product or a product component.	Chiplets, high bandwidth memory, interposers		X	X	X		
	Packaged semiconductor tampering	Insertion at the subassembly level of malicious packaged semiconductor components into the product implementation with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into the product or a subassembly thereof.	ASICs, FPGAs, DSPs, SoCs			X	X		
	Subassembly tampering	Insertion at the systems level of malicious subassemblies integrated into the product implementation with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into the product.	System boards, Host bus adapters, out-of-band management controllers, taps			X	X		
	End product tampering	Alteration or substitution of a finished product with the intent to introduce malicious behaviour into a larger system context, alter value distribution or realize other illegitimate intent.	Substitutions, counterfeits, swaps, insertions, alterations			X	X	X	X
	Tagalongs	Attachment of a malicious standalone system or device to a finished product with the intent to provide clandestine monitoring of, and/or interference with, a product's intended function.	Trackers, sensors, airgap bridges				X	X	X

Table A.1 *(continued)*

Attack subsurface	Threat	Short description	Threat target and technique examples	Threat mapping to SCLC phases					
				Concept	Development	Source/ manufacturing	Transport	Utilization/ support	Retirement
	Theft and loss	The absence, removal or destruction of a tangible asset, or the unauthorized acquisition of trade secrets due to actions taken by a malicious actor, or by environmental hazards [a] encountered within the product life cycle.	Interdiction, shipment hijacking, reverse engineering	X	X	X	X	X	X
Software	Workflow toolchain tampering	Alteration of software tool executable images, supporting code or configuration parameters, and/or insertion of malicious code with the intent to disrupt, damage, corrupt or otherwise cause life cycle harm to software system elements of a product.	Compilers, interpreters, development source libraries, dev pipelines	X	X	X	X	X	X
	uCode tampering (see F.1)	Alteration of microcode within, or the insertion of malicious microcode into, a semiconductor product dependent on low-level code during the development, production, utilization or support life cycle stages with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into a product, or a product component.	Processor, data plane, control plane, crypto		X	X	X	X	
	Firmware tampering (see F.2)	Alteration of firmware within, or the insertion of malicious firmware into, a subsystem of an information technology product during the development, production, utilization or support life cycle stages with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into a product, or a product component.	Pre-execution takeover, controls bypass/disable		X	X	X	X	
	System SW tampering (see F.3)	Alteration of system-level software within, or the insertion of malicious system-level software into, an information technology product during the development, production, utilization or support life cycle stages with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into a product, a product component, or a larger system-of-systems.	Operating Systems, virtual machine monitors (VMM) / hypervisors, management subsystems, device drivers		X	X	X	X	
	Application SW tampering	Alteration of application software within, or the insertion of malicious application code into, an information technology product during the production, utilization or support life cycle stages with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into a product, or a larger system-of-systems (e.g. network).	Trojans, remote access trojan (RAT), root kits, trackers, spyware			X	X	X	

Table A.1 *(continued)*

Attack subsurface	Threat	Short description	Threat target and technique examples	Threat mapping to SCLC phases					
				Concept	Development	Source/ manufacturing	Transport	Utilization/ support	Retirement
	Misconfigurations	Intentional or unintentional introduction of errors or flaws into the system settings or parameters governing the intended function and/or performance of a product.	Platform, system, network, storage, apps			X	X	X	X
	Theft and loss	The absence, removal or destruction of a software asset, or the unauthorized acquisition of trade secrets due to actions taken by a malicious actor, or by environmental hazards [a] encountered within the product life cycle.	DSL compromise, exfiltration, reverse engineering	X	X	X	X	X	X
Data	Data management toolchain tampering	Alteration of software executable images, supporting code or configuration settings or parameters, and/or insertion of malicious code with the intent to disrupt, damage, corrupt or otherwise cause harm to data storage system elements of a product, or the data storage systems supporting the product life cycle.	Revision control system (RCS), configuration management systems (CMS), configuration management database (CMDB)	X	X	X	X	X	X
	Unauthorized operations	The unauthorized invocation of data operations for creating, reading, updating/modifying, or deleting data managed or consumed by a product, or related to supporting the product's life cycle.	Create, read, update, delete (CRUD)	X	X	X	X	X	X
	Integrity faults	The unintentional modification or destruction of data due to technical or operational errors, faults or failures occurring within a product, or related to supporting the product's life cycle.	Media failures, transmission faults, uncorrectable errors	X	X	X	X	X	X
	Theft and loss	The absence, removal or destruction of a data asset, or unauthorized data access due to actions taken by a malicious actor, or by environmental hazards [a] encountered within the product life cycle.	Exfiltration, reverse engineering	X	X	X	X	X	X

[a] In this context, the term "hazards" and "attacks" are described as follows:

Hazards: naturally occurring dangers or man-made sources or conditions creating the potential to unintentionally inflict harm or induce an adverse impact on an ICT asset within the life cycle environment.

Attacks: conscious, motivated and focused efforts on the part of an intelligent actor having malice aforethought to intentionally inflict harm or induce an adverse impact on an ICT asset within the life cycle environment.

Annex B
(informative)

Typical threats for hardware

Tables B.1 to B.9 show a number of typical threats for hardware, which are included in Table A.1.

Table B.1 — Workflow toolchain tampering

What (threat)	Alteration of production and support tool sets and/or insertion of malicious devices, mechanisms, configurations, software or other objects into the product toolchain with the intent to disrupt, damage, corrupt or otherwise cause “out of spec.” which deliver unexpected results condition to product life cycle workflows, equipment and process outputs.		
Who (actors)	**Likely actors**	**Needed facilitators**	**Potential targets**
	— Intellectual property thieves — Industrial competitors — Intelligence services	— Internal equipment support personnel — Toolchain providers — Toolchain support personnel — Third party software providers	— Developer — Manufacturer — OEM — End user
When (life cycle stage)	— Concept — Development — Production — Utilization — Support — Retirement		
Where	— Design and development facilities — Production facility/factories — Warehouses — Shipping vehicles while in transport — Deployment staging environments — Operational environments — Refurbish, recycle, return centres		
Why	To disrupt, damage, corrupt or otherwise cause “out of spec.” condition to product life cycle workflows, equipment and process outputs.		
How: tactics, techniques and procedures (TTPs)	— Attaining remote access to the toolchain through the exploit of vulnerable network services operating on the edge of a trust boundary. — Attaining physical or logical access to the target toolchain by social engineering (e.g. intimidating insiders). — Introducing malicious hardware or code into the toolchain through the support supply-chain.		

Table B.1 *(continued)*

Safeguards and countermeasures	— Establishing physical controls (e.g. gates, guards, guns, dogs). — Establishing firewalls to limit remote access for each employee. — Establishing proper review process and mechanisms. — Ensure persons having privileged access to tool-chain components are sufficiently trained to recognize social engineering attempts to gain access. — Develop and operate an effective supply-chain risk management (SCRM) programme to evaluate suppliers and service providers and their sources.

Table B.2 — Semiconductor IP (SIP) tampering

<table>
<tr><td rowspan="2"><b>Who (actors)</b></td><td colspan="3"></td></tr>
</table>

<table>
<tr><td><b>What (threat)</b></td><td colspan="3">Insertion of malicious semiconductor intellectual property (SIP) into the product design with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into the product or a product component.</td></tr>
<tr><td rowspan="2"><b>Who (actors)</b></td><td><b>Likely actors</b></td><td><b>Needed facilitators</b></td><td><b>Potential targets</b></td></tr>
<tr><td>— Industrial competitors/ rivals
— Nation-state intelligence services</td><td>— Systems architects
— Logic block designers
— Simulation and test specialists
— Process (fab) engineers</td><td>— Developer
— Manufacturer
— OEM
— End user</td></tr>
<tr><td><b>When (life cycle stage)</b></td><td colspan="3">— Development stage
— Production</td></tr>
<tr><td><b>Where</b></td><td colspan="3">— Design and development facilities
— Production facility/factories</td></tr>
<tr><td><b>Why</b></td><td colspan="3">To disrupt, damage, corrupt or otherwise cause “out of spec.” condition to product life cycle workflows, equipment and process outputs.</td></tr>
<tr><td><b>How: tactics, techniques and procedures (TTPs)</b></td><td colspan="3">— Recruitment of insiders through social engineering (e.g. personal gain, intimidation/ threat)
— Interdiction of digital transfers between SIP supplier and consumer</td></tr>
<tr><td><b>Safeguards and countermeasures</b></td><td colspan="3">— Setting firewalls to limit access for each employee
— Setting proper peer review process and mechanisms [i.e. Secure development life cycle (SDL)]
— Validation testing plan to identify functionality or altered functionality
— SIP digital fingerprinting with secure out-of-band fingerprint dissemination
— SIP block-chaining</td></tr>
</table>

Table B.3 — Semiconductor fabrication tampering

What (threat)	Alteration of semiconductor fabrication resources and/or production processes at the chip-level with the intent to insert unauthorized, flawed, disabled or otherwise untrustworthy functionality into a product or product component.		
Who (actors)	**Likely actors**	**Needed facilitators**	**Potential targets**
	— Industrial competitor	— Contract foundries	— Fabless chip designers
	— Nation-state intelligence service	— Mask vendors	— Merchant silicon vendors
		— Toolchain providers	— Secure products OEMs
		— Toolchain support personnel	— Defence supply-chain vendors
When (life cycle stage)	Production		
Where	Production facilities/fabs		
Why	To degrade reliability, gain privileged instruction access, expose secrets, create side-channel, and/or weaken encryption or degrade other functionality		
How: tactics, techniques and procedures (TTPs)	— Mask tampering — Place and route tampering — Sub-gate dopant tampering		
Safeguards and countermeasures	— Tamper detection circuitry — Optical inspection against baseline (“golden chip”) — Side-channel variance analysis — Tamper focused functional testing		

Table B.4 — Unpackaged semiconductor tampering

What (threat)	Insertion at the discrete part level of malicious unpackaged semiconductor components and/or packaging technology into the product implementation with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into the product or a product component.		
Who (actors)	**Likely actors**	**Needed facilitators**	**Potential targets**
	— Industrial competitor/rival — Nation-state intelligence service — Military adversary	— Discrete device (chiplet) suppliers — Package technology suppliers — Internal equipment support personnel — Toolchain providers — Toolchain support personnel — Outsourced semiconductor assembly and test (OSAT) vendors	— Developer — Manufacturer — OEM — End user
When (life cycle stage)	Production		
Where	Production facility/factories		
Why	— To disrupt, damage, corrupt or otherwise cause “out of spec.” condition to product life cycle workflows, equipment and process outputs. — To introduce covert functionality.		
How: tactics, techniques and procedures (TTPs)	— Eliciting cooperation of packaging vendor (e.g. outsourced semiconductor assembly and test) through social, financial, political or other forms of pressure and enticements. — Introducing malicious unpackaged components through legitimate supply-chain distribution channels, or through component interdiction.		
Safeguards and countermeasures	— Implementing detection methods at the chip level to detect unauthorized or altered components prior to packaging [e.g. physically unclonable functions (PUFs)], setting proper review process and mechanisms to evaluate threat exposure introduced by outsourcing assembly, test and packaging of chip-level devices (e.g. chiplets) into multi-chip packaged products (see ISO/IEC 20897-1). — Developing technical methods to assess and detect out-of-spec package substrates and assemblies.		

Table B.5 — Packaged semiconductor tampering

What (threat)	Insertion at the subassembly level of malicious packaged semiconductor components into the product implementation with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into the product or a subassembly thereof.		
Who (actors)	**Likely actors**	**Needed facilitators**	**Potential targets**
	— Industrial competitor/ rival — Nation-state intelligence service — Military adversary — Counterfeit suppliers	— Discrete component suppliers — Internal equipment support personnel — Toolchain providers — Toolchain support personnel	— Developer — Manufacturer — OEM — End user
When (life cycle stage)	Production		
Where	Production facility/factories		
Why	— To disrupt, damage, corrupt or otherwise cause “out of spec.” condition to product life cycle workflows, equipment and process outputs. — To introduce covert functionality. — To distribute counterfeit product.		
How: tactics, techniques and procedures (TTPs)	— Eliciting cooperation of system assembly and test vendors (e.g. contract manufacturing vendors) through social, financial, political or other forms of pressure and enticements. — Introducing malicious components through legitimate supply-chain distribution channels, or through component interdiction. — Redirection to counterfeit supply-chain.		
Safeguards and countermeasures	— Setting proper review process and mechanisms to evaluate threat exposure introduced by outsourcing subassembly manufacturing (e.g. contract manufacturing). — Developing technical methods to assess and detect out-of-spec discrete devices. — Developing and implementing supply-chain risk management (SCRM) program.		

Table B.6 — Subassembly tampering

What (threat)	Insertion at the systems level of malicious subassemblies integrated into the product implementation with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into the product.		
Who (actors)	**Likely actors**	**Needed facilitators**	**Potential targets**
	— Industrial competitor/rival — Nation-state intelligence service — Military adversary — Counterfeit suppliers	— Subassembly suppliers — Internal equipment support personnel — Toolchain providers — Toolchain support personnel — Third party software providers	— Developer — Manufacturer — OEM — End user
When (life cycle stage)	Production		
Where	Production facility/factories		
Why	— To disrupt, damage, corrupt or otherwise cause “out of spec.” condition to product life cycle workflows, equipment and process outputs. — To introduce covert functionality. — To distribute counterfeit product. — To dump used/refurb product.		
How: tactics, techniques and procedures (TTPs)	— Obtaining access to the production facility by social engineering (e.g. intimidation insiders, paid tamper by insider). — Eliciting cooperation of system assembly and test vendors (e.g. contract manufacturing vendors) through social, financial, political or other forms of pressure and enticements. — Introducing malicious, counterfeit or degraded components through legitimate supply-chain distribution channels, or through component interdiction.		
Safeguards and countermeasures	— Setting proper review process and mechanisms to evaluate threat exposure introduced by outsourcing product manufacturing (e.g. contract manufacturing). — Developing technical methods to assess and detect out-of-spec components. — Developing and implementing supply-chain risk management (SCRM) program.		

Table B.7 — End product tampering

What (threat)	Alteration or substitution of a finished product with the intent to introduce malicious behaviour into a larger system context, alter value distribution or realize other illegitimate intent. Once a product has completed its manufacturing process, some attackers can attempt to alter the physical aspects of the product in order to introduce a backdoor or vulnerability that can be exploited remotely. Numerous companies exist today that provide the ability to perform a teardown of physical devices. A teardown can provide an attacker with information to change the product functionality or to insert trojan functionality. Tampering of a product can also occur at the system level rather than individual components. This can include insertion or replacement of legitimate components within a system. Detection of such compromises can range from a simple visual inspection to complex power analysis to requiring a full teardown of the device.		
Who (actors)	**Likely actors**	**Needed facilitators**	**Potential targets**
	— Anyone with physical access to the product and the technological capability to implement an alteration or substitution.	— Complicit employee — Shipping/distribution contractor	— Manufacturer — OEM — End user
When (life cycle stage)	— Production — Support		
Where	— Assembly — Storage — Transit		
Why	To enable high-privilege access to product over remote connection, enable out-of-band exfiltration of product information, induce malfunction later, extract product or user data via covert channel, and to take over control of the product remotely.		
How: tactics, techniques and procedures (TTPs)	Injecting trojan or backdoor circuitry into the product, or adding extra component to existing product to perform required task (e.g. hardware or software implant).		
Safeguards and countermeasures	— Visual inspection — Component imaging — Verification of operations within predefined thresholds, such as voltage — Software image cryptographic signing		

Table B.8 — Tagalongs

What (threat)	Attachment of a malicious standalone system or device to a finished product with the intent to provide clandestine monitoring of, and/or interference with, a product's intended function, or to surveil or collect information about the target (owner or end-user consumer) of the product. See details in Annex E.		
Who (actors)	**Likely actors**	**Needed facilitators**	**Potential targets**
	— Law enforcement — Intelligence services — Military units — Political activists — Private investigators	— End user — Commercial freight and delivery services — Private couriers — Distributors/wholesalers — Retailers — Service providers — Other insiders	— Developer — Manufacturer — OEM — End user consumers — Political campaigns — Criminal organizations — Military units — Government officials
When (life cycle stage)	— Utilization — Support — Retirement		
Where	— Product transportation – tagalongs can be attached or affixed to a product destined for the target consumer during transit from a supplier. The tagalong can be introduced at the point of origin/departure from a supplier, or through interdiction of the product while in route to the target. — End user consumer site — Product wholesale and retail distribution points – tagalongs can be introduced by middle market actors		
Why	— To introduce malicious behaviour into a larger system context, or other malevolent intent. — To acquire geospatial data to determine target’s location and travel pattern/profiling. — To bridge an airgap. — To collect audio, visual and communication signals/traffic/data. — To establish remote and/or persistent presence.		
How: tactics, techniques and procedures (TTPs)	Adding malicious device or function to the target product by adversarial actors or unintentional operation by end user.		
Safeguards and countermeasures	— Anonymous supply-chain product acquisition — RF scanning/monitoring — Visual inspection — RF shielded operational environment — Extreme temperature exposure — Salt water immersion		

Table B.9 — Theft and loss

What (threat)	The absence, removal or destruction of a tangible asset, or the unauthorized acquisition of intellectual property (IP) due to actions taken by a malicious actor, or by environmental hazards encountered within the product life cycle. Theft or loss of a product can happen at any stage across the product life cycle. During development, IP belonging to an organization can be seized on corporate networks or willingly provided by a malicious insider. Once the final product is manufactured, its theft or loss can have a variety of consequences based on a number of different factors. For example, some products, such as integrated circuits, often require provisioning to prepare the part for full release. Until that process is completed, the security of the part is often considered incomplete. The theft or loss of a part before the provisioning process is completed can result is easier extraction of critical assets or IP from within the part. This ensures sure parts of prime interest to highly sophisticated and/or well-funded adversaries.		
Who (actors)	**Likely actors**	**Needed facilitators**	**Potential targets**
	— Anyone with physical access to the asset or the element on which the asset is located. — Anyone with access to network or storage element where IP is stored.	— Complicit employee — Shipping contractor — Warehouse employee	— Developer — Manufacturer — OEM — End User
When (life cycle stage)	— Development — Production — Utilization — Support — Retirement		
Where	— Product storage — Product transportation — Corporate networks — Production facilities		
Why	— To compromise the functionality of the final product — To obtain early or cheap access to advanced technology — To gain technological advantage by stealing superior or unsupported IP — To damage the brand recognition of a manufacturer or supplier — To build functional or execution capabilities at a fraction of the cost and/or time		
How: tactics, techniques and procedures (TTPs)	Gaining access to asset of IP via social engineering of employee or contractor.		
Safeguards and countermeasures	— Asset tracking and accountability — Visual inspection — Setting firewalls and user permissions to restrict access to sensitive IP — Audit access logs to trade sensitive IP		

Annex C
(informative)

Typical threats for software

Tables C.1 to C.6 shows a number of typical threats for software, which are included in Table A.1.

Table C.1 — Workflow toolchain tampering

What (threat)	Alteration of software tool executable images, supporting code or configuration parameters, and/or insertion of malicious code with the intent to disrupt, damage, corrupt or otherwise cause life cycle harm to software system elements of a product.		
Who (actors)	**Likely actor**	**Needed facilitators**	**Potential target**
	— Industrial competitors — Nation-state intelligence services	— Internal equipment support personnel — Toolchain providers — Toolchain support personnel — Third party software providers	— Developer — Manufacturer — OEM — End user
When (life cycle stage)	— Concept — Development — Production — Utilization — Support — Retirement		
Where	— Design and development facilities — Production facility/factories — Warehouses — Shipping vehicles while in transport — Deployment staging environments — Operational environments — Refurbish, recycle, return centres		
Why	To disrupt, damage, corrupt or otherwise cause “out of spec.” condition to product life cycle workflows, equipment and process outputs.		
How: tactics, techniques and procedures (TTPs)	Gaining physical or logical access to the target toolchain by social engineering (e.g. intimidation insiders)		
Safeguards and countermeasures	— Setting firewalls to limit access for each employee — Setting proper review process and mechanisms		

Table C.2 — Microcode Tampering

What (threat)	Alteration of microcode within, or the insertion of malicious microcode into, a semiconductor product dependent on low-level code during the development, production, utilization or support life cycle stages with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into a product or a product component.		
Who (actors)	**Likely actors**	**Needed facilitators**	**Potential targets**
	— Industrial adversaries — Law enforcement — Nation-state intelligence services	— Cooperative/compelled/ coerced OEMs — Reverse engineering services providers — Transportation services providers — Operational services providers — Maintenance/support services providers	— Market competitors/rivals — Investigation targets — Geo-political nation-state rivals — Political leaders — Political dissidents — Coercion targets — Private individuals
When (life cycle stage)	— Development — Production — Utilization — Support		
Where	— Design and development facilities — Production facility/factories — Warehouses — Shipping vehicles while in transport — Deployment staging environments — Operational environments		
Why	To weaken encryption, intercept/tap live I/O streams (video, voice, data), establish/maintain persistence, remote access, modify micro-architectural behaviour, induce side-channel.		
How: tactics, techniques and procedures (TTPs)	— Initial OEM deployments. — Product interdiction and tampering. — Product maintenance/updates.		
Safeguards and countermeasures	— Enforced code transparency policies — Device logic transparency — Static code analysis — Provenance verification — Cryptographic hashing — Supplier and service provider vetting — External product operational observability		

Table C.3 — Firmware Tampering

What (threat)	Alteration of firmware within, or the insertion of malicious firmware into, a subsystem of an information technology product during the development, production, utilization or support life cycle stages with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into a product or a product component.		
Who (actors)	**Likely actors**	**Needed facilitators**	**Potential targets**
	— Industrial adversaries — Law enforcement — Nation-state intelligence services	— Cooperative/compelled /coerced OEMs — Reverse engineering services providers — Transportation services providers — Operational services providers — Maintenance/support services providers	— Market competitors/rivals — Investigation targets — Geo-political nation-state rivals — Political leaders — Political dissidents — Coercion targets — Private individuals
When (life cycle stage)	— Development — Production — Utilization — Support		
Where	— Design and development facilities — Production facility/factories — Warehouses — Shipping vehicles while in transport — Deployment staging environments — Operational environments		
Why	— Intercept/tap live I/O streams (video, voice, data) — To establish/maintain persistence — Remote access — Denial of Service — Induce side-channel		
How: tactics, techniques and procedures (TTPs)	— Initial OEM deployments — Product interdiction — Product maintenance/updates — Operational compromise		
Safeguards and countermeasures	— System-level hardware Root of Trust (RoT) — Device-level semiconductor RoT		

Table C.4 — System Software Tampering

What (threat)	Alteration of system-level software within, or the insertion of malicious system-level software into, an information technology product during the development, production, utilization or support life cycle stages with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into a product, a product component or a larger system-of-systems.		
Who (actors)	**Likely actor**	**Needed facilitators**	**Potential target**
	— Disgruntled employee — Compromised insider	— Individuals or organizations with access to the software while under development or its associated network — Individuals or organizations with access to the end product — Individuals or organizations with access to the product distribution system — Individuals involved in loading system software onto ICT product	— Developer — Manufacturer — OEM — End user
When (life cycle stage)	— Development — Production — Utilization — Support		
Where	— Design and development facilities — Distribution staging environment — Operational environment		
Why	— Compromise system functionality — Remote access — To establish/maintain persistence — To exfiltrate information assets		
How: tactics, techniques and procedures (TTPs)	— Injecting into source code by disgruntled employee or malicious insider. — Compromising deployment server. — Compromising update server. — Modifying unsigned code. — Extracting code signing key from manufacturer.		
Safeguards and countermeasures	— Maintaining and auditing source code repository logs. — Keeping signing keys in high-security hardware with minimum employee access. — Signing all code using NIST-compliant algorithms and keys.		

Table C.5 — Application Software Tampering

What (threat)	Alteration of application software within, or the insertion of malicious application code into, an information technology product during the production, utilization or support life cycle stages with the intent to introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into a product, or a larger system-of-systems (e.g. network).		
Who (actors)	**Likely actor**	**Needed facilitators**	**Potential target**
	— Individuals or organizations with access to the software while under development or its associated network — Individuals or organizations with access to the end product — Individuals or organizations with access to the product distribution system	— Internal equipment support personnel — Toolchain providers — Toolchain support — Third party software providers	— Developer — Manufacturer — OEM — End user
When (life cycle stage)	— Production — Utilization — Support		
Where	— Production facility — End user site		
Why	To introduce unauthorized, flawed, disabled or otherwise untrustworthy functionality into a product, or a larger system-of-systems (e.g. network).		
How: tactics, techniques and procedures (TTPs)	Insertion of malicious application code into an information technology product during the production, utilization or support life cycle stages.		
Safeguards and countermeasures	— Maintaining and auditing source code repository logs. — Keeping signing keys in high-security hardware with minimum employee access. — Signing all code using NIST-compliant algorithms and keys.		

Table C.6 — Theft and Loss

What (threat)	The absence, removal or destruction of a software asset, or the unauthorized acquisition of trade secrets due to actions taken by a malicious actor or by environmental hazards encountered within the product life cycle. Theft or loss of software/firmware typically revolves around an attacker's ability to access the original source code used in the creation of the end product. Access to the end binary can be of concern if there is a need to protect the confidentiality of the IP contained therein. Additionally, loss can become a critical concern if the original IP is lost as well as any backups that existed. Such an occurrence has resulted in more than one organization going out of business. The ability to attack an organization in such a manner would make this of prime interest to disgruntled employees, competitors or well-funded adversaries seeking to eliminate capabilities.

Table C.6 *(continued)*

Who (actors)	**Likely actor**	**Needed facilitators**	**Potential victims**
	— Disgruntled employee — Compromised employee — Criminal individuals or organizations motivated by financial gain through sale of stolen intellectual property or data. — Industrial espionage agent working on behalf of an industrial competitor or nation-state backed entity.	— Warehouse employee — Logistics employee — Factory employee — Authorized service provider — Individual that breaches physical security and gains unauthorized access to the product	— Developer — End user
When (life cycle stage)	— Development — Production — Utilization — Support		
Where	— Production facility — End user site — In transport		
Why	— To enable usage of software — To extract IP/trade secrets — To create their own product — To destroy competitor's advantage — To reduce financial cost from legitimate usage — To enhance own capabilities by integrating superior IP — Financial income through creation of competitive product		
How: tactics, techniques and procedures (TTPs)	— Reverse engineer solution — Extracting source code — Bypass licensing — Creating fake licenses		
Safeguards and countermeasures	— Maintaining and auditing access logs for source code repositories. — Utilizing obfuscation techniques to strengthen resistance to reverse engineering. — Minimizing employee access to trade secrets/IP. — Utilizing trusted execution environments for validation of licenses and usage/storage of key material.		