|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | |  | | |  | | |  |
| 文件名稱 Document Name： | | | | 版本Version： | | |  | | |  |
|  | 網路安全系統開發程序書  Cybersecurity System Development Procedure |  | | | | |  | | |  |
| 文件編號 Document No.： | | | | 合計頁次：共  Total Page(s): | | |  | | 頁Page(s) | |
|  | |
|  |  |  | |  | |
|  | | | | (不含本封面及文件修訂履歷頁Not include this  cover and document revision history) | | | | | | |
| (制)修訂單位 (Issued) Revised Department： | | | |  | | | |  | | |
| (制)修訂頁次 (Issued) Revised Page(s)： | | |  | | (日期 Date： / / ) | | | | | |
|  | | |  | |  | | | | | |
| □ 新訂 New Released  □ 修訂 - 修訂內容參閱「文件修訂履歷」  Revision – refer to the “Revision History of Document”.    □ 發行廠別 Published Plant：□ LAK □ LGA □ LAMX □ LAC □ GZP □ DYP  　　　　　　　　　　　　　 □ \_\_\_\_\_\_\_\_ | | | | | | | | | | |
| 審核意見或建議Comment or Suggestion:  請參考PLM系統簽核記錄 Please refer to the approval history on PLM system. | | | | | | | | | | |
| 核 准 Approval | | 審 查 Confirmation | | | | 申 請 Applicant | | | | |
| PLM 簽核  Approved by PLM | | PLM 簽核  Approved by PLM | | | |  | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 版本  Version | 修訂日期Revised Date | 修訂人員Issuer | 修訂內容  Revised Conent |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Content**

[1. Purpose 1](#_Toc115274543)

[2. Scope of application 1](#_Toc115274544)

[3. TERMS AND ABBREVIATIONS 1](#_Toc115274545)

[3.1. Terms and Definitions 1](#_Toc115274546)

[3.2. Abbreviated Terms 3](#_Toc115274547)

[4. Overview of Cybersecurity Development Phase 5](#_Toc115274548)

[4.1. Cybersecurity Development Phase Overview 5](#_Toc115274549)

[4.2. Objectives of the Cybersecurity Development Phase 6](#_Toc115274550)

[5. CS system level development phase 6](#_Toc115274551)

[5.1. CS System Level Development Phase Flow 6](#_Toc115274552)

[5.2. CS System Level Development Phase Steps and Deliverables 7](#_Toc115274553)

[5.3. Timing of CS system level development phase 10](#_Toc115274554)

[6. System level CS development plan 10](#_Toc115274555)

[7. Technical Cybersecurity Requirement Definition 10](#_Toc115274556)

[7.1. Selecting Security Controls 11](#_Toc115274557)

[7.2. Technical Cybersecurity Requirement (TCR) definition 12](#_Toc115274558)

[7.2.1. Derivation of Technical Cybersecurity Requirement (TCR) from security function components 13](#_Toc115274559)

[7.3. External CS Requirement (XCR) definition 15](#_Toc115274560)

[7.4. Creation of Technical Cybersecurity concept (TCC) 16](#_Toc115274561)

[8. System design 16](#_Toc115274562)

[8.1. CS HSI Definition 16](#_Toc115274563)

[8.2. System design 17](#_Toc115274564)

[8.2.1. CS requirements for system design 18](#_Toc115274565)

[8.2.2. CS Architecture Design Principles 18](#_Toc115274566)

[8.2.3. Security architecture design 18](#_Toc115274567)

[8.2.3.1. Secure security features 18](#_Toc115274568)

[8.2.3.2. Define Security Domain 19](#_Toc115274569)

[8.2.3.3. Domain separation mechanism 20](#_Toc115274570)

[8.2.3.4. CPU operation mode 21](#_Toc115274571)

[8.2.3.5. Point of entry for the attacker 21](#_Toc115274572)

[8.2.3.6. Log Design 22](#_Toc115274573)

[8.2.3.7. Log security measures 22](#_Toc115274574)

[8.3. System safety impact analysis 23](#_Toc115274575)

[8.4. System vulnerability analysis 24](#_Toc115274576)

[8.4.1. Identification of assets 26](#_Toc115274577)

[8.4.2. Identify threat scenarios 26](#_Toc115274578)

[8.4.3. Identify Attack path 26](#_Toc115274579)

[8.4.4. Risk analysis/Risk assessment/Risk management 27](#_Toc115274580)

[8.4.5. Anomaly response 27](#_Toc115274581)

[8.5. System CS Test Item Definition 27](#_Toc115274582)

[8.6. Vulnerability test item definition 28](#_Toc115274583)

[8.7. Verification of system CS design 32](#_Toc115274584)

[9. System-level validation 32](#_Toc115274585)

[9.1. System CS test environment construction 33](#_Toc115274586)

[9.2. System CS test conducted 34](#_Toc115274587)

[9.3. Verification of system CS test results 34](#_Toc115274588)

[10. CS validation 35](#_Toc115274589)

[10.1. CS Validation Phase Overview 35](#_Toc115274590)

[10.2. CS Validation Phase Objectives 35](#_Toc115274591)

[10.3. CS Validation Plan 35](#_Toc115274592)

[11. System Forms： 37](#_Toc115274593)

[12. References： 37](#_Toc115274594)

# Purpose

This document and other internal standards related to Cybersecurity response cover the entire area of Cybersecurity in the engineering of electrical and electronic (E/E) systems on the road. The implementation of processes corresponding to these documents ensures proper consideration of cyber security and enables the engineering of E/E systems that follow changes in technology and attack methods.

This document defines procedures for system-level development of product development.

# Scope of application

It is aimed at new development and derivative development of the system.

# TERMS AND ABBREVIATIONS

## Terms and Definitions

Table3‑1 list of Terms and Definitions

|  |  |
| --- | --- |
| Term | Definition |
| Items | A system or combination of systems that provides vehicle-level functionality. It is also commonly referred to as a unit or controller. |
| Cybersecurity Goals (CG) | A description of the requirements of the top-level cybersecurity perspective. It is common to describe assets (Information, features, etc.) in combination with security attributes (Confidentiality, integrity, availability, etc.). |
| External Cybersecurity Requirement (XCR) | A CS request that is expected to be addressed by an external element (Elements that are outside the scope of the CSMS development process for the item). They vary from requirements for external systems to requirements for manufacturing and operation. |
| Functional Cybersecurity Requirement (FCR) | Cybersecurity functional requirements derived at the item level. Assigned to any system that is a component of an item. |
| Functional Cybersecurity Concept (FCC) | Block diagram defining how Functional Cybersecurity Requirements (FCR) are assigned to system components. |
| Technical Cybersecurity Requirement (TCR) | A technical requirement that defines how functional CS requirements (FCR) are realized by system components. It is assigned to HW and SW elements which are system components. |
| Technical Cybersecurity Concept (TCC) | A block diagram defining how Technical Cybersecurity Requirement (TCR) are assigned to system components. |
| CS HSI Specification | Based on the system design specifications and TCC, the specifications show the roles and relationships between hardware and software. Used to agree on topics that span both hardware and software development. The HSI specification will continue and be detailed during the hardware and software development process. |
| Assets | An asset is something of value to a product's stakeholders. Therefore, some protection is necessary. Information, functions, etc. |
| Security property  (Security attribute) | A characteristic of an asset that represents what cyber threats the asset is protected from. Confidentiality, Integrity and Availability. |
| Confidentiality | Information (Assets) is protected so that unauthorized persons (and equipment) cannot (Read) view the information (Assets). |
| Integrity | Information (Assets) is protected so that unauthorized persons (and equipment) cannot (Write) modify the information (Assets). |
| Availability | The property is protected to ensure that authorized persons (and equipment) have (with no hindrance) access to the property. |
| Asset Intrusion Event | An asset's security attributes are compromised. It represents an undesirable event related to an asset. |
| Damage scenario | An adverse and undesirable consequence of an asset infringement event. A damage scenario describes both an asset infringement event, which is a causal event, and an Predicted Damage, which is a consequential event.  (For example, a third party may change the calibration settings of the engine ECU, causing the engine to fail to comply with emission regulations.) |
| Predicted Damage | An adverse and undesirable consequence of an asset infringement event. The Predicted Damage describes only the resulting events of the damage scenario. |
| Threat scenario | A threat scenario is a description of a potential action that would result in a damage scenario. Express in the set of (Attack methods that cause asset infringement events).  (Example: A third party changes the calibration setting of the engine ECU by rewriting the memory from the DLC by the UDS.) |
| CS claim | A statement of acceptance of risk. |
| Risk | The impact of uncertainty about cyber security in roadside vehicles, expressed in terms of attack potential and impact. |
| FCR/TCR intrusion event | State of the system against FCR and TCR. The purpose of system level CS design is to design the system so that this condition does not occur. |
| Attack Method | Any action on a component of the system that results in an FCR/TCR intrusion event. |
| Countermeasure TCR | A technical requirement that defines how the system components block an attack in order to prevent it from being established. |
| security architecture | A design policy, program structure, and system for safely implementing security functions against threats. |

## Abbreviated Terms

Table3‑2 list of abbreviated terms

|  |  |
| --- | --- |
| Term | Definition |
| CAL | Cybersecurity Assurance Level |
| CC | Common Criteria |
| CPU | Central Processing Unit |
| CS | Cybersecurity |
| DIA | Development Interface Agreement |
| DLC | Data link coupler |
| XCR | eXternal Cybersecurity Requirement |
| ECU | Electronic Control Unit |
| FT | Fault Tree |
| HCR | Hardware Cybersecurity Requirement |
| HSI | Hardware Software Interface |
| HW | Hardware |
| IPA | Information-technology Promotion Agency |
| IPsec | Security Architecture for Internet Protocol |
| ASO | Automotive Standards Organization |
| ASPAR | Automotive Software Platform and Architecture |
| ISEC | Information Technology Security Evaluation and Certification Scheme |
| OEM | Original Equipment Manufacturer |
| QM | Quality Management |
| SCR | Software Cybersecurity Requirement |
| SW | Software |
| TCC | Technical Cybersecurity Concept |
| TCR | Technical Cybersecurity Requirement |

# Overview of Cybersecurity Development Phase

## Cybersecurity Development Phase Overview

The Cyber Security Development Phase refers to cyber security activities at the system, hardware, and software level in the V-shaped model of product development.

Figure4‑1 Flow of Cybersecurity Activities for Product Development in the V-Shaped Model (Based on ISO 21434)The flow of cyber security activities for product development in the V-shaped model is shown in Fig. On the left side of the V-shaped model, the cyber security requirements are refined at each level and the architectural design is implemented. Furthermore, on the right side of the V-shaped model, integration and verification corresponding to the left side will be performed.

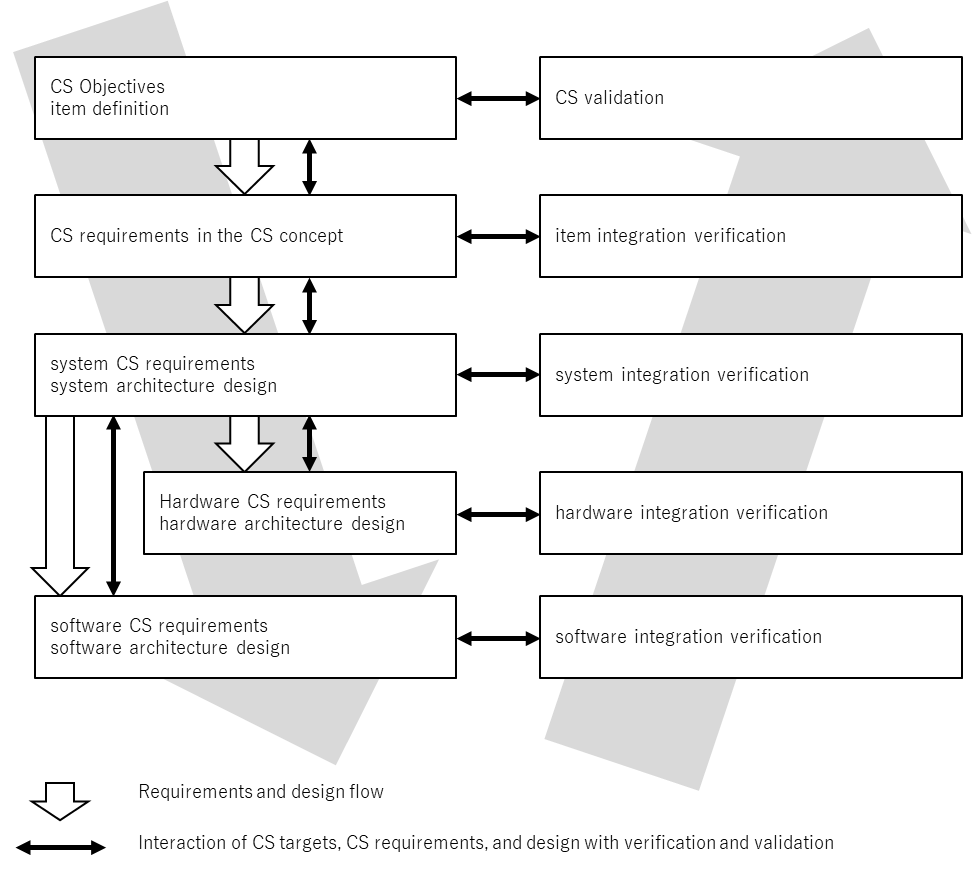


Figure4‑1 Flow of Cybersecurity Activities for Product Development in the V-Shaped Model (Based on ISO 21434)

Cyber security requirements and architectural design refinements are implemented at each level on the left side of the V-shaped model. Activities include the following.

\*Comply with high-level cybersecurity requirements

\*prevent new vulnerabilities from entering

\*Identify and manage known vulnerabilities

At each level on the right side of the V-shaped model, integration and verification are performed for each level on the corresponding left side.

\*Configure and integrate systems and components based on the correct cyber security settings

\*Validate and comply with cyber security requirements assigned to systems and components

\*Verify that identified vulnerabilities are properly managed

## Objectives of the Cybersecurity Development Phase

The objectives of the Cyber Security Development phase are:

1. Define refined cyber security requirements and architectural design
2. Verify that refined server security requirements and architectural design comply with high-level cyber security requirements
3. Identify and manage vulnerabilities in the design
4. Provide evidence that the component complies with the Cybersecurity specification and does not contain unnecessary features related to Cybersecurity

# CS system level development phase

## CS System Level Development Phase Flow

The diagram below shows the flow of implementation regarding the deliverables and security requirements related to the security requirements at each level.

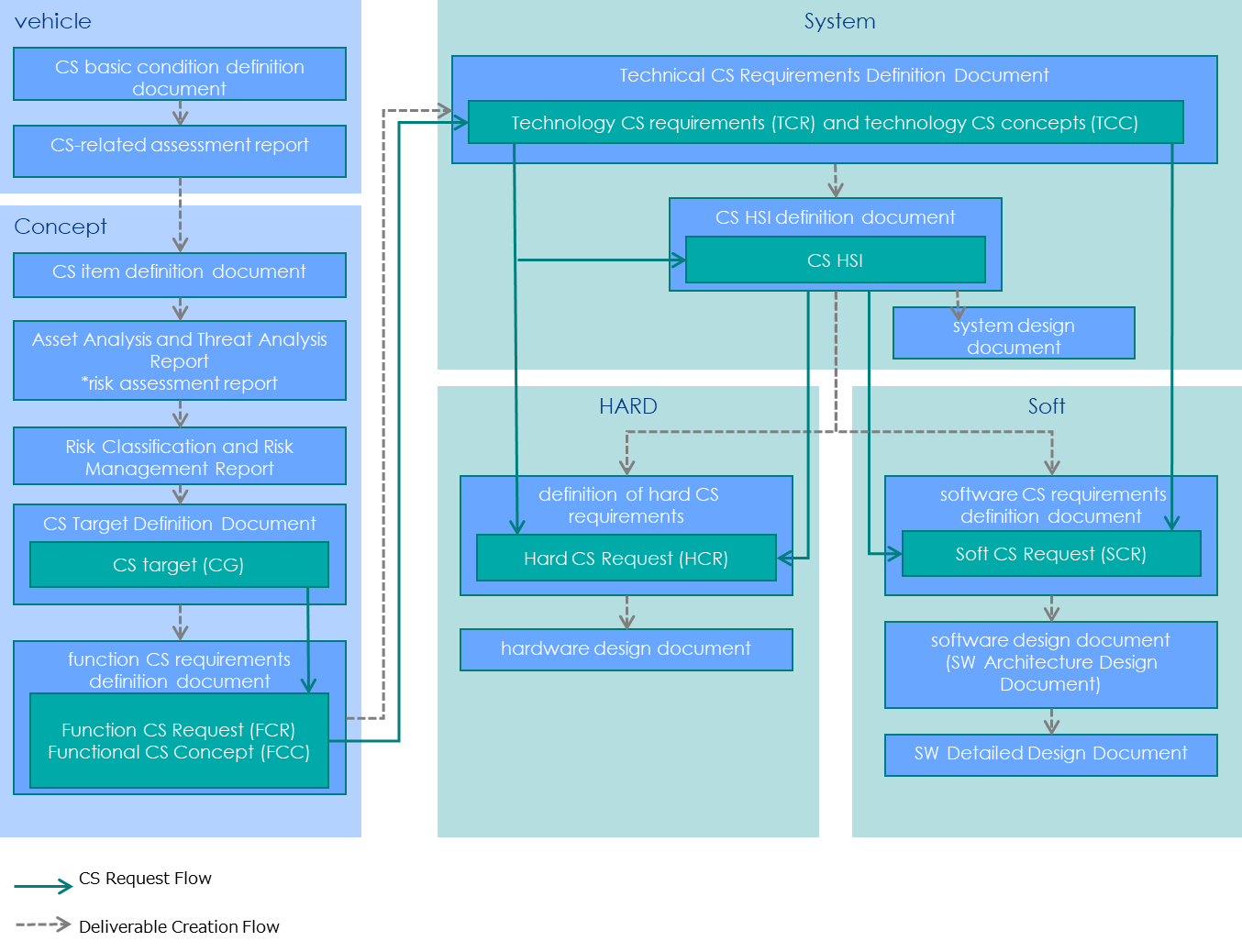


Figure5‑1 CS Request and Deliverable Creation Flow

At the system level, the following deliverables are created:

[Technical Cybersecurity Requirement (TCR), Technical CS Concepts (TCC), CS Hardware and Software Interface Specifications (HSI)]

For the Functional Cybersecurity Concept (FCC), create Technical Cybersecurity Requirement (TCR) that embody measures. In addition, by designing a system that meets Technical Cybersecurity Requirement (TCR), Technical Cybersecurity Requirement (TCR) are placed on the hardware and software, CS HSI is defined, and a technical CS concept (TCC) is created.

## CS System Level Development Phase Steps and Deliverables

The outline and flow of each step at the CS system level, and the deliverables of the steps are shown.

Table5‑1 System-level development phase steps and deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| Step | | Summary | Key Deliverables |
| system CS planning | - | Plan system-level CS activities. Plan in consideration of new, derived, and diversion. | \*System CS Plan |
| Technical Cybersecurity Requirement Definition | Selection of security functions (Controls) | Select a security function (Controls) that meets the Cybersecurity Goals (CG) or meets the functional CS requirements (FCR). | \*Security features (Controls) |
| Technical Cybersecurity Requirement (TCR) definition | A means for realizing a Functional Cybersecurity Requirement (FCR) by HW and SW is defined as a Technical Cybersecurity Requirement (TCR). | \*Technical Cybersecurity Requirement (TCR) |
| External CS Request (XCR) definition | Define CS requests addressed by external elements (Elements that are outside the scope of the CSMS development process for the item). | \*External Cybersecurity Requirement (XCR) |
| Creation of Technical Cybersecurity concept (TCC) | A functional block for realizing a technical CS request (TCR) is specified, detailed and arranged at a hardware/software level. | \*Technical Cybersecurity Concept (TCC) |
| system design | CS HSI Definition | This paper defines the cooperation specification between HW and SW related to the realization of technical CS requirement (TCR). | \*CS HSI definition document |
| system design | To design a system level that includes safety requirements and CS requirements (TCR/TCC) in addition to the requirements of original functions. | \*system design document |
| safety impact analysis | Review the safety impact of CS requirements and adjust CS requirements. | \*System Safety Impact Analysis Report |
| system vulnerability analysis | To identify a threat scenario for an asset including a protected asset of a security function and evaluate whether the threat scenario is established. | \*System Vulnerability Analysis Report |
| System CS Test Item Definition | Define test items to verify that functional CS requests (FCR)/technical CS requests (TCR) are implemented correctly. | \*System CS Test Item Definition Document |
| Vulnerability test item definition | Define test items to verify the existence of vulnerabilities that could potentially be exploited by potential threats. | \*Vulnerability test item definition document |
| Verification of system CS design | It is necessary to confirm the validity and appropriateness of the deliverables prepared at the system level, and verification is carried out for that purpose. | \*System CS Plan Verification Report  \*System CS Design Verification Report  \*CS Assessment Report |
| System CS Test | System CS test environment construction | Build a system-level test environment. | \*System CS Test Environmental Report |
| System CS test conducted | Conduct and document CS request-based functional and vulnerability tests. | \*System CS Test Results Report  \*vulnerability test results report |
| Verification of system CS test results | Deliverables created at the system level need to be validated and assessed. | \*CS Assessment Report |
| CS validation | CS Validation Plan | Prepare a test plan and prepare a plan for CS validation. | overall test plan |
| Implementation of CS validation | Conduct validation according to the CS validation test plan and record the results. | CS Validation Report |

## Timing of CS system level development phase

Technical Cybersecurity Requirement are implemented by the design department prior to PDR or received by the OEM and reviewed with the certificate department. The Functional Cybersecurity Concept (FCC) and the technical CS concept (TCC) shall be implemented and reviewed before the start of hardware and software design at the latest.

# System level CS development plan

Plan system-level CS activities.

# Technical Cybersecurity Requirement Definition

In the technical CS requirement definition, a technical CS requirement (TCR) is derived, which embodies the functional CS requirement (FCR) with respect to the Functional Cybersecurity Concept (FCC). The positioning of Technical Cybersecurity Requirement is shown.

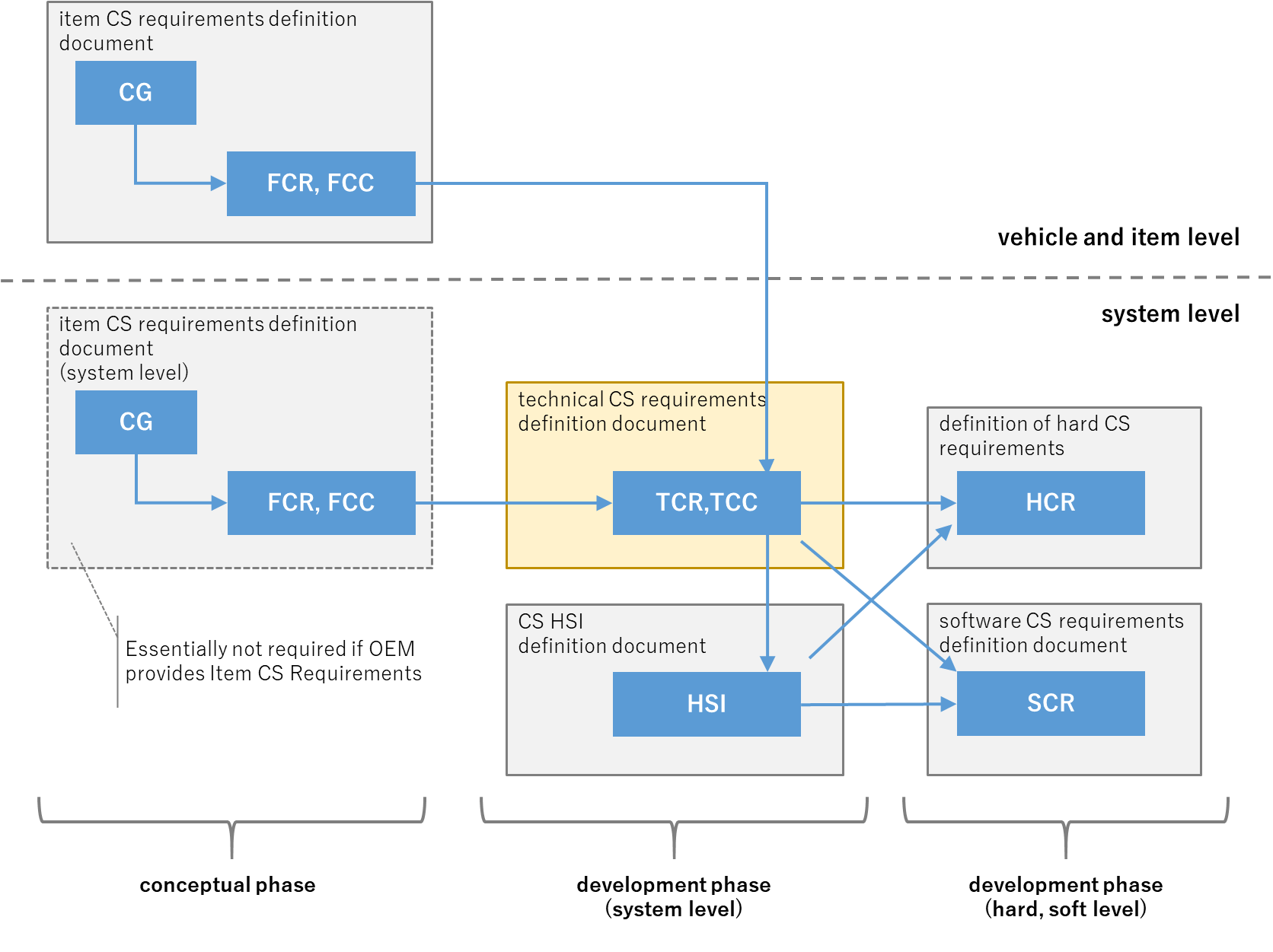


Figure7‑1 Positioning of Technical Cybersecurity Requirement

## Selecting Security Controls

If the completeness of the functional security requirement (FCR) has been verified in the concept phase, this step can be omitted.

Purpose:

To select a security function component satisfying a function security requirement (FCR).

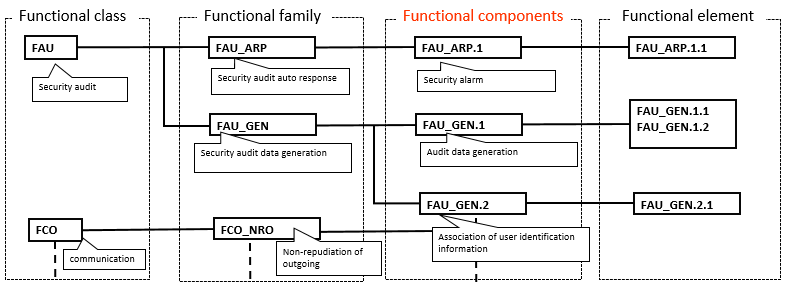
Input:

\*Item CS Requirements Statement (Functional Cybersecurity Requirement (FCR))

Work:

In order to show that the Technical Cybersecurity Requirement (TCR) are comprehensively derived from the functional CS requirements (FCR) for the security function, the security function is selected from the security function requirements collection (From Common Criteria Part 2). Select a Functional Cybersecurity Requirement (FCR) from Common Criteria Part 2.

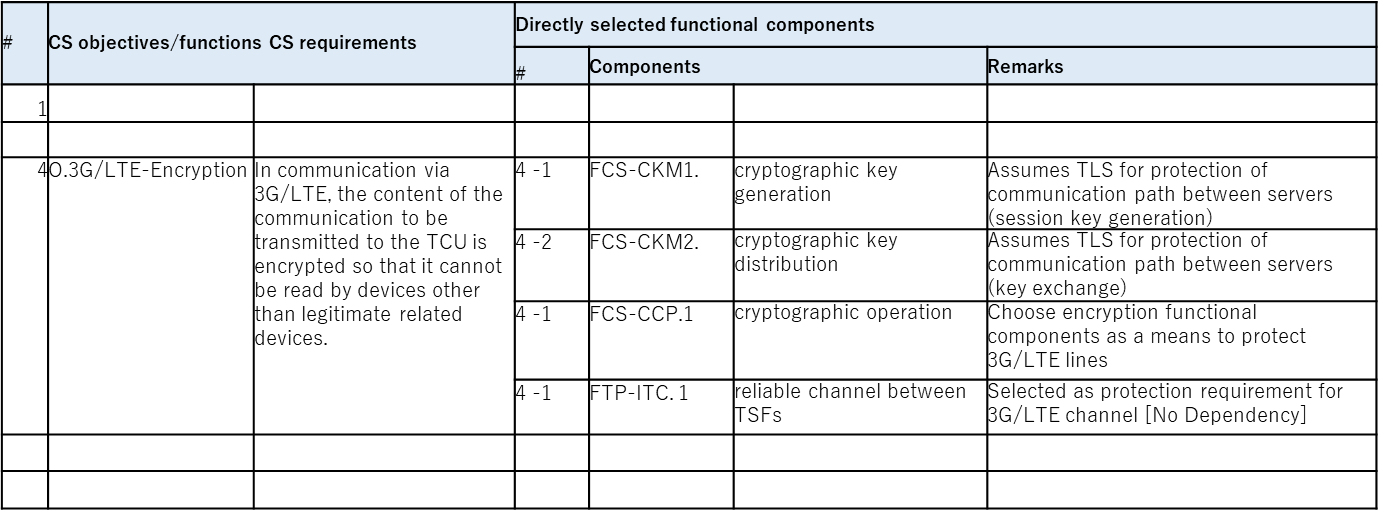
To the security function component hierarchyFigure7‑2 Hierarchical structure of security functional requirementsIt is shown in.



Security Guide

Figure7‑2 Hierarchical structure of security functional requirements

Table7‑1 Selected Security Feature Component Examples



## Technical Cybersecurity Requirement (TCR) definition

Based on the Functional Cybersecurity Requirement (FCR) and function CS concept (FCC) created in the concept phase, the CS request is progressively detailed.

Input:

\*Basic system design [additional support information]

Forms:

\* XXX

Work:

The Technical Cybersecurity Requirement (TCR) are derived, in which the functional CS requirements (FCR) are embodied to the granularity that can be implemented. When the Functional Cybersecurity Requirement (FCR) becomes the CS request for the post-development phase, the following message is displayed:7.3External CS Request (XCR) definition defines the CS request.

Examples of Technical Cybersecurity Requirement (TCR) are shown.

Table7‑2 Example of description of Technical Cybersecurity Requirement (TCR) to realize functional CS requirements (FCR)

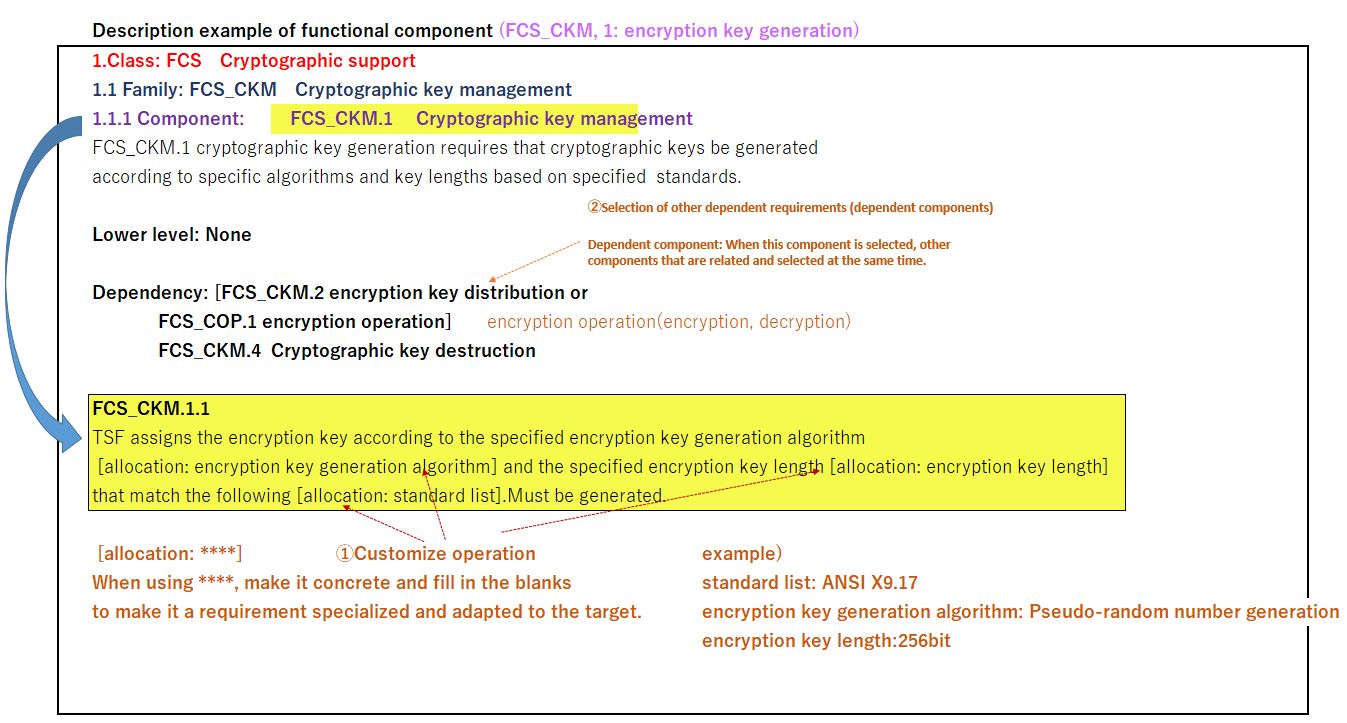
|  |  |
| --- | --- |
| Functional Cybersecurity Requirement (FCR) | Technical Cybersecurity Requirement (TCR) |
| Communication path A shall be protected by encryption and tampering detection. | \*The communication method of channel A is TLS 1.2.  \*Only Table X supports TLS cipher suites |
| The system's diagnostic and repro functions should be protected by authentication. | \*UDS security access is used as the authentication method for diagnostic and repro functions.  \*The encryption key for security access is held in the microcomputer secure element. |

### Derivation of Technical Cybersecurity Requirement (TCR) from security function components

If you have selected security feature components according to 7.1, by specifying the functional elements, the Technical Cybersecurity Requirement (TCR) can be comprehensively derived. The procedure for deriving the CS request from the security function component is described below.

* Identifying Functional Elements

Search the body of CC part 2 based on the selected security function component. Examples of functional component descriptions (Figure7‑3 Examples of functional component descriptionsThe corresponding functional element is specified by referring to (3).



Security Guide

Figure7‑3 Examples of functional component descriptions

* Create Functional Requirements

The four operations (Following repetitive, allocation, selection, and detailing operations) defined in the Common Criterion Part 1 are performed on the [The portion to be filled in (allocated portion)] of the specified function element, and the specific contents are filled in to obtain the CS request.

* 1. repetitive operation

Create multiple functional requirements from one functional element when you want to use the same functional element multiple times. Each of the plurality of elements has a part different from each other, and the different parts are performed by other operations.

* 1. allocation operation

The specification of a value or set of values that embody and are specific to a variable element of a functional element. The set of values means that a plurality of standards (Standards, etc.) can be described, for example, when Standard List is described.

* 1. selection operation

Select one or more items from the candidates specified in the variable elements of the functional element.

* 1. detailing operation

To specify the content of the original text in more detail other than assignment and selection of functional elements to variable elements.

## External CS Requirement (XCR) definition

An external CS request is a CS request that is expected to be dealt with by an external element (Elements that are outside the scope of the CSMS development process for the item), and it is a request to an external system or a request to production or operation/maintenance. This section defines the requirements for the external systems to be linked, and the requirements for manufacturing, operation/maintenance, etc., based on the Cybersecurity Requirement (TCR/HCR/SCR).

Purpose:

Define requirements for external systems and manufacturing, operation/maintenance, etc.

Input:

\*Cybersecurity Requirement (TCR/HCR/SCR)

Deliverables:

\*External CS requirements definition document/response results report

Forms:

\* XXX

Work:

To specify the procedures for management transfer of requirements to external systems, Production / Plant and After-sales department.

* Deployment to Production / Plant and After-sales department

The XXX shall be issued to Production / Plant or After-sales department, requesting them to fill in the contents to be addressed in the production, operation and maintenance phases. XXX shall contain the ID of the cybersecurity requirement (TCR/HCR/SCR).

* Deployment to other divisions

Issue XXX and request other departments to fill in the content to be addressed.

## Creation of Technical Cybersecurity concept (TCC)

Purpose:

The technical CS concept (TCC) is described by visualizing the assignment of Technical Cybersecurity Requirement (TCR) to HW and SW elements.

Input:

\*Technical Cybersecurity Requirement (TCR)

Deliverables:

\*Technical Cybersecurity Requirement Statement (Technical Cybersecurity Concept (TCC))

Forms:

\* XXX

Work:

A functional block for realizing a technical CS request (TCR) is specified, detailed and arranged at a hardware/software level. At this time, when it is determined that a functional block not included in the system basic design is necessary, the functional block is newly added and arranged in the system basic design.

# System design

A technical CS requirement (TCR) is arranged in hardware and/or software, and a CS HSI definition document defining interaction of hardware and software and a system design document reflecting a technical CS concept (TCC) are defined.

## CS HSI Definition

Purpose:

\*Define the interface specification between the functions of the system-level hardware software to which the Technical Cybersecurity Requirement (TCR) are assigned as the hardware-software-level specification.

Input:

\*Technical Cybersecurity Requirement Definition Document

Deliverables:

\*CS HSI definition document

Forms:

\* XXX

Work:

Some of the technical CS requirement (TCR) derived at the system level require the cooperation of hardware (HW) and software (SW). For such technical CS requirement (TCR), the hardware (HW) and software (SW) interface specification are defined as HSI specification.

Specifically, the hardware HSI is interrupt, port allocation, various control registers, etc., and the software HSI is interrupt processing, port control, control specification of various registers, API, etc.

Since the HSI Definition Document itself is created as various specification, it should be created as an arbitrary format specification document, and the reference document information of the HSI Definition Document should be described in the CS HSI form.

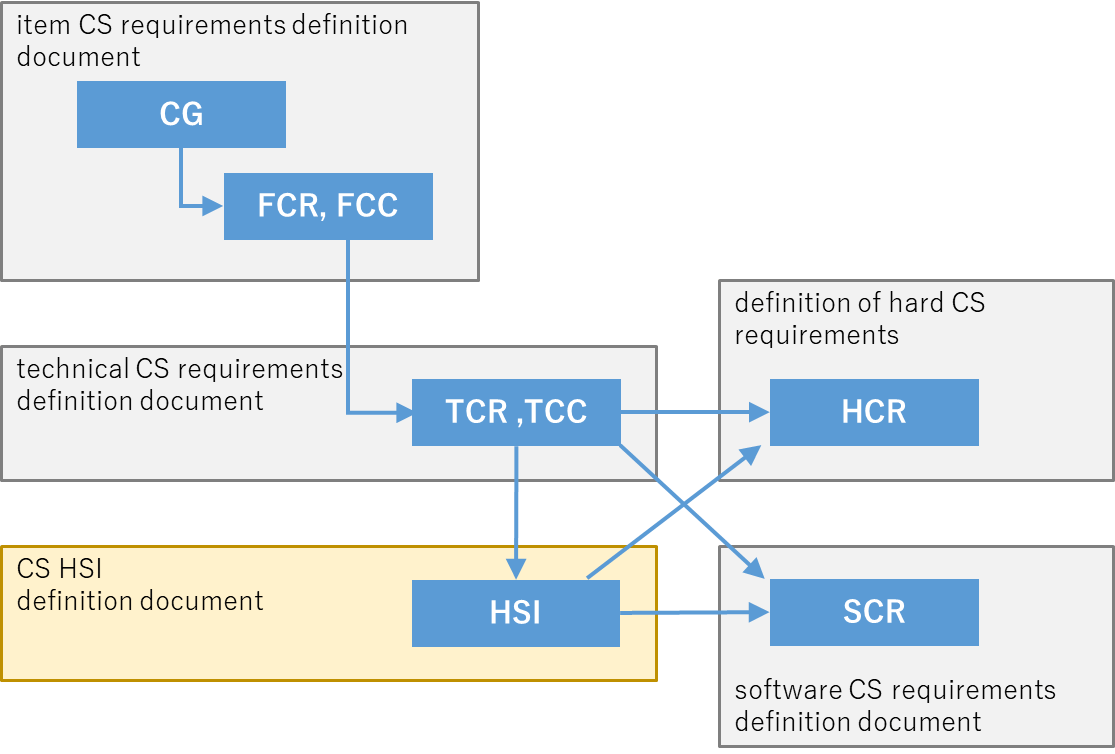


Figure8‑1 CS HSI Request Positioning

## System design

Purpose:

To design a system level that includes safety requirements and CS requirements (TCR/TCC) in addition to requirements of original functions.

Input:

\*Technical Cybersecurity Requirement Definition Document

Deliverables:

\*system design document

Work:

To reflect a TCR to a system basic design by considering validity at a hardware/software level.

### CS requirements for system design

To design a system level incorporating safety requirements and CS requirements in addition to requirements of original functions. At the system level, when safety requirements and CS requirements are met in addition to the requirements for original functions, design at that level is carried out. It is the same as general development activities except that CS requirements are included in the design requirements. The design documents shall be described so as to show how CS requirements (TCR) are reflected in the design.

Reflect improved CS requirements in the design (TCR) or TCRs added through system vulnerability analysis in the system design document.

### CS Architecture Design Principles

Confirm the security policy specified by the FCE (Guidelines for secure design and implementation grouped by system platform and interface), and review and apply it to each project.

### Security architecture design

Consider a security architecture to prevent the security function itself from being bypassed or tampered with, and describe it in the system design document. If consideration at the software level is required, it should be described in the SW requirement definition document and the SW architecture design document as considerations at the time of detailed design and implementation.

#### Secure security features

Consider mechanisms to prevent the security functions themselves from being bypassed or tampered with.

Example of Measures to Prevent Bypass)

* + Close unnecessary interfaces
  + Enable security features to work with parameters that are not in the expected order or range

\*The concrete implementation method is examined in architectural design and detailed design.

Examples of measures to prevent tampering)

* + Check data contents beforehand for input from programs other than security functions.
  + Clarify the dependencies between the product and the external environment, and confirm the actions that can be performed on the product side.
  + Isolate security domain (To be described later) + protect input to security features

\*The concrete implementation method is examined in architectural design and detailed design.

#### Define Security Domain

Define the security domain as follows:

* Whether to place features that are not related to security features in a security domain
* A feature placed in a security domain is a list of resources that the feature has access to

\*Security domain does not include security features or assets to be protected

* Features that do not reside in a security domain justify the need for a security domain

This requirement defines the scope and environment (Security Domain) that isolates potentially vulnerable programs (Function) from attack or from the impact of attack on security features. In addition, the data for handling personal information should be stored in secure storage, or measures such as encryption or obfuscation of the data should be considered for data exchange.

Note: Understanding Security Domains

To prevent unauthorized influence on security functions and protected assets by confining untrusted program operations without giving unnecessary authority and separating domains.

A security domain is an underlying concept that prevents bypassing or tampering with security features, and building a security architecture (Mechanisms to protect against attacks so that security functions work correctly) on top of it leads to an efficient, robust implementation. By limiting the route, it becomes difficult to bypass the security function. However, measures to prevent bypass and tampering are required for a regular interface.

Example of how to implement security domain definition)

* + Defined by CPU operating mode:

- Security policy settings, etc.

* + Defined by logical address space:

- Using the separation kernel (INTEGRITY from Green Hills Software)

* + Software defined:

- Hypervisor Usage

* + Defined by time management:

- Processor Usage Management

* + Defined by network management:

- Network segments, use of IPsec, etc.

#### Domain separation mechanism

8.2.3.2The implementation method (separation method) of the security domain is described in relation to the result of the security domain definition created in.

\*This requirement is omitted for functions that do not require a security domain.

In this section,8.2.3.2Based on the definition of the security domain, it defines how the program (Function) which determines that the security domain is required implements the security domain. Clarify how security domains are implemented to ensure that their functionality is a restricted mechanism that prevents access outside of the security domain.

Example of how to achieve security domain isolation)

* + Enabled by CPU operating mode:

- Use hardware provided protected processing modes

- Security policy settings

* + Realization with Logical Address Space:

- Memory Management

* + Software Enabled:

- Hypervisor Usage

* + Time Management Enabled:

- Manage and constrain processor usage per domain

* + Network Management Enabled:

- Network segment segmentation by communication type

#### CPU operation mode

When it is defined that the security domain is realized by the CPU operation mode, the CPU operation mode in which the function is arranged is described. The CPU operation mode is one of the ways to realize a security domain.

\*This requirement is omitted for functions that do not require a security domain. If the security domain is realized by a method other than the CPU operation mode, this requirement is omitted.

For example, if the chip is based on the ARM architecture, indicate whether the function is placed in the secure area Secure World using Trust Zone technology or the normal area Normal World.

Note:What is the Trust Zone?

It is a security technology used in Cortex-A processors from ARM Ltd. Trust Zone implements two virtual cores on a physical core, creating two domains. Therefore, the resources in the secure area cannot be accessed from the normal area and are protected at the hardware level. A buffer overflow attack in the normal realm will not access memory in the secure realm. However, since the operation timing is different between the normal area and the secure area at the time level, there may be a latency that is not acceptable for the throughput in a function that emphasizes synchronization and real time.

Remarks:

The CPU operation mode may be described in the column where the definition of the security domain and the determination of the domain separation mechanism are described.

#### Point of entry for the attacker

Check the entry point of the attacker for this function by referring to the technical CS requirement definition document, and confirm that the following are considered for the possible entry point.

* Unwanted interfaces must be closed
* Do not create intrusion points that are not stated in the design document (For testing and debugging) (Be sure to erase before shipping.).
* The components that are authorized for Secure World are minimal.

\*For chips with ARM architecture

* Implementing a reduction in the amount of code that is executed by default
* Limit the range of users and IDs that can access the code
* Implement code privilege reduction

Note:

Intrusion points include all interfaces, protocols, and executable code. Ingress points should be limited to those absolutely necessary for the function and should be minimized.

The purpose of this requirement is to review the point of entry of the product in terms of whether it is really necessary and to prevent unnecessary points of entry from being created or remaining at the time of shipment. The purpose of this study is to reduce the risk of being targeted by an attacker, such as the existence of an unexamined break-in point for a product, the custom creation of a break-in point that is not necessary for that function, the existence of an unnecessary break-in point for this product due to the diversion of a function for another product, or the product being shipped without the existence of a break-in point for testing and debugging that is not described in a design document by a developer. Especially for test and debug points, make sure to delete the points before shipping the product and make sure not to create any undocumented intrusion points in the design document.

#### Log Design

Determine whether security logging is required for the function and design the log.

Note: Understanding the Security Log

The log records events such as identity authentication, access authorization, handling of critical resources, and use of privileges, and helps detect signs of security breaches and collect and analyze evidence after breaches have occurred.

#### Log security measures

Implement a mechanism for ensuring a trail through log recording and measures against leakage of logs themselves.

Sidebar 1: Trailing

An attacker who has penetrated a system may manipulate the system so that his or her own trace does not remain in the log. A mechanism to prevent such maneuvering is called a mechanism to secure a trail.

There are the following threats, and measures against them are necessary.

* + Log Tampering:

Attackers can interfere with logging and remove traces of their own fraud. Therefore, protection measures against illegal writing and deletion are necessary.

* + Unlogged:

An attacker could fill the log with a large number of requests in advance, and then break into the system without leaving a log. Therefore, measures against full capacity are necessary.

Example of a mechanism for securing a trail)

* + Use a write-once storage device that cannot be rewritten or deleted. or give only the minimum necessary write permission (follow the principle of least privilege).
  + Alerts when a log may exceed its capacity. Otherwise, if the storage capacity is exceeded, it is returned to the beginning and overwritten to prevent the occurrence of unlogged data.

Sidebar 2: Log Leak Prevention

There is a high possibility that log data itself, user information, important information, and the like are recorded, and the leakage of log data by access or attack from an outsider becomes a security problem.

Examples of log leakage countermeasures

* + Do not log key access passwords (PIN) or other important information
  + When more important information is recorded, the entire communication is encrypted and recorded.
  + The value of the authentication result data prevents spoofing by issuing a different value each time authentication is performed.

## System safety impact analysis

At the system level, the mutual effects of safety requirements and CS requirements are analyzed through review and both requirements are adjusted. In the absence of functional safety requirements (The safety goal is not set and the FSR is not placed.), this analysis is unnecessary.

For example, if the requirement of "Reset the microcomputer if an attack is detected" is defined in the CS requirement, the safety mechanism is stopped and the safety requirement is adversely affected. On the other hand, if a request of "Turn off power to QM circuits when a hazardous condition is detected" is defined in the safety request, the CS function included in the QM area is stopped, which has an adverse effect on the CS request. Review the interaction between CS requirements derived at the system level and safety requirements. If there is an effect, it is adjusted between both requests.

Purpose:

Review the safety impact of CS requirements and adjust CS requirements.

Input:

\*Technical Cybersecurity Requirement Definition Document

Deliverables:

\*System Safety Impact Analysis Report

Forms:

\* XXX

Steps:

Refer to XXX. The Technical Cybersecurity Requirement (TCR) and the technical CS concept (TCC) shall be adjusted according to the review results, and the adjusted results shall be reflected in the Technical Cybersecurity Requirement definition document. In addition, it is reflected in the CSHSI definition document and the system design document according to the result of reflection in the technical CS requirement definition document.

## System vulnerability analysis

The conceptual diagram of system vulnerability analysis is shown in Figure8‑2 .

Conduct a vulnerability analysis at the system level to identify the attack methods that infringe the CS requirements and the vulnerabilities that can be exploited (1). Countermeasures are considered for the identified attack methods and vulnerabilities, and the countermeasures are defined as new CS requirements (2).

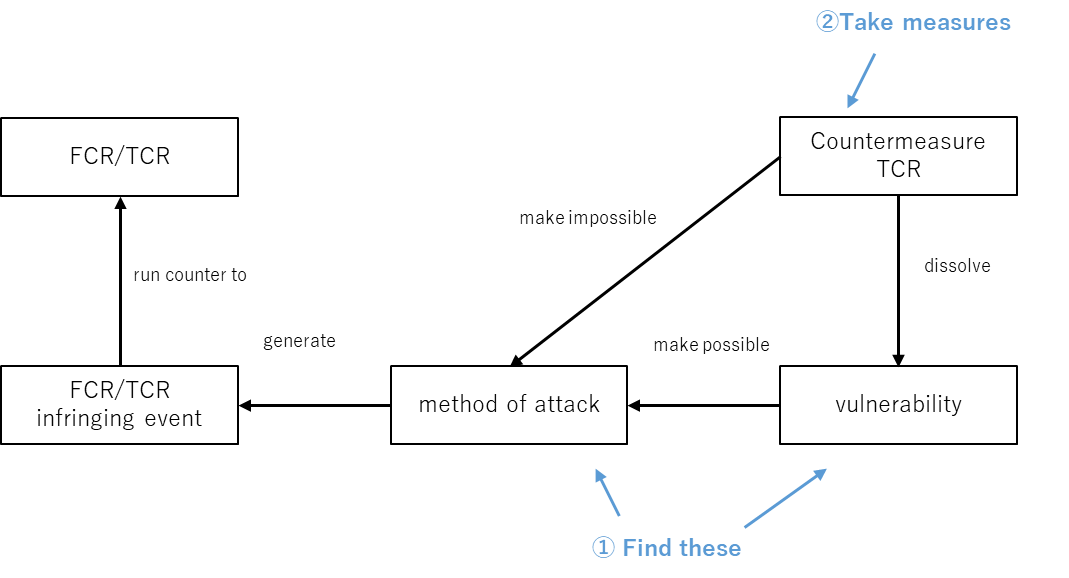


Figure8‑2 Conceptual diagram of system vulnerability analysis

The XXX in the product development phase uses the same analysis methodology as the "Threat Analysis and Risk Assessment (TARA)" in the concept phase. The analysis methodology and the procedure for creating work products are described in XXX.

The following figure shows the relationship between CS requirements to be derived in the concept phase and product development phase and the analysis procedures to derive the requirements.

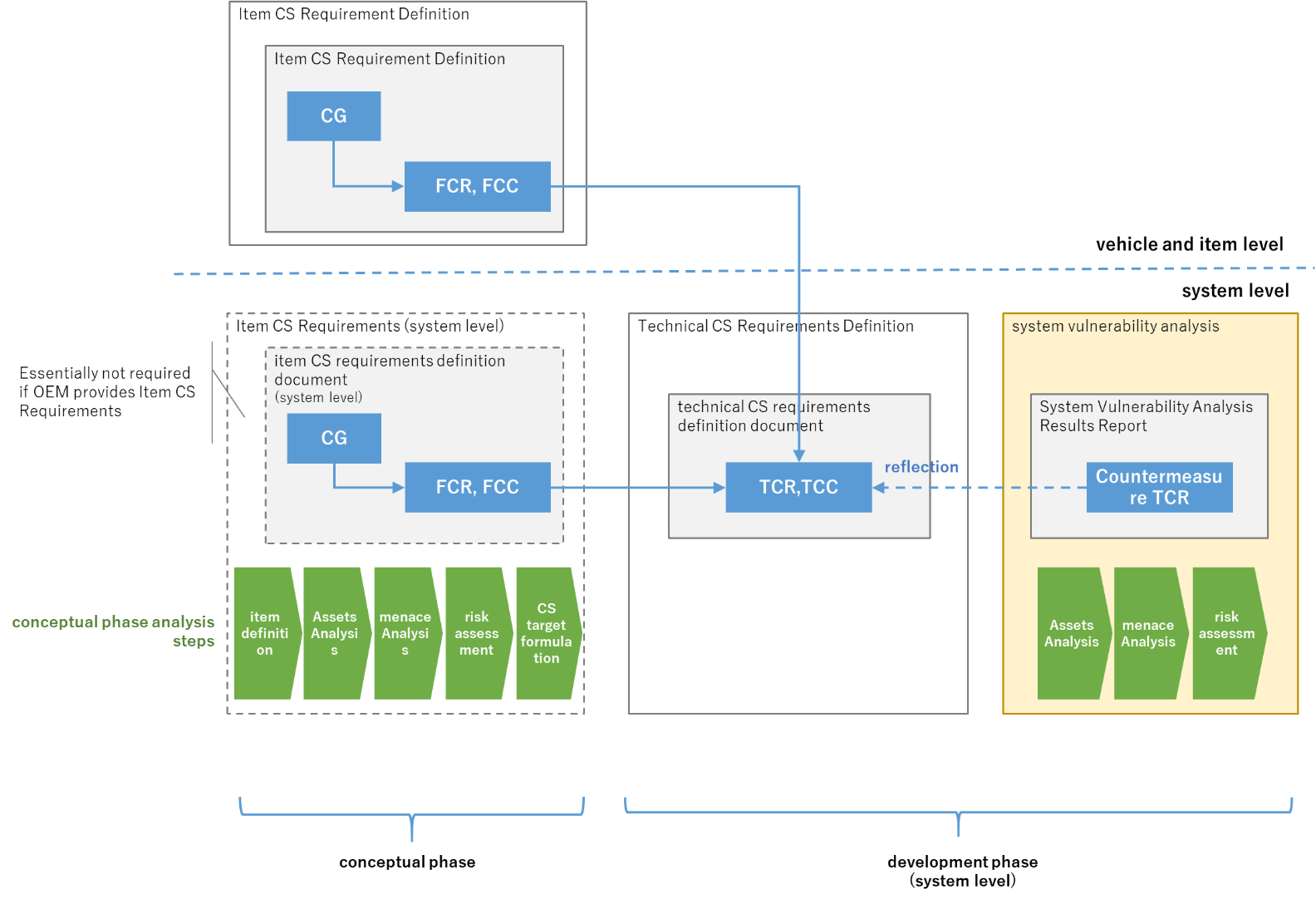


Figure8‑3 CS requirements and analysis procedures for each phase

Purpose:

Identify attack methods and vulnerabilities that violate functional CS requirements (FCR)/Technical Cybersecurity Requirement (TCR) and define Technical Cybersecurity Requirement (TCR) as needed.

Input:

\*system design document

Deliverables:

\* Asset Analysis Report

Forms:

\* XXX

Steps:

<Approach to System Vulnerability Analysis>

Here, FCR/TCR intrusion event is considered as a intrusion of a security protected asset, and an asset intrusion event in the system is identified. A threat scenario causing an asset intrusion event is derived, and an attack path for realizing the threat scenario is specified. In addition, identify vulnerabilities that can be exploited in that attack path and assess their attack feasibility.

<method of implementation>

Implement the procedures

### Identification of assets

Create a evaluation target model diagram (DFD) based on the system configuration diagram to identify primary and secondary assets. When conducting asset analysis at the system level, the security functions to be realized by FCR/TCR and their related information (data) should be derived as secondary assets.

* primary asset

Information used by non-security features (Data) and the features themselves. When a CS request is placed in the behavior of a primary asset, the asset becomes the primary asset.

* secondary asset

The information (Data) used by the function to protect the primary asset and the function itself. The function to realize only CS request is regarded as a secondary asset.

### Identify threat scenarios

Conduct a threat analysis to identify threat scenarios.

### Identify Attack path

Conduct attack path analysis and vulnerability analysis to identify attack paths, and conduct risk analysis/risk assessment/risk management if vulnerabilities are found that could lead to FCR/TCR intrusion events.

### Risk analysis/Risk assessment/Risk management

Decide how to respond to vulnerabilities through risk analysis/risk assessment/risk management. If security countermeasures are to be taken, the countermeasure TCR should be reflected in the technical CS requirement definition document, and the system design incorporating the countermeasure TCR should be reflected in the CS HSI definition document and system design document.

### Anomaly response

Prompt communication to stakeholders such as OEMs through an escalation path should any of the following occur:

* A vulnerability or attack method that is difficult or unreasonable to deal with by the system in charge alone was found.
* It does not violate functional CS requirements (FCR) and Technical Cybersecurity Requirement (TCR), but has discovered vulnerabilities and attack methods that could lead to some form of security damage.

## System CS Test Item Definition

Define test items to verify that CS requests are correctly implemented at the system level.Figure8‑5 Relationship between CS requests and CS testsshows the relationship between the CS request and the test item.

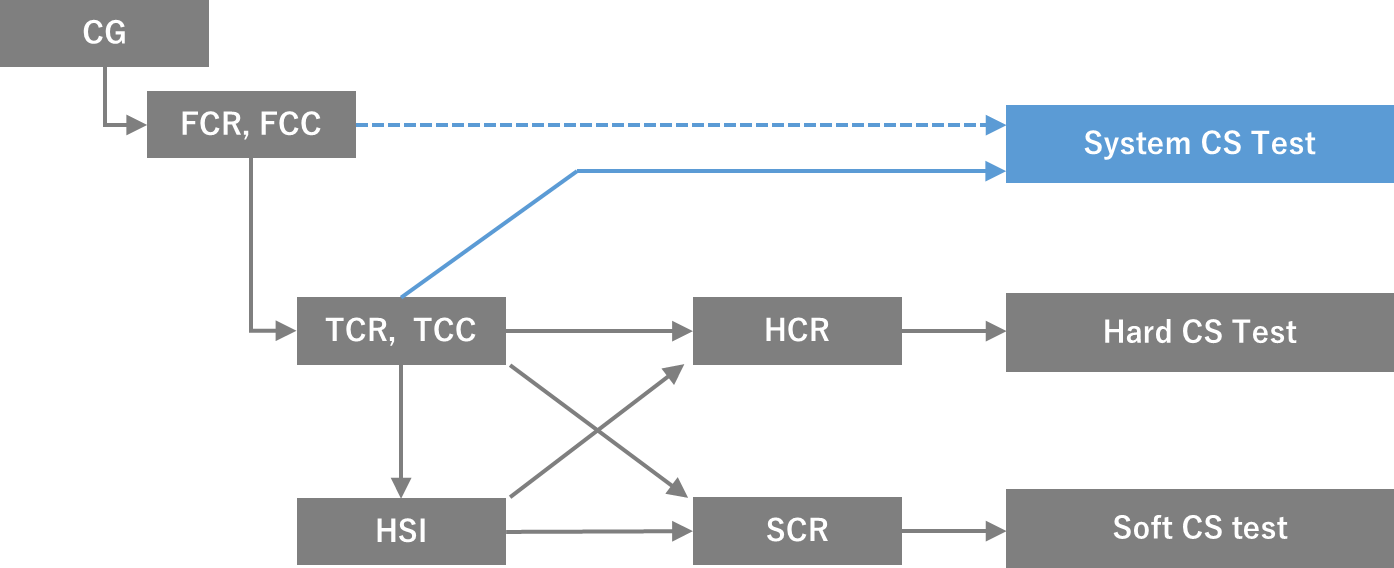


Figure8‑6 Relationship between CS requests and CS tests

Purpose:

Define test items to verify that functional CS requests (FCR)/technical CS requests (TCR) are implemented correctly.

Input:

\*Technical Cybersecurity Requirement Definition Document

Deliverables:

\*System CS Test Item Definition and Results Report

Forms:

\* XXX

Steps:

For each technical CS request (TCR), create a test item using a request-based test. If there is a technical CS requirement that does not create a test item (TCR), provide a reasonable explanation that the test item is not necessary. If necessary, create corresponding test items for Functional Cybersecurity Requirements (FCR) that are determined to require testing.

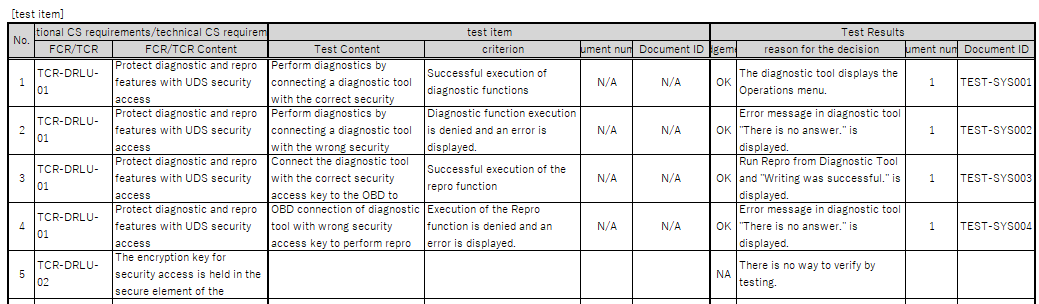


Figure8‑7　System CS Test Item Definition and Results Report

## Vulnerability test item definition

Vulnerability testing is a test conducted to verify that the assets (information, features) of a product to be protected are sufficiently protected against a threat in security risk management through product development.

The vulnerability testing aspect of this document focuses on validating the existence of vulnerabilities that could be exploited by potential threats.

Purpose:

Determine which vulnerability tests to perform.

Input:

\*item CS requirements definition document

Deliverables:

\*Vulnerability test item definition document

Steps:

In terms of vulnerability testing, vulnerability test items are specified referring to the vulnerability test requirements defined in ASPAR XXX. The vulnerability test items to be executed may be specified by the OEM, and in such case, the specified vulnerability test items shall be followed. If not specified by the OEM, select a vulnerability test item from the ASPAR XXX.

<How to Choose Vulnerability Test Requirements>

* Applying Vulnerability Testing Requirements

The application of the test requirements shall be determined by comparing the function of the ECU to be tested with the implementation conditions of the vulnerability test requirements (following test requirements) indicated in XXX. In addition, the implementation conditions and items to be confirmed in each test requirement shall be selected and applied within the necessary and sufficient scope in consideration of the security requirements and functional requirements imposed on the ECU to be tested.

* Classification of vulnerability testing requirements

around the subject to be testedTable8‑1 Classification of Vulnerability Test Requirements (\* 1)Classification of test requirements in terms of.

Table8‑1 Classification of Vulnerability Test Requirements (\* 1)

|  |  |  |  |
| --- | --- | --- | --- |
| Classification | Description | Examples of possible threats | typical entry point |
| Hardware | Testing that mainly checks for vulnerabilities in debug ports and debugging functions, such as JTAG connectors and persistent microcomputer debug pins. | Firmware extraction from debug port, etc. | JTAG connector  Debug pin of microcomputer |
| netwrok | Test mainly to check whether interface and network communication functions are vulnerable, such as checking the settings of wireless LAN devices and Bluetooth devices, checking the remaining unnecessary communication services, and checking the settings of TLS communication. | Eavesdropping by decrypting encrypted data using WEP and connections that abuse remaining services | Wi-Fi  Bluetooth  IP communication  CAN communication |
| Application | Tests to check for vulnerabilities in applications, mainly web applications and diagnostic applications, such as checking for injection vulnerabilities and user access control  \*except for debugging through the debug port | OS command injection, etc. | Input form of the in-vehicle application |

<Vulnerability testing process>

As specified by the OEMTable8‑1 Classification of Vulnerability Test Requirements (\* 1)Basically, vulnerability tests should be conducted at the hard level for classification "Hardware", at the soft level for classification "Application", and at the system level for classification "netwrok".

In the case of requirements that are considered to be better to be implemented at a level other than the above, the implementation process of the relevant test requirements shall be changed as necessary in agreement with the ordering party of the test requirements (OEM, etc.).

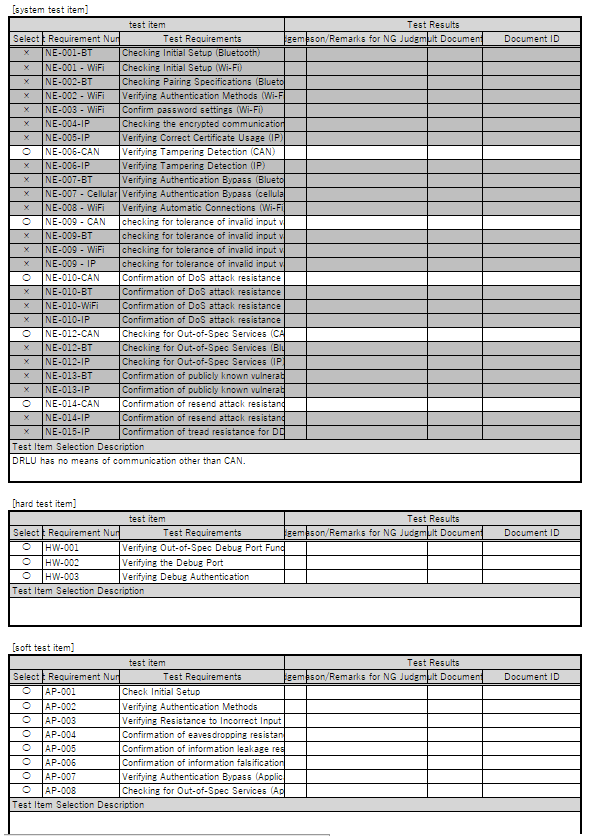


Figure8‑8 Vulnerability Test Item Definition and Results Report

## Verification of system CS design

It is necessary to confirm the validity and appropriateness of the deliverables prepared at the system level, and verification is carried out for that purpose.

Purpose:

Review system-level design deliverables.

Input:

\*system design document

Deliverables:

\*System CS Plan Verification Report

Forms:

* XXX

Steps:

# System-level validation

Testing from a CS perspectiveFigure9‑1 CS Test OverviewThere are two styles, "CS Request Based Functional Testing" which is made with the green color flow and "vulnerability testing" which is made with the orange color flow. In building a test environment and conducting CS tests, a test environment is built to conduct tests of both systems, and tests are conducted.

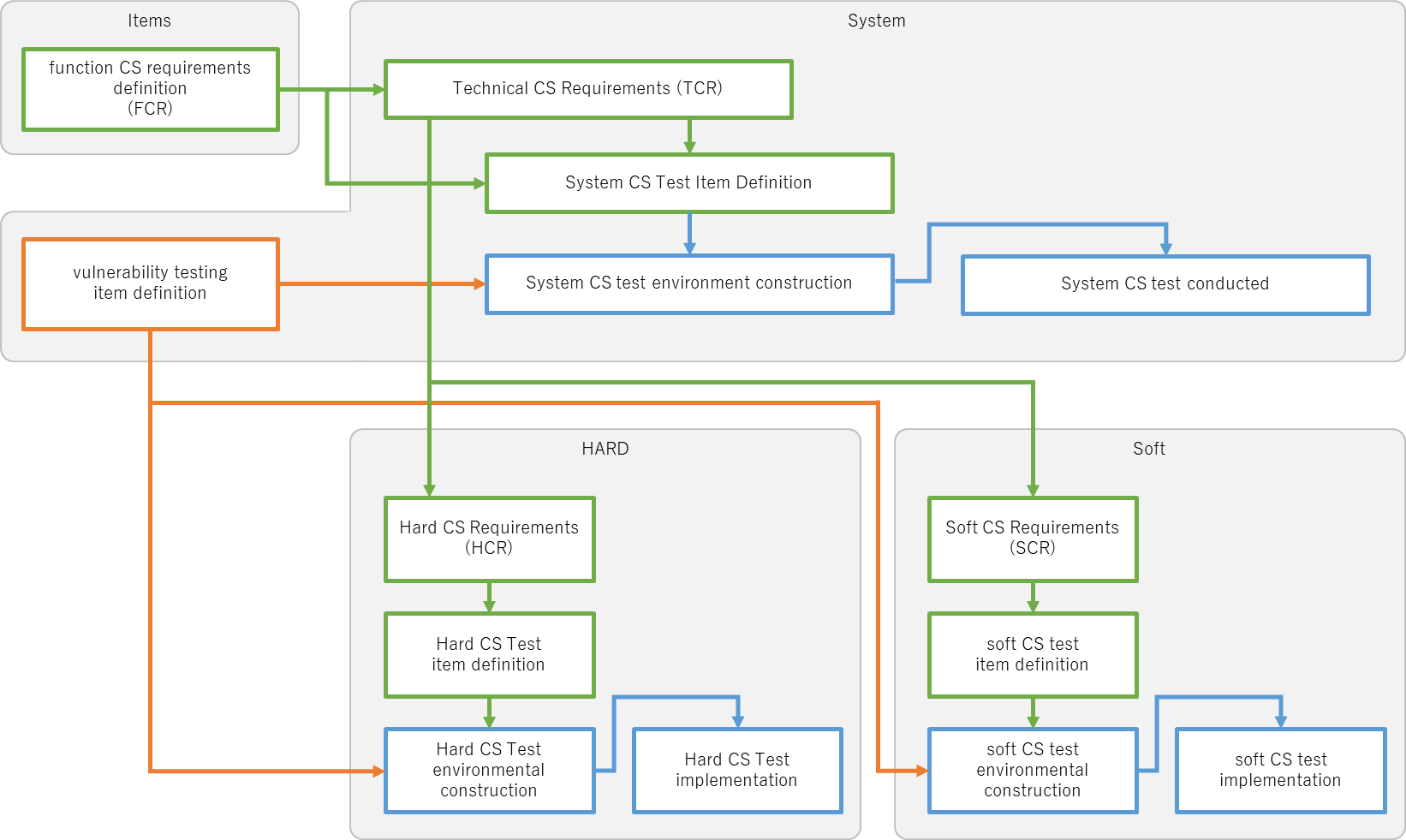


Figure9‑1 CS Test Overview

## System CS test environment construction

Purpose:

Build a system-level test environment.

Input:

\*System CS Test Item Definition Document

Deliverables:

\*System CS Test Environmental Report

Steps:

Build a test environment to perform the tests described in System CS Test Item Definitions and Vulnerability Item Definitions (system level).

Create a System CS Test Environment Report after the completion of the system-level CS test environment construction. The following points should be explained in the report.

1. All test validation items can be performed in the test environment provided
2. The test results are based on evidence of the methods used to record and store data.

A bad example: A person visually checks the output of the inspection tool and writes the result of OK/NG.

Good example) Save the output of the inspection tool as a file and write the OK/NG result.

1. The test results must be reproducible.

Example)Version records of test objects, inspection tools, etc., storage of random test data, etc.

## System CS test conducted

Purpose:

Conduct and document CS request-based functional and vulnerability tests.

Input:

\*System CS Test Item Definition Document

Deliverables:

\*System CS Test Results Report

Forms:

\* XXX

Steps:

Perform the tests described in System CS Test Item Definition and Vulnerability Test Item Definition (system level).

Record test results in the System CS Test Results Report and the Vulnerability Test Results Report. If the basis data for judgment of the test results cannot be sufficiently described in these CS forms, separately prepare evidence materials describing the test results in detail and refer to them from the CS forms. The evidence materials do not need to be submitted, but they should be stored and managed so that they can later respond to OEM inquiries. If the test results show an NG, return to the appropriate process to correct the problem.

## Verification of system CS test results

Deliverables created at the system level need to be validated and assessed.

Purpose:

Review system-level CS test results.

Input:

\*System CS Test Environment Construction Report

Deliverables:

\*System CS Test Environmental Report

Forms:

\*XXX

Steps:

# CS validation

## CS Validation Phase Overview

This chapter describes activities related to cyber security verification for vehicle-level items. This activity will take place after the integration of components is completed. This item is verified in a vehicle-level operating environment with all settings intended for manufacturing and operation vehicle use, i.e., test functions disabled.

## CS Validation Phase Objectives

The purpose of the CS validation phase is to:

1. Review Cybersecurity Goalss (CG) and CS claims
2. Verify items meet CS objectives (CG)
3. Verify that the residual risk of the item is acceptable

## CS Validation Plan

Purpose:

Prepare a test plan and prepare a plan for CS validation.

Input:

\*item CS requirements definition document

Deliverables:

\*overall test plan

Steps:

The implementation of CS validation is planned in the system CS plan and the overall test plan. The aspects of system-level validation are described below.

* Verification of CG targets (CG) and CS claims

Where CS objectives (CG) and CS claims have been developed by the OEM, the need for implementation will be determined through coordination with the DIA and the customer.

When carrying out verification of CG targets (CG) and CS claims at the system level, the test plan should be prepared in the overall test plan, and the specific plan should be described in the validity test plan.

* Validation of CS objectives (CG)

Perform system-level penetration testing as part of validation of CS objectives (CG). Where CS objectives (CG) and CS claims have been developed by the OEM, penetration testing at the vehicle or item level will be determined by coordination with the DIA and the customer.

The execution of the penetration test at the system level is planned in the system CS plan and the whole test plan, and the concrete plan is described in the penetration test plan.

Penetration testing is basically required if the CS assurance level (CAL) specified in the system CS project plan is CAL3 or higher.

Implementation of CS validation

Purpose:

Conduct validation according to the CS validation test plan and record the results.

Input:

\*System design document

Deliverables:

\*CS Validation Report

Steps:

Conduct CS validation based on the developed test plan. Vulnerabilities identified by CS validation are managed by vulnerability management.

# System Forms：

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Form Title | Application systems | Path |
| 1. |  |  |  |
| 2. |  |  |  |

# References：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | DMS Document No. | Document Title | Version | Storage unit |
| 1. |  |  |  |  |
| 2. |  |  |  |  |