関係各部署　御中

To departments concerned

Confidentiality classification	PROTECTED 関係者外秘	原紙保管 Storage of original	M/Y /
		コピー保管 Storage of copy	M/Y /

侵入検知　エントリーポイント向け Host IDS　要求仕様書 Requirements Specification of Host-based IDS for Entry Point

制御電子プラットフォーム開発部
制御ネットワーク・アーキ開発室 4G
System Network & Architecture Development Dept. 4G
E/E Architecture Development Div.

No. SEC-ePF-IDS-HIE-REQ-SPEC-a00-05-a

承認 Approved by	調査 Checked by	作成 Created by	
平林	松井	竹山	2022/11/25

適用先 Target	エントリーポイント ECU/VM のうち、別文書にて定義される特定の ECU/VM Allocated to entry-point ECUs / VMs specified by another document.
特記 Special note	【展開規則　Distribution rule】 必要に応じて、関係会社・関係部署（海外事業体、ボデーメーカ、ECU サプライヤ）への展開をお願いします。 Please distribute this document to affiliated companies, or departments (e.g. overseas business entities, car body manufacturers, or ECU suppliers) if necessary. 【問合せ先　Contact information】 制御電子プラットフォーム開発部　制御ネットワーク・アーキ開発室 セキュリティ仕様問合せ窓口 System Network & Architecture Development Dept. E/E Architecture Development Div. Contact for Security Inquiries email: epf-sec-sp@mega.tec.toyota.co.jp

トヨタ自動車株式会社

変更履歴

Version	変更内容	日付	変更者
a00-00-a	新規作成	2021/04/05	46F 4G 稲垣
a00-00-b	英訳追加	2021/05/14	46F 4G 稲垣
a00-01-a	要求具体化、可読性向上	2021/08/06	46F 4G 竹山
a00-02-a	死活監視の SEv 生成に関わる要求を削除 バージョン読み出し機能に関わる要求を削除	2021/12/03	46F 4G 竹山
a00-03-a	参照文書を追加 T.B.D.の記載を削除 QSEv 保管の要求を修正 IDSHER_07208 追加 IDSHER_02101 の検知技術例修正 IDSHER_04101 修正 IDSHER_02301 修正 IDSHER_07102 修正 IDSHER_07108 修正	2022/02/03	46F 4G 竹山
a00-04-a	・要求一覧にハードウェア関連要件の列を追加 ・IDSHER_02301 文言の修正 ・IDSHER_04101 文言の修正 ・IDSHER_07102 Context Data の項目の明確化 ・IDSHER_07108 文言の修正 ・IDSHER_07109 QSEv 保管の要求を変更 ・IDSHER_07111 UserDefineDTC, DID の要求追加 ・IDSHER_07110 QSEv 読み出しの SID を明確化 ・IDSHER_07202 削除 ・IDSHER_07204 QSEv 消去の SID を明確化	2022/04/29	46F 4G 竹山
a00-04-b	・IDSHER_07111 UserDefMemoryDTC の値修正 ・IDSHER_07110 ダイアグ仕様参照を追記 ・IDSHER_07204 ダイアグ仕様参照を追記	2022/05/20	46F 4G 竹山
a00-04-c	・表 2-2 誤記修正 ・IDSHER_12201 誤記修正 ・IDSHER_07109 補足の一部を要求として記載	2022/06/09	46F 4G 竹山

a00-05-a	・3.1.1. 検知機能に検知機能の説明を追加 ・IDSHER_04101 具体化のための文言の修正 ・IDSHER_01601 具体化のための文言の修正 ・IDSHER_01101 適用条件の追加 ・IDSHER_01102 適用条件の追加 ・IDSHER_01201 適用条件の追加 ・IDSHER_01202 適用条件の追加 ・IDSHER_01501 適用条件の追加 ・IDSHER_01502 適用条件の追加 ・IDSHER_01401 適用条件の追加、具体化のための文言の修正 ・IDSHER_02301 具体化のための文言の修正 ・IDSHER_12201 要求の具体化 ・3.1.1. 検知機能の各要求の補足説明を Appendix として追加	2022/11/25	46F 4G 竹山

目次

1. はじめに

1.1. 本書の目的

エントリーポイント向けホスト型侵入検知システム（以下、本システム）の目的は、エントリーポイントへの侵入またはその試みを検知し、記録することである。本システムによって記録されるログは、米国国立標準研究所（NIST）が作成したサイバーセキュリティ対策に関するフレームワークにおける「検知」機能（参照文書[1]）の実現に用いられる。この本システムの要求を定義することが、本書の目的である。

1.2. 適用先

本書は、エントリーポイント ECU/VM のうち、別文書にて定義される特定の ECU/VM に適用される。

1.3. 前提条件

無し

1.4. 要求事項の記載

【要求事項：**】と記載されているものが要求である。ここで、<補足>と記載されているものは単に補足事項であり要求ではない。

1.5. 関連文書

上位文書、参照文書を本節にて示す。なお、関連文書のバージョンは ECU の要求仕様書に従うこと。

1.5.1. 上位文書

表 1-1：上位文書

No.	文書名	Ver.
1	車両サイバーセキュリティコンセプト定義書	-

1.5.2. 参照文書

表 1-2：参照文書

No.	文書名	Ver.
1	Framework for Improving Critical Infrastructure Cybersecurity Version 1.1, National Institute of Standards and Technology, April 16, 2018, https://www.nist.gov/publications/framework-improving-critical-infrastructure-cybersecurity-version-11	1.1

2	QSEv 生成要求仕様書	-
3	AUTOSAR_PRS_IntrusionDetectionSystem	R20-11
4	AUTOSAR_SWS_IntrusionDetectionSystemManager	R20-11
5	AUTOSAR_SWS_AdaptiveIntrusionDetectionSystemManager	R20-11
6	車両サイバーセキュリティ及びプライバシー用語定義書	-
7	欠番	-
8	タイムスタンプ要求仕様書	-
9	TOYOTA Phase6 Diagnostics Communication and Reprogramming standard specifications	-
10	侵入検知 多層分離機能向け Host IDS 要求仕様書	-
11	侵入検知 検知マスタ要求仕様書	-
12	PF LAN 仕様書	-
13	車載 Ethernet 通信機能仕様書	-

1.6. 用語定義

本書で用いる用語を以下に示す。なお、本システムの一部は AUTOSAR にしたがい実装されることを想定しているため、本書では AUTOSAR で定義されている用語を利用する。それらの意味については参照文書[3]、[4]、[5]を参照されたい。その他用語については、参照文書[6]を参照されたい。

表 1-3：用語一覧

用語	解説
-	-

2. 要求概要

2.1. システムコンテキスト

本システムのシステムコンテキストをデータフローダイアグラムで示す（図 2-1）。円は本システムを、四角は本システムと情報やサービスのやり取りを行う主体を表す。

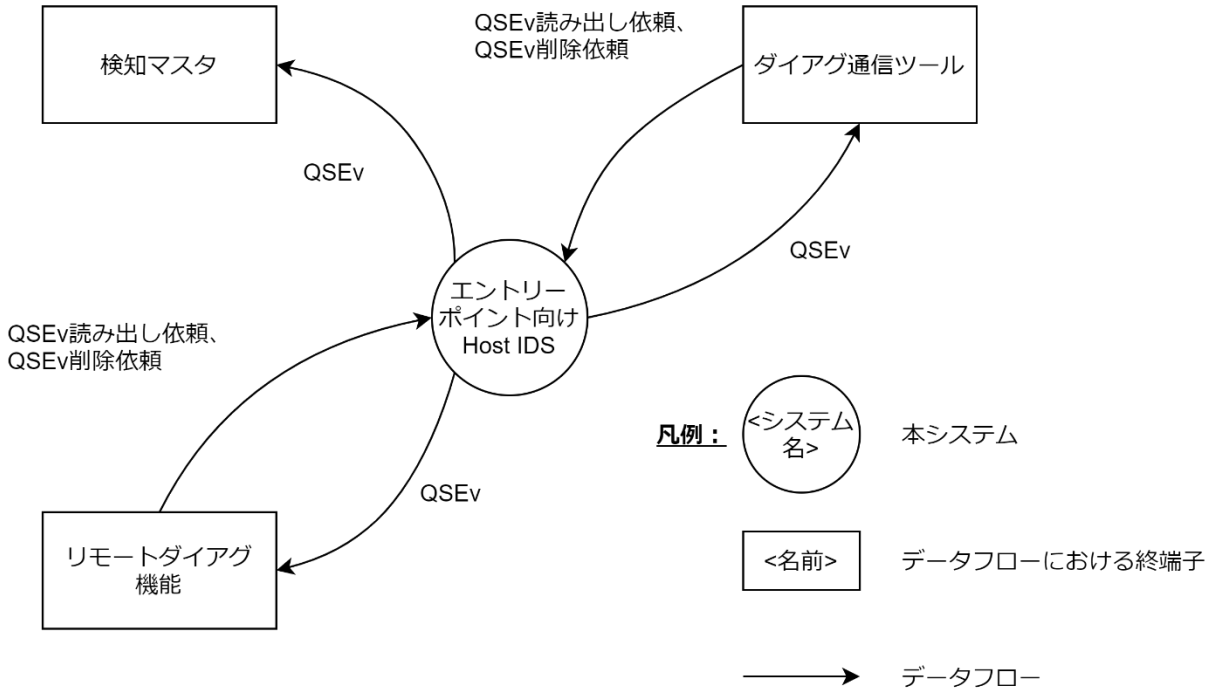

図 2-1：システムコンテキスト

2.2. システム動作概要

本システムは、表 2-1 に示す事象のいずれかが生じたとき、アクティビティ図（図 2-2）で示すとおりの動作をする。

表 2-1：本システムの動作始点となる事象

事象番号	本システムの動作始点となる事象
①	本システム搭載先 ECU・VM への侵入発生
②	本システムに保管されている QSEv の読み出し依頼
③	本システムに保管されている QSEv の削除依頼

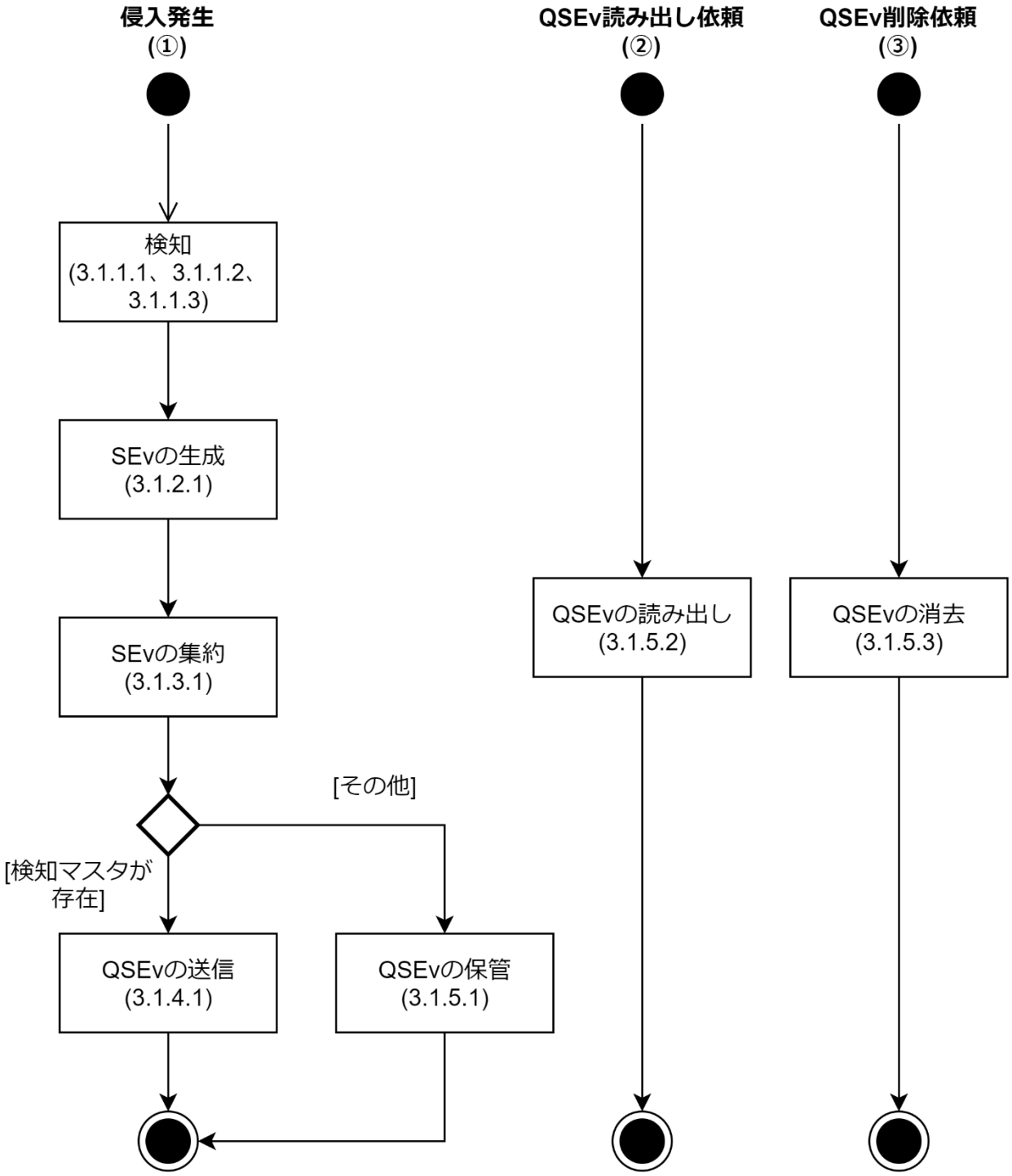
侵入発生
(①)
検知
(3.1.1.1、3.1.1.2、
3.1.1.3)
SEvの生成
(3.1.2.1)
SEvの集約
(3.1.3.1)
[その他]
[検知マスタが
存在]
QSEvの送信
(3.1.4.1)
QSEvの保管
(3.1.5.1)
QSEv読み出し依頼
(②)
QSEvの読み出し
(3.1.5.2)
QSEv削除依頼
(③)
QSEvの消去
(3.1.5.3)

図 2-2：本システム動作

2.3. 要求一覧

本書で定義する要求の一覧を表 2-2 に示す。また、ハードウェア選定時に参照すべき要件をハードウェア関連要求として示す。ハードウェアの採否は各要件に従うこと。

表 2-2：要求一覧

分類			要求 ID	ハードウェア関連要求
機能要求	検知機能	遠隔車外との通信に対する 1 層目防御機能の停止の検知	IDSHER_04101	No
		遠隔車外との通信を終端する機能の不正動作の検知	IDSHER_01601	No
			IDSHER_01101	No
			IDSHER_01102	No
			IDSHER_01201	No
			IDSHER_01202	No
			IDSHER_01501	No
			IDSHER_01502	No
			IDSHER_01401	No
		エントリーポイント領域の CSP/PSP・ソフトウェアの改ざんの検知	IDSHER_02101	No
			IDSHER_02301	No
	SEv 生成機能	SEv の生成	IDSHER_07102	No
	QSEv 生成機能	SEv の集約	IDSHER_07103	No
	QSEv 送信機能	QSEv の送信	IDSHER_07108	No
			IDSHER_07208	No
	QSEv 保管機能	QSEv の保管	IDSHER_07109	No
			IDSHER_07111	No
		QSEv の読み出し	IDSHER_07110	No
		QSEv の消去	IDSHER_07204	No
品質要求			IDSHER_12201	No
設計値			IDSHER_03401	No
			IDSHER_03402	No

3. システム要求

本章では本システムのシステム要求を定義する。

3.1. 機能要求

本節では機能要求を定義する。

3.1.1. 検知機能

検知は、図 3-1 で示すとおり、大きく分けて下記の三つの観点で行う。

1. 遠隔車外との通信に対する 1 層目防御機能の停止の検知（図 3-1 の No.1）
2. 遠隔車外との通信を終端する機能の不正動作の検知（図 3-1 の No.2）
3. エントリーポイント領域の CSP/PSP・ソフトウェアの改ざんの検知（図 3-1 の No.3）

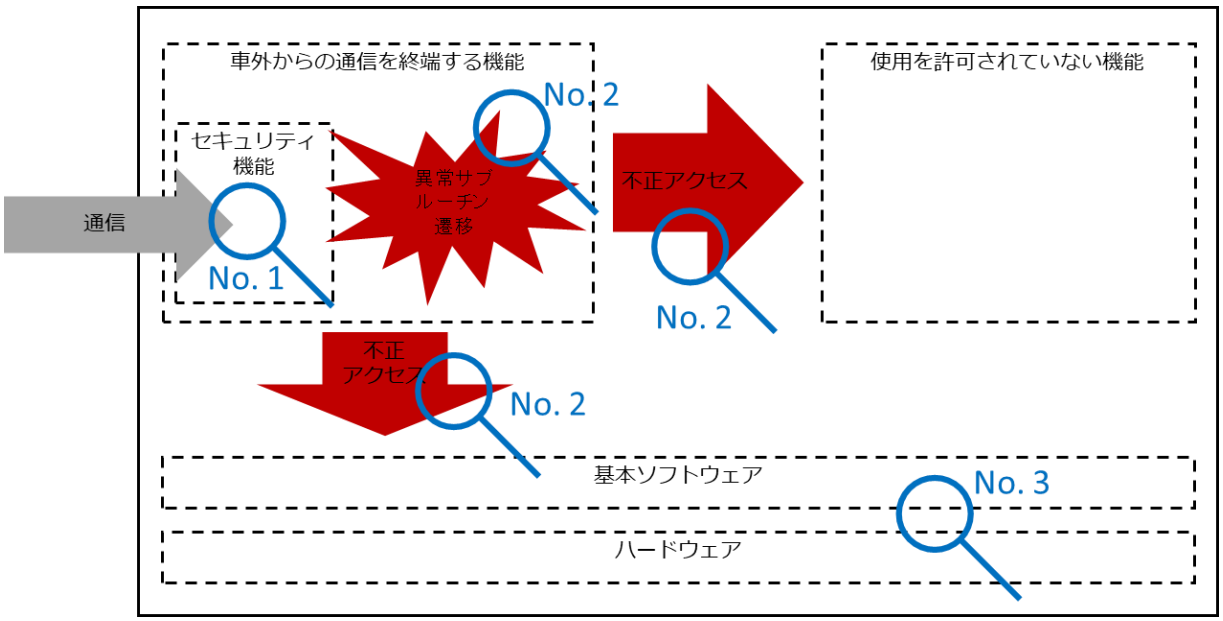

図 3-1：遠隔車外との通信を終端するエントリーポイント ECU/VM

ここで『遠隔車外との通信を終端する機能』とは、遠隔車外の接続先と直接的に通信を行う OS/OSS を利用したソフトウェアを指す。そして、当該機能は、遠隔車外との通信に対する 1 層目防御機能を持つ前提である。

一例として、TLS が『遠隔車外との通信に対する 1 層目防御機能』に、TLS 通信を終端するソフトウェアが『遠隔車外との通信を終端する機能』に、それぞれ該当する。また、ここで『TLS 通信を終端するソフトウェア』とは、例えば『TLS の機能を実現する共有ライブラリ(TLS ライブラリ)や、TLS ライブラリをロードする実行形式ファイル(Executable file)によって構成されるアプリケーションのプロセ

ス』を指す。
なお、仮に『その他のソフトウェア』が『TLS 通信を終端するソフトウェア』との通信を介して間接的に車外と情報の授受を行う場合でも、プロセスが異なるならば、『その他のソフトウェア』については『遠隔車外との通信を終端する機能』には該当しない。

また、本書はその末尾に、各検知機能の要求に対する、監視対象、監視すべき異常や要求を実現する例を記載した Appendix. A を持つ。

3.1.1.1. 遠隔車外との通信に対する 1 層目防御機能の停止の検知

【要求事項：IDSHER_04101】
遠隔車外との通信に対する 1 層目防御機能が常駐ソフトウェア(常駐プロセス)として設計される場合に、本要求事項は適用される。当該ソフトウェアが設計上、動作すべき状況において動作していない場合、検知機能は SEv 生成機能に異常を通知する必要がある。ただし、参照文書[10]で定義される『遠隔車外との通信に対する 1 層目防御機能の停止』(IDSHMR_04101)が適用されるならば、本システムの検知機能と参照文書[10]で定義される検知機能の両方が、同一の事象に対して、SEv 生成機能に異常を通知してはならない。すなわち、本要求で捉えられる、ある事象が発生した際、本システムの検知機能と参照文書[10]で定義される検知機能のいずれか片方のみが、SEv 生成機能に異常を通知する必要がある。
※ここで『動作していない』とは、当該ソフトウェアが実行されていない状態を指し、当該ソフトウェアのコードや設定ファイルの改ざんによりセキュリティ機能が無効化されている状態を含まない。

3.1.1.2. 遠隔車外との通信を終端する機能の不正動作の検知

3.1.1.2.1. 制御フローの異常検知

【要求事項：IDSHER_01601】
遠隔車外との通信を終端する機能を構成するソフトウェアの実行中に正規の制御フローとして起こりえない関数遷移が行われたまたは試みられたとき、検知機能は SEv 生成機能に異常を通知する必要がある。
※ここで『正規の制御フローとして起こりえない関数遷移』とは、関数ポインタまたはスタック上のリターンアドレスの改ざんに起因して発生する間接コール(関数ポインタを用いた関数呼び出し)またはリターンを指し、コードの改ざんに起因して発生する事象を含まない。

3.1.1.2.2. 不揮発性メモリへの不正アクセス検知

【要求事項：IDSHER_01101】
遠隔車外との通信を終端する機能を構成するソフトウェアがファイルシステムを利用しパスによってエントリーポイント領域の不揮発性メモリにアクセスできる仕組みを持ち、かつ、当該ソフトウェアの動作に不必要なアクセスが許可されないよう設計される場合に、本要求事項は適用される。当該ソフトウェアが、パスによって、下記の操作のいずれかをその操作が許可されていないエントリーポイント領域の不揮発性メモリに対して行ったまたは試みたとき、検知機能は SEv 生成機能に異常を通知する必

要がある。

- 読み出しアクセス
- 書き込みアクセス
- 実行アクセス
- 属性の変更

※ここで不揮発性メモリとは、MCU/SoC 内蔵やディスクリート型などの実装形態に依らず、不揮発性を持ってデータを保持することができる物理的または論理的なデバイスを指す。なお、HSM は IO(ペリフェラル)の一種であり、不揮発性メモリに該当しない。
※ここで属性とは、読み出しアクセス・書き込みアクセス・実行アクセスの可否を指す。

【要求事項：IDSHER_01102】
遠隔車外との通信を終端する機能を構成するソフトウェアがアドレスによってエントリーポイント領域の不揮発性メモリにアクセスできる仕組みを持ち、かつ、当該ソフトウェアの動作に不必要なアクセスが許可されないよう設計される場合に、本要求事項は適用される。当該ソフトウェアが、アドレスによって、下記の操作のいずれかをその操作が許可されていないエントリーポイント領域の不揮発性メモリに対して行ったまたは試みたとき、検知機能は SEv 生成機能に異常を通知する必要がある。

- 読み出しアクセス
- 書き込みアクセス
- 実行アクセス
- 属性の変更

※ここで不揮発性メモリとは、MCU/SoC 内蔵やディスクリート型などの実装形態に依らず、不揮発性を持ってデータを保持することができる物理的または論理的なデバイスを指す。なお、HSM は IO(ペリフェラル)の一種であり、不揮発性メモリに該当しない。
※ここで属性とは、読み出しアクセス・書き込みアクセス・実行アクセスの可否を指す。

3.1.1.2.3. 揮発性メモリへの不正アクセス検知

【要求事項：IDSHER_01201】
遠隔車外との通信を終端する機能を構成するソフトウェアがファイルシステムを利用しパスによってエントリーポイント領域の揮発性メモリにアクセスできる仕組みを持ち、かつ、当該ソフトウェアの動作に不必要なアクセスが許可されないよう設計される場合に、本要求事項は適用される。当該ソフトウェアが、パスによって、下記の操作のいずれかをその操作が許可されていないエントリーポイント領域の揮発性メモリに対して行ったまたは試みたとき、検知機能は SEv 生成機能に異常を通知する必要がある。

- 読み出しアクセス
- 書き込みアクセス
- 実行アクセス
- 属性の変更

※ここで揮発性メモリとは、実装形態に依らず、揮発性を持ってデータを保持することができる物理的または論理的なデバイスを指す。
※ここで属性とは、読み出しアクセス・書き込みアクセス・実行アクセスの可否を指す。

【要求事項：IDSHER_01202】
遠隔車外との通信を終端する機能を構成するソフトウェアがアドレスによってエントリーポイント領域の揮発性メモリにアクセスできる仕組みを持ち、かつ、当該ソフトウェアの動作に不必要なアクセスが許可されないよう設計される場合に、本要求事項は適用される。当該ソフトウェアが、アドレスによって、下記の操作のいずれかをその操作が許可されていないエントリーポイント領域の揮発性メモリに対して行ったまたは試みたとき、検知機能は SEv 生成機能に異常を通知する必要がある。
- 読み出しアクセス
- 書き込みアクセス
- 実行アクセス
- 属性の変更

※ここで揮発性メモリとは、実装形態に依らず、揮発性を持ってデータを保持することができる物理的または論理的なデバイスを指す。
※ここで属性とは、読み出しアクセス・書き込みアクセス・実行アクセスの可否を指す。

3.1.1.2.4. IO(ペリフェラル)への不正アクセス検知

【要求事項：IDSHER_01501】
遠隔車外との通信を終端する機能を構成するソフトウェアがファイルシステムを利用しパスによってエントリーポイント領域の IO(ペリフェラル)にアクセスできる仕組みを持ち、かつ、当該ソフトウェアの動作に不必要なアクセスが許可されないよう設計される場合に、本要求事項は適用される。当該ソフトウェアが、パスによって、下記の操作のいずれかをその操作が許可されていないエントリーポイント領域の IO(ペリフェラル)に対して行ったまたは試みたとき、検知機能は SEv 生成機能に異常を通知する必要がある。
- 読み出しアクセス
- 書き込みアクセス
- 属性の変更

※ここで IO(ペリフェラル)とは、実装形態に依らず、データの入出力ができる物理的または論理的なデバイスを指す。
※ここで属性とは、読み出しアクセス・書き込みアクセスの可否を指す。

【要求事項：IDSHER_01502】
遠隔車外との通信を終端する機能を構成するソフトウェアがアドレスによってエントリーポイント領域の IO(ペリフェラル)にアクセスできる仕組みを持ち、かつ、当該ソフトウェアの動作に不必要なアクセスが許可されないよう設計される場合に、本要求事項は適用される。当該ソフトウェアが、アドレス

によって、下記の操作のいずれかをその操作が許可されていないエントリーポイント領域の IO(ペリフェラル)に対して行ったまたは試みたとき、検知機能は SEv 生成機能に異常を通知する必要がある。

- 読み出しアクセス
- 書き込みアクセス
- 属性の変更

※ここで IO(ペリフェラル)とは、実装形態に依らず、データの入出力ができる物理的または論理的なデバイスを指す。

※ここで属性とは、読み出しアクセス・書き込みアクセスの可否を指す。

3.1.1.2.5. 機能の不正使用検知

【要求事項：IDSHER_01401】

エントリーポイント領域が、使用に際して何らかの権限を必要とする機能を持ち、かつ、遠隔車外との通信を終端する機能を構成するソフトウェアにおいて、その動作に不必要な機能の使用が許可されないよう設計される場合に、本要求事項は適用される。

当該ソフトウェアが、使用に際して何らかの権限を必要とする機能のうち、使用を許可されていない機能を使用したまたは使用を試みたとき、検知機能は SEv 生成機能に異常を通知する必要がある。

3.1.1.3. エントリーポイント領域の CSP/PSP・ソフトウェアの改ざんの検知

3.1.1.3.1. CSP/PSP の改ざん検知

【要求事項：IDSHER_02101】

エントリーポイント領域が CSP/PSP に該当するデータを不揮発性メモリに持つ場合に、本要求事項は適用される。当該データの使用時に当該データが改ざんされているとき、検知機能は SEv 生成機能に異常を通知する必要がある。

※ここで不揮発性メモリとは、MCU/SoC 内蔵やディスクリート型などの実装形態に依らず、不揮発性を持ってデータを保持することができる物理的または論理的なデバイスを指す。なお、HSM は IO(ペリフェラル)の一種であり、不揮発性メモリに該当しない。

※ここで使用時とは、不揮発性メモリに格納されている CSP/PSP を揮発性メモリまたは HSM に展開するとき、を含む。

3.1.1.3.2. ソフトウェアの改ざん検知

【要求事項：IDSHER_02301】

エントリーポイント領域に配置されるソフトウェアについて、それぞれのソフトウェアの起動時に下記のいずれかが改ざんされているとき、検知機能は SEv 生成機能に異常を通知する必要がある。ここで、改ざん検知は完全性が保証された領域から行われる必要がある。ただし、参照文書[10]で定義される『ソフトウェアの改ざん検知』(IDSHMR_01601) が適用されるならば、本システムの検知機能と参照文書[10]で定義される検知機能の両方が、同一の事象に対して、SEv 生成機能に異常を通知してはならない。すなわち、本要求で捉えられる、ある事象が発生した際、本システムの検知機能と参照文書[10]で定義

される検知機能のいずれか片方のみが、SEv 生成機能に異常を通知する必要がある。

- 当該ソフトウェアのコード
- リプログラミングでのみ更新可能な、当該ソフトウェアの振舞いを制御するデータ

3.1.2. SEv 生成機能

3.1.2.1. SEv の生成

【要求事項：IDSHER_07102】

SEv 生成機能は、検知機能から異常が通知されるたびに、SEv（表 3-1）を生成し QSEv 生成機能に通知する必要がある。ここで、Event Definition ID と Context Data は、表 3-2 に従って設定される必要がある。また、Context Data は、ビッグエンディアンにて内容が設定される必要がある。

表 3-1：異常通知により生成される SEv

Field Name	Length	Description
Security Event ID	16 bit	QSEv 生成機能が QSEv に設定する Event Definition ID と Sensor Instance ID の組み合わせを一意に識別するための情報を設定する。 - Event Definition ID は、異常を検知した検知機能の要求 ID に基づいて設定される（表 3-2）。 - Sensor Instance ID は、固定値 0 である。 <補足> 本フィールドは、AUTOSAR CP では IdsMInternalEventId 型の引数として実現される。
Context Data Size	8 bit or 32 bit	Context Data のバイト長。ソフトウェアの設計者等が Event Definition ID 毎にその Context Data の長さに応じてどちらか一方を選択する。
Context Data	Variable length	検知された異常についての情報を格納するバイト列であり、異常を通知した検知機能の要求 ID に基づいて設定する。また、その異常が発生した時点でのダイアグタイムスタンプ等も設定する。

表 3-2：要求 ID ごとの Event Definition ID と Context Data

対応要求 ID	Event Definition ID	イベント概要	Context Data		
			Field Name	Length [Byte]	概要
IDSHER_04101	0x8110	遠隔車外との通信に対する 1 層目防御機能の停止	Format Version	1	Fixed value: 0x01
			Diagnostic timestamp (※1)	7	ダイアグタイムスタンプのトリップカウンタと時間カウンタ
			Diagnostic clock Information (※1)	6	ダイアグタイムスタンプの時刻情報
			Diagnostic vehicle odometer information (※1)	4	ダイアグタイムスタンプの累積走行距離情報
			Software ID Size	1	Software ID のバイト長(0~16)
			Software ID	Variable length	技術制約により取得困難な場合(※2)を除き、停止したソフトウェアを一意に識別するための情報(※3)を記録すること。 （例）当該ソフトウェアの実行ファイル名、当該ソフトウェアに紐づいた識別番号など
IDSHER_01601	0x8120	制御フローの異	Format Version	1	Fixed value: 0x01

		常	Diagnostic timestamp (※1)	7	ダイアグタイムスタンプのトリップカウンタと時間カウンタ
			Diagnostic clock information (※1)	6	ダイアグタイムスタンプの時刻情報
			Diagnostic vehicle odometer information (※1)	4	ダイアグタイムスタンプの累積走行距離情報
			Software ID Size	1	Software ID のバイト長(0~16)
			Software ID	Variable length	技術制約により取得困難な場合(※2)を除き、制御フローの異常が発生したソフトウェアを一意に識別するための情報(※3)を記録すること。 （例）当該ソフトウェアの実行ファイル名、当該ソフトウェアに紐づいた識別番号など
			Function ID Size	1	Function ID のバイト長(0~16)
			Function ID	Variable length	技術制約により取得困難な場合(※2)を除き、正規ではない実行箇所の移動が行われた関数を一意に識別するための情報(※3)を記録すること。 （例）当該関数名、当該関数に紐づいた識別番号など
IDSHER_01101 IDSHER_01102 IDSHER_01201 IDSHER_01202 IDSHER_01501 IDSHER_01502	0x8130	メモリへの不正アクセス検知	Format Version	1	Fixed value: 0x01
			Diagnostic timestamp (※1)	7	ダイアグタイムスタンプのトリップカウンタと時間カウンタ
			Diagnostic clock information (※1)	6	ダイアグタイムスタンプの時刻情報
			Diagnostic vehicle odometer information (※1)	4	ダイアグタイムスタンプの累積走行距離情報
			Software ID Size	1	Software ID のバイト長(0~16)
			Software ID	Variable length	技術制約により取得困難な場合(※2)を除き、不正アクセスを行ったソフトウェアを一意に識別するための情報(※3)を記録すること。 （例）当該ソフトウェアの実行ファイル名、当該ソフトウェアに紐づいた識別番号など
			Memory or IO ID Size	1	Memory or IO ID のバイト長(0~16)
			Memory or IO ID	Variable length	技術制約により取得困難な場合(※2)を除き、不正アクセスが行われたメモリまたは IO(ペリフェラル)一意に識別するための情報(※3)を記録すること。 （例）当該メモリのアドレス、当該 IO のデバイス名、当該 IO に紐づいた識別番号など
IDSHER_01401	0x8140	許可されていない機能の使用	Format Version	1	Fixed value: 0x01
			Diagnostic timestamp (※1)	7	ダイアグタイムスタンプのトリップカウンタと時間カウンタ

			Diagnostic clock information (※1)	6	ダイアグタイムスタンプの時刻情報
			Diagnostic vehicle odometer information (※1)	4	ダイアグタイムスタンプの累積走行距離情報
			Software ID Size	1	Software ID のバイト長(0~16)
			Software ID	Variable length	技術制約により取得困難な場合(※2)を除き、機能の不正使用を行ったソフトウェアを一意に識別するための情報(※3)を記録すること。 （例）当該ソフトウェアの実行ファイル名、当該ソフトウェアに紐づいた識別番号など
			Function ID Size	1	Function ID のバイト長(0~16)
			Function ID	Variable length	技術制約により取得困難な場合(※2)を除き、不正使用された機能を一意に識別するための情報(※3)を記録すること。 （例）当該機能名、当該機能に紐づいた識別番号など
IDSHER_02101 IDSHER_02301	0x8150	CSP/PSP/ソフトウェアの改ざん	Format Version	1	Fixed value: 0x01
			Diagnostic timestamp (※1)	7	ダイアグタイムスタンプのトリップカウンタと時間カウンタ
			Diagnostic clock information (※1)	6	ダイアグタイムスタンプの時刻情報
			Diagnostic vehicle odometer information (※1)	4	ダイアグタイムスタンプの累積走行距離情報
			CSP/PSP or Software ID Size	1	CSP/PSP or Software ID のバイト長(0~16)
			CSP/PSP or Software ID	Variable length	技術制約により取得困難な場合(※2)を除き、改ざんされた CSP/PSP またはソフトウェアを一意に識別するための情報(※3)を記録すること。 （例）当該 CSP/PSP 名、当該ソフトウェアの実行ファイル名、当該ソフトウェアに紐づいた識別番号など

(※1) 情報の詳細は参照文書[8]を参照

(※2) 取得するために、OS や BSW の改修が必要となる場合等を想定

(※3) 開発元(ECU 設計部署やサプライヤ等)が発生原因や発生箇所を特定するために有効な情報を定義する

3.1.3. QSEv 生成機能

3.1.3.1. SEv の集約

【要求事項：IDSHER_07103】

QSEv 生成機能は、参照文書[2]に定義される方式を用いて、通知される SEv を Security Event ID ご

とに集約し QSEv を生成する必要がある。Security Event ID ごとの集約の設定は【要求事項：IDSHER_03402】で定義する。

3.1.4. QSEv 送信機能

3.1.4.1. QSEv の送信

【要求事項：IDSHER_07108】

検知マスタ(参照文書[11])がいずれかの ECU に存在する場合に、本要求事項は適用される。QSEv 生成機能が QSEv を生成する度に、QSEv 送信機能は、自 ECU から検知マスタへ送信する際に用いるプロトコルに応じて下記で定義されるフレームを生成し、検知マスタへ送信する必要がある。

に示す構造のデータを、下記に示す領域へ設定して通信フレームを生成し、それを検知マスタへ送信する必要がある。

・自 ECU が、CAN 通信または CAN FD 通信を用いて、フレームを送信する場合：

参照文書[12]において QSEV_DATA_[ECU ノード名(※1)]で定義される Data Label を含むフレーム。なお、QSEV_DATA_[ノード名]には、図 3-2 で定義されるデータが格納される。

・自 ECU が、Ethernet 通信を用いて、フレームを送信する場合：

表 3-3 で定義される構造のフレーム。なお、表 3-3 の IDS Message には図 3-2 で定義されるデータが格納される。

Protocol Version	Protocol Header	IdsM Instance ID	Sensor Instance ID	Event Definition ID	Count	Reserved	Context Data (※2)
msb … lsb	msb … lsb	msb … lsb	msb … lsb	msb … lsb	msb … lsb	msb … lsb	msb … lsb

図 3-2：データ構造

表 3-3：フレームフォーマット（Ethernet の場合）

Layer	Protocol	Description	Note
L2	Ethernet	各フィールドの値は、参照文書[13]に従うこと なお、Destination MAC address は、検知マスタが搭載される CEN2 の MAC address を指定すること	
L3	IPv4	各フィールドの値は、参照文書[13]に従うこと なお、Destination IP address は、検知マスタが搭載される CEN2 の IP address を指定すること	
L4	TCP	各フィールドの値は、参照文書[13]に従うこと 但し、Destination Port Number、Source Port Number は、下記に示す値を指定すること ・Destination Port Number ：50004 (0xC354) ・Source Port Number ：50004 (0xC354)	

L5	IDS	各フィールドの値は、下記データをビックエンディアン方式で格納し、構成すること ・Message ID (4Byte) ：ALL 0 ・Length (4Byte) ：Message ID, Length, IDS Message のデータ長の和 ・IDS Message (Variable) ：図 3-2 で定義されるデータ	IDS プロトコルの詳細は、参照文書[3]参照

（※1）[ECU ノード名]は、自 ECU を示すノード名に置換すること。該当する Data Label が参照文書[12]存在しない場合は、本書の発行元部署に相談すること。
（※2）CAN FD 通信及び Ethernet 通信に適用される。CAN 通信は適用対象外。

【要求事項：IDSHER_07208】
QSEv 送信機能が、QSEv を検知マスタへ送信する場合に本要求は適用される。QSEv 送信機能による QSEv 送信がネットワークの WakeUp 要因もしくは Sleep 阻害要因となる場合、QSEv 送信機能は QSEv を送信してはならない。
<補足>
本要求事項の目的は、『QSEv の送信が、バッテリ上がりの原因となることを避けるため』である。

3.1.5. QSEv 保管機能

3.1.5.1. QSEv の保管

【要求事項：IDSHER_07109】
検知マスタがいずれの ECU にも存在しない場合に、本要求事項は適用される。QSEv 保管機能は、QSEv 生成機能が生成する最新[NumberOfQSEvs]個の QSEv を、Event Definition ID 毎に不揮発性メモリに保管する必要がある。ただし、QSEv 保管機能は、不意のリセット（バッテリ瞬断、低電圧等）時に QSEv を保管しなくてもよい。なお、QSEv 保管機能は、不揮発性メモリの書き込み回数上限を考慮し設計される必要がある。

<補足>
検知マスタがいずれかの ECU に存在する場合には、QSEv を保管するかは任意である。
不揮発性メモリの書き込み回数上限を考慮した設計の例として、IG-ON 中は RAM 領域に QSEv をバッファリングし、IG-OFF 時に不揮発性メモリに書き込む設計が挙げられる。

【要求事項：IDSHER_07111】
QSEv 保管に関する UserDefMemoryDTC および DID は表 3-4、表 3-5、表 3-6 に従う必要がある。
UserDefMemoryDTC および DID は以下の方針で定義している。
・UserDefMemoryDTC：Event Definition ID ごとに定義
・DID：QSEv 全体で一つ定義、かつ、全 Event Definition ID に対して共通で一つ定義

表 3-4：UserDefMemoryDTC 関連情報

UserDefMemoryDTC	FTB	対応 Event Definition ID	Memory Selection
U2B21	0x00	0x8110	0x14
U2B22	0x00	0x8120	0x14
U2B23	0x00	0x8130	0x14
U2B24	0x00	0x8140	0x14
U2B25	0x00	0x8150	0x14

表 3-5：QSEv 保管に関する DID

DID	Data	Length [Bit]
0xA910	Protocol Version	4
	Protocol Header	4
	IdsM Instance ID	10
	Sensor Instance ID	6
	Event Definition ID	16
	Count	16
	Reserved	8
	Context Data (※1)	Variable Length

(※1) CAN FD 通信及び Ethernet 通信に適用される。CAN 通信は適用対象外。

表 3-6：QSEv 保管データ例(Event Definition ID:0x8110 の QSEv を 5 件保管)

UserDefMemoryDTC	FTB	UserDefDTC SnapshotRecordNumber	Description
U2B21	0x00	0x01	最新[NumberOfQSEvs]個の QSEv のうち、最も古い QSEv (DID: 0xA910)
		0x02	最新[NumberOfQSEvs]個の QSEv のうち、2 番目に古い QSEv (DID: 0xA910)
		0x03	最新[NumberOfQSEvs]個の QSEv のうち、3 番目に古い QSEv (DID: 0xA910)
		0x04	最新[NumberOfQSEvs]個の QSEv のうち、4 番目に古い QSEv (DID: 0xA910)
		0x05	最新[NumberOfQSEvs]個の QSEv のうち、最も新しい QSEv (DID: 0xA910)

3.1.5.2. QSEv の読み出し

【要求事項：IDSHER_07110】

QSEv が不揮発性メモリに保管される場合に、本要求事項は適用される。不揮発性メモリに保管される QSEv は、オフボードクライアントおよびオンボードクライアントからダイアグ通信 SID 0x19 (Sub Function 0x17/0x18)で読み出しできる必要がある。ただし、前述の QSEv が一時的に揮発性メモリ上に置かれている場合、揮発性メモリ上のそれら QSEv が読み出される必要がある。

ダイアグ通信の詳細は、参照文書[9]を参照。

3.1.5.3. QSEv の消去

【要求事項：IDSHER_07204】

QSEv が不揮発性メモリに保管される場合に、本要求事項は適用される。不揮発性メモリに保管される QSEv は、オフボードクライアントからダイアグ通信 SID 0x14 (QSEv 出力用 MemorySelection 0x14)で消去できる必要がある。

ダイアグ通信の詳細は、参照文書[9]を参照。

3.2. 品質要求

本節では品質要求を定義する。

【要求事項：IDSHER_12201】

本システムおよび生成される QSEv は、遠隔車外との通信を終端する機能から改ざんされないよう、当該機能から書き込みアクセス禁止とする必要がある。

<補足>

- 実現技術例：OS の機能を用いたアクセス制御
- 本システムおよび生成される QSEv への不正な書き込みアクセスは、不揮発性メモリ/揮発性メモリ/IO(ペリフェラル)への不正アクセス検知(IDSHER_01101, IDSHER_01102, IDSHER_01201, IDSHER_01202, IDSHER_01501, IDSHER_01502)に基づいて検知が行われるため、本要求事項では検知対象としない。
- 本システムの改ざんは、ソフトウェアの改ざん検知(IDSHER_02301)に基づいて検知が行われるため、本要求事項では検知対象としない。

3.3. 制約

無し

3.4. 設計値

本節では設計値を定義する。

【要求事項：IDSHER_03401】

本節で定義する設計値は各要求で定められる条件下で設定変更可能である必要がある。

【要求事項：IDSHER_03402】

QSEv 生成・保管は表 3-7 の設計値を用いて行われる必要がある。なお、単位などの設計値に関する条件は　　表 3-8 と表 3-9 に従う必要がある。

表 3-7：QSEv 生成・保管の設計値

名称	Event Definition ID	Sensor Instance ID	設定値（※1）
IdsMEventAggregationTimeInterval	0x8110	0x0	0.3
	0x8120	0x0	0.3
	0x8130	0x0	0.3
	0x8140	0x0	0.3
	0x8150	0x0	0.3
IdsMContextDataSourceSelector	0x8110	0x0	IDSM_FILTERS_CTX_USE_FIRST
	0x8120	0x0	IDSM_FILTERS_CTX_USE_FIRST
	0x8130	0x0	IDSM_FILTERS_CTX_USE_FIRST
	0x8140	0x0	IDSM_FILTERS_CTX_USE_FIRST
	0x8150	0x0	IDSM_FILTERS_CTX_USE_FIRST
NumberOfQSEvs	0x8110	0x0	5
	0x8120	0x0	5

	0x8130	0x0	5
	0x8140	0x0	5
	0x8150	0x0	5

表 3-8：QSEv 生成設計値メタ情報

名称	単位	型	下限値	上限値
IdsMEventAggregationTimeInterval (※２)	sec	EcucFloatParamDef	0.05	10.00
IdsMContextDataSourceSelector	-	EcucEnumerationParamDef	IDSM_FILTERS_CTX_USE_FIRST	IDSM_FILTERS_CTX_USE_LAST

※１：IdsMEventAggregationTimeInterval および IdsMContextDataSourceSelector の設定値がハイフン「-」であるのは集約を行わないことを意味する。

※２：設定値列に記載の値と同じ値を設定できない場合、記載の設定値より小さく、かつ、設定可能な設計値のうち、最大の値が設定される必要がある。

表 3-9：QSEv 保管設計値メタ情報

名称	説明	単位	下限値	上限値
NumberOfQSEvs	QSEv の保管件数	-	0	10

Appendix. A. 要求事項の監視対象や実現手段の例示

本章では、検知機能の要求に対して監視対象、監視すべき異常、実現手段の例示を行う。

A.1. 遠隔車外との通信に対する 1 層目防御機能の停止の検知【IDSHER_04101】

A.1.1. 本要求の監視対象

監視対象は常駐ソフトウェア(常駐プロセス)として設計される、遠隔車外との通信に対する 1 層目防御機能である。

一例として、遠隔車外との通信に対する 1 層目防御機能が常駐ソフトウェア(常駐プロセス)として設計される場合とは、TLS が『ユーザ空間に常駐し、遠隔車外との TLS 通信を集約するプロキシ機能を提供するソフトウェア』として設計される場合が該当する。

A.1.2. 監視すべき異常

A.1.1 で述べた常駐プロセスにおいて、監視すべき異常は、当該プロセスが動作していないことである。

一例として、当該プロセスが最初から動作していない場合や、最初は動作していたが途中から動作が停止した場合、検知機能は異常として検知する必要がある。なお、ソフトウェアのコードや設定を不正に変更され、本来意図する動作ができなくなる事象は、ソフトウェアの改ざん検知(IDSHER_02301)に基づいて検知が行われるため、本要求事項では検知対象としない。

A.1.3. 本要求の実現例

本要求の実現例として、プロセスの生存監視が挙げられる。本要求を満たす手法として、下記の方式が挙げられる。

- 監視者が、本要求の対象となるプロセスが動作していることを定期的に確認する。
- 本要求の対象となるプロセスが、動作していることを定期的に監視者に伝える。

A.2. 制御フローの異常検知【IDSHER_01601】

A.2.1. 本要求の監視対象

監視対象は遠隔車外との通信を終端する機能を構成するソフトウェアである。

一例として、TLS 通信を終端するソフトウェアについては、TLS の機能を実現する共有ライブラリ(TLS ライブラリ)や、TLS ライブラリをロードする実行形式ファイル(Executable file)に A.2.3 の検知技術を使用する。

A.2.2. 監視すべき異常

監視すべき異常は、正規の制御フローとして起こりえない関数遷移である。その例を下記に示す。

- 関数ポインタの改ざんに起因して発生する間接コール
  図 3-3 に示されるプログラムは関数 Func1 から関数ポインタ用いて関数 Func2 をコールする。ここで、において、前記関数ポインタの改ざんによって、関数の先頭以外の箇所に遷移が行われる場

合、検知する必要がある。

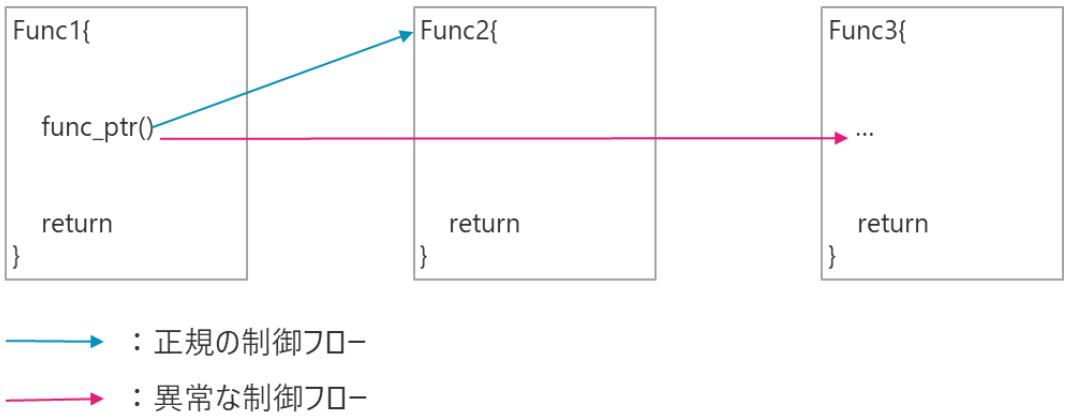

図 3-3：間接コールによる遷移における異常

・ スタック上のリターンアドレスの改ざんに起因して発生するリターン

図 3-4 に示されるプログラムは関数 Func1 から関数 Func2 を呼び出し、Func2 の処理を終えた後 Func1 へとリターンする。ここで、図 3-4 において、スタック上のリターンアドレスの改ざんによって、関数呼出し元以外の箇所に遷移が行われる場合、検知する必要がある。

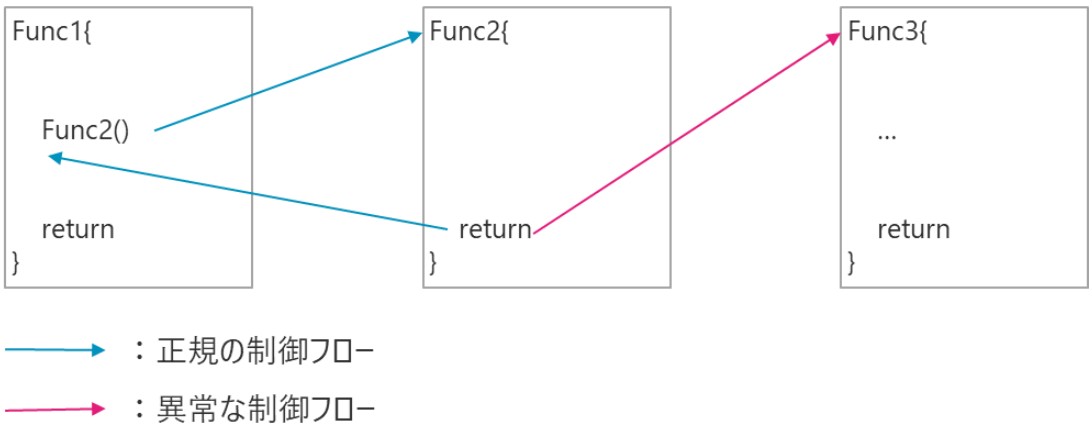

図 3-4：リターンによる遷移における異常

なお、コードの改ざんについては、ソフトウェアの改ざん検知(IDSHER_02301)に基づいて検知が行われるため、本要求事項では検知対象としない。

A.2.3. 本要求の実現例

本要求の実現例として、Control Flow Integrity (CFI)やシャドウスタックが挙げられる。

・ CFI

一例として、間接コールの遷移先となるアドレスの検証を行うコードをプログラムに挿入し、遷移

の際に検証することで、正規の制御フローとして起こり得ない遷移を検知する。

・ シャドウスタック
一例として、関数コールの際、通常のスタックに加え、別の専用の領域にもリターンアドレスを格納し、関数のリターンの際に両者が同一であるかを検証することで、正規の制御フローとして起こり得ない遷移を検知する。

A.3. パスを用いた不正アクセスの検知【IDSHER_01101, IDSHER_01201, IDSHER_01501】

A.3.1. 本要求の監視対象

監視対象は下記の条件を全て満たすソフトウェアである。

・ 車外からの通信を終端する機能を構成するソフトウェアであること
・ ファイルシステムを利用しパスを用いてリソースへアクセスできる仕組みを持つ
・ 不揮発性メモリ、揮発性メモリまたはIO(ペリフェラル)に対してアクセス可能なこと

ここでパスとはリソースの所在を表す識別子の 1 つである。一例として、データの所在を示す/path/to/data といった文字列がパスに該当する。

本要求のIO(ペリフェラル)の例として、CAN コントローラ、Ethernet コントローラ、USB コントローラ、Wi-Fi モジュール、Bluetooth モジュール、HSM などが挙げられる。

A.3.2. 監視すべき異常

監視すべき異常は前記ソフトウェアによる、許可されていないリソースへのアクセスである。当該ソフトウェアがープロセスとして動作している場合について、パスを用いた不正アクセスの一例を示す。

・ パスを用いた不正アクセス
当該ソフトウェアは、EP 領域の不揮発性メモリにおいて、図 3-5 に示す通り、/foo/bar/以下のディレクトリやファイルに対して書き込みアクセスを許可されているが、/foo/baz/以下のディレクトリやファイルへの書き込みアクセスを許可されていないとする。このとき、当該ソフトウェアが書き込みアクセスを許可されていない/foo/baz/file_6 への書き込みを行った、または試みた場合、異常として検知する必要がある。一例として不揮発性メモリへの不正アクセスの例を述べたが、揮発性メモリやIO(ペリフェラル)への不正アクセスにおいても、同様に検知する必要がある。

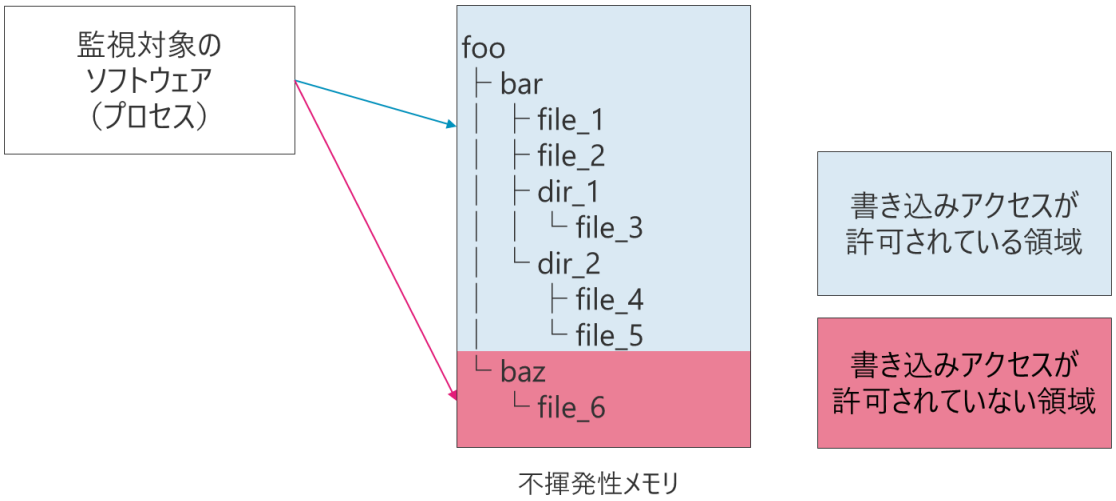

図 3-5：パスを用いた不正アクセス

なお、本要求事項の前提として、当該ソフトウェアの動作に不必要なアクセスは許可されない設計となっている必要がある

A.3.3. 本要求の実現例

本要求の実現例として、ファイルアクセスの監視が挙げられる。一例として、Linux では、ファイルのそれぞれに読み出しアクセス、書き込みアクセス、実行アクセスの権限を設定できる。許可されていない操作を試みた場合、権限違反として実行されず検知される。

A.4. アドレスを用いた不正アクセスの検知【IDSHER_01102, IDSHER_01202, IDSHER_01502】

A.4.1. 本要求の監視対象

監視対象は下記の条件を全て満たすソフトウェアである。

- 車外からの通信を終端する機能を構成するソフトウェアであること
- アドレスを用いてリソースへアクセスできる仕組みを持つこと
- 不揮発性メモリ、揮発性メモリまたは IO(ペリフェラル)に対してアクセス可能なこと

ここでアドレスとはリソースの所在を表す識別子の 1 つである。一例として、デバイス上の番地を表す 0x1000 といった数値がアドレスに該当する。

本要求の IO(ペリフェラル)の例として、CAN コントローラ、Ethernet コントローラ、USB コントローラ、Wi-Fi モジュール、Bluetooth モジュール、HSM などが挙げられる。

A.4.2. 監視すべき異常

監視すべき異常は前記ソフトウェアによる、許可されていないリソースへのアクセスである。当該ソフ

トウェアが一プロセスとして動作している場合について、アドレスを用いた不正アクセスの一例を示す。

・ アドレスを用いた不正アクセス
当該ソフトウェアは、EP 領域の揮発性メモリにおいて、に示す通り、上記ソフトウェアによる書き込みアクセスが許可されている領域と許可されていない領域を持つとする。このとき、当該ソフトウェアが書き込みアクセスを許可されていない領域内の 0x1000 に対して書き込みアクセスを行った、または試みた場合、異常として検知する必要がある。一例として揮発性メモリへの不正アクセスの例を述べたが、不揮発性メモリや IO(ペリフェラル)への不正アクセスにおいても、同様に検知する必要がある。

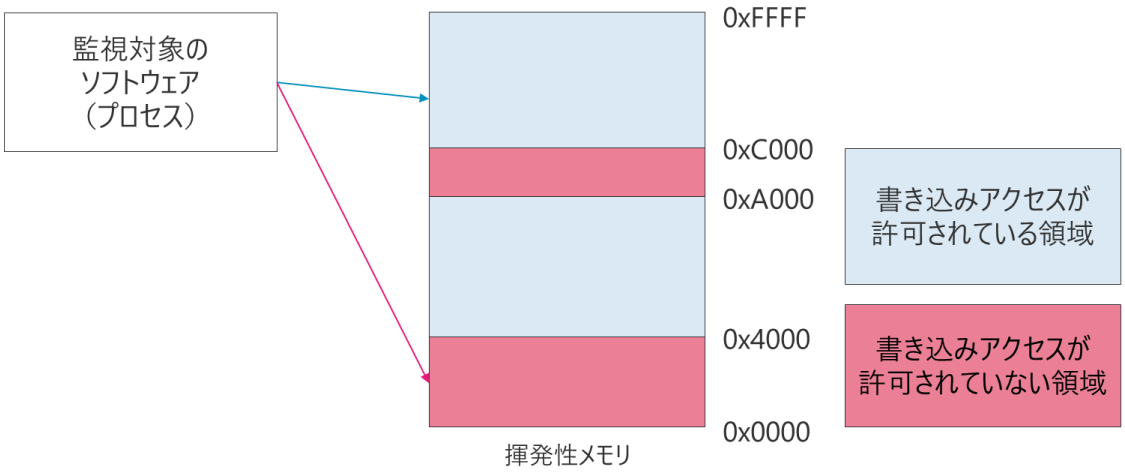

図 3-6：アドレスを用いた不正アクセス

なお、本要求事項の前提として、当該ソフトウェアの動作に不必要なアクセスは許可されない設計となっている必要がある

A.4.3. 本要求の実現例

本要求の実現例として、メモリアクセスの監視が挙げられる。一例として、Linux では MMU (Memory Management Unit)を用いてメモリ管理を行う。許可されていないメモリアクセスが発生した場合、セグメンテーション違反として検知される。

A.5. 機能の不正使用の検知【IDSHER_01401】

A.5.1. 本要求の監視対象

監視対象は車外からの通信を終端する機能を構成するソフトウェアである。

A.5.2. 監視すべき異常

一例として、使用に際して何らかの権限を必要とする機能とは、システムコールが該当する。
監視対象となるソフトウェアが使用を許可されていないシステムコールを使用するまたは使用を試み

た場合、異常として検知される必要がある。なお、本要求事項の前提として、当該ソフトウェアの動作に不必要な機能の使用は許可されない設計となっている必要がある。

A.5.3. 本要求の実現例

本要求の実現例として、システムコールの使用監視が挙げられる。システムコールの使用監視では、監視対象となるソフトウェアごとに、当該ソフトウェアが利用しうるシステムコールを事前にリストとして定義する。当該ソフトウェアがシステムコールを発行するたび、許可リスト方式で検証を行い、検証に失敗した場合、そのシステムコールの実行を防ぐ。

A.6. CSP/PSP の改ざん検知【IDSHER_02101】

A.6.1. 本要求の監視対象

監視対象はエントリーポイント領域が不揮発性メモリに持つ CSP/PSP である。
ここで、CSP の例として、秘密鍵、鍵生成に使用する乱数シード、パスワードが挙げられる。また、PSP の例として、公開鍵、公開鍵証明書、自己署名証明書が挙げられる。なお、CSP/PSP の定義は参照文書[6]を参照する必要がある。

A.6.2. 監視すべき異常

監視すべき異常は、CSP/PSP に該当するデータの使用時に当該データが改ざんされている事象である。
ここで、当該データの使用時とは、当該データを使用する任意の操作を指す。たとえば、暗号化処理、復号処理、認証処理や、前記処理を行うために、不揮発性メモリ上の当該データを揮発性メモリや HSM に展開する処理はすべて当該データの使用に該当する。

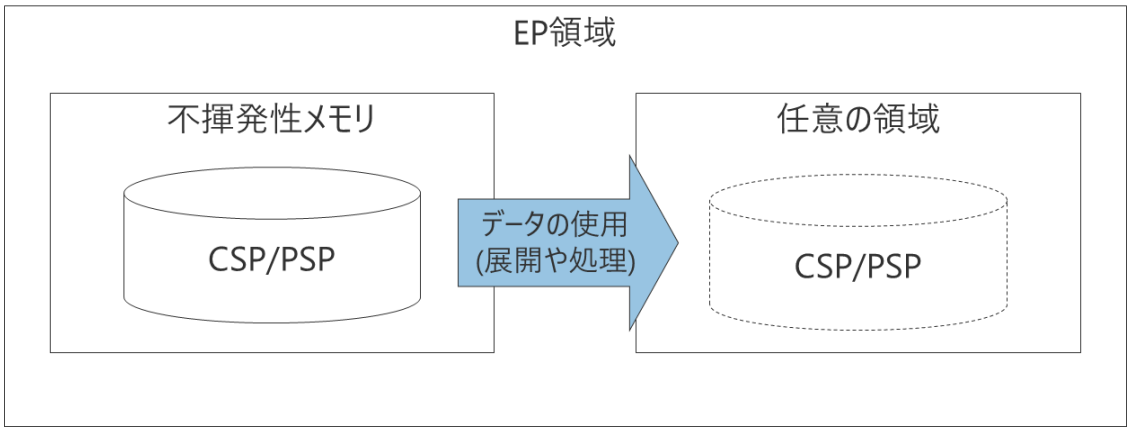

図 3-7：CSP/PSP の改ざん検知

A.6.3. 本要求の実現例

本要求の実現例として、ファイルの改ざん検知が挙げられる。ファイルの改ざん検知では、監視対象とするデータの MAC を事前に生成しておき、当該データの使用時に MAC の検証を行い、事前に生成した値と照合し、異なる場合、異常として検知される。

ここで、上記 MAC の生成に用いる鍵は、HSM に保管するなど、適切に管理される必要がある。

A.7. ソフトウェアの改ざん検知【IDSHER_02301】

A.7.1. 本要求の監視対象

監視対象は、エントリーポイント領域に配置される機能を構成するソフトウェアのコードである。
また、リプログラミングでのみ更新可能な、当該ソフトウェアの振舞いを制御するデータも、監視対象に該当する。一例として、当該ソフトウェアの振舞いを制御するデータとは、当該ソフトウェアのコンフィグデータ(設定ファイル)を指す。

A.7.2. 監視すべき異常

監視すべき異常は、当該ソフトウェアの起動時に監視対象が改ざんされていることである。
一例として、リプログラミングによってのみ、ソフトウェアを追加・更新できる ECU/VM においては、当該 ECU/VM の起動時にソフトウェアが改ざんされているとき、検知機能は異常を通知する必要がある。

なお、その他の手段によってソフトウェアを追加・更新できる ECU/VM においては、その手段に応じて、具体的な監視すべき異常とその異常を検知する手段について検討する必要がある。

A.7.3. 本要求の実現例

本要求の実現例として、セキュアブートが挙げられる。

- セキュアブート
  ECU/VM の起動時に監視対象となるソフトウェアの完全性の検証を行うことで、改ざんされたソフトウェアが動作することを防ぐ。

Revision history

Version	Change	Date	Reviser
a00-00-a	First version issued	2021/04/05	46F 4G Inagaki
a00-00-b	Translation into English added	2021/05/14	46F 4G Inagaki
a00-01-a	Requirements fleshed, readability improved	2021/08/06	46F 4G Takeyama
a00-02-a	Heartbeat SEv creation deleted	2021/12/03	46F 4G Takeyama
a00-03-a	- References added - T.B.D. deleted - Requirements about QSEv storing modified - IDSHER_07208 added - Detection method example of IDSHER_02101 modified - IDSHER_04101 modified - IDSHER_02301 modified - IDSHER_07102 modified - IDSHER_07108 modified	2022/02/03	46F 4G Takeyama
a00-04-a	- Hardware-related requirement added in List of requirements - IDSHER_02301 requirement modified - IDSHER_04101 requirement modified - IDSHER_07102 description of Context Data clarified - IDSHER_07108 requirement modified - IDSHER_07109 QSEv storing requirement modified - IDSHER_07111 UserDefineDTC and DID requirement added - IDSHER_07110 SID for QSEv read clarified - IDSHER_07202 deleted - - IDSHER_07204 SID for QSEv deletion clarified	2022/04/29	46F 4G Takeyama
a00-04-b	- IDSHER_07111 UserDefMemoryDTC value modified - IDSHER_07110 diagnostic specification reference added - IDSHER_07204 diagnostic specification reference added	2022/05/20	46F 4G Takeyama
a00-04-c	- Table 2-2 Editorial errors corrected - IDSHER_12201 Editorial errors corrected - IDSHER_07109 The part of the note moved to requirement	2022/06/09	46F 4G Takeyama

a00-05-a	- Explanation of detection function in 3.1.1 added - IDSHER_04101 requirement modified for clarification - IDSHER_01601 requirement modified for clarification - IDSHER_01101 allocation condition added - IDSHER_01102 allocation condition added - IDSHER_01201 allocation condition added - IDSHER_01202 allocation condition added - IDSHER_01501 allocation condition added IDSHER_01502 allocation condition added - IDSHER_01401 allocation condition added and requirement modified for clarification - IDSHER_02301 requirement modified for clarification - IDSHER_12201 requirement clarified - Supplementary information of requirements in 3.1.1 added as appendix.	2022/11/25	46F 4G Takeyama

Table of contents

1. Introduction

1.1. Purpose of this document

The goal of Host-based IDS for Entry Point (hereinafter referred to as *this system*) is to detect and log intrusion into an entry point and the attempt. Log recorded by this system is used to realize the *detection* function in the framework for cybersecurity (Reference [1]) defined by National Institute of Standards and Technology (hereinafter referred to as *NIST*). The purpose of this document is to define the requirements of this system.

1.2. Target

This document is allocated to entry-point ECUs/VMs specified by another document.

1.3. Prerequisites

None

1.4. Description of requirements

We describe requirements as [Requirement: **] in this document where <Note> means just a supplementary note.

1.5. Input documents

Inputs documents, and references are shown in this section. If the specification of the ECU specifies the version of the reference, follow it.

1.5.1. Input documents

Table 1-1: Input documents

No.	Document name	Ver.
1	Vehicle Cyber Security Concept Definition Document	Latest

1.5.2. References

Table 1-2: References

No.	Document name	Ver.

1	Framework for Improving Critical Infrastructure Cybersecurity Version 1.1, National Institute of Standards and Technology, April 16, 2018, https://www.nist.gov/publications/framework-improving-critical-infrastructure-cybersecurity-version-11	1.1
2	Requirements specification of QSEv creation	-
3	AUTOSAR_PRS_IntrusionDetectionSystem	R20-11
4	AUTOSAR_SWS_IntrusionDetectionSystemManager	R20-11
5	AUTOSAR_SWS_AdaptiveIntrusionDetectionSystemManager	R20-11
6	Terms and Definitions related to Vehicle Cybersecurity and Privacy	-
7	Deleted	-
8	Time Stamp requirement specification	-
9	TOYOTA Phase6 Diagnostics Communication and Reprogramming standard specifications	-
10	Requirements Specification of Host-based IDS for Multi-layered Separation Function	-
11	Requirements Specification of Intrusion Detection Master	-
12	PF LAN Specification	
13	Automotive Ethernet communication function specification	

1.6. Glossary

We define terms used in this document. Since some parts of this system are expected to be implemented in accordance with AUTOSAR requirements, we use terms defined by AUTOSAR. See [3], [4] and [5] for the terms. See [6] for the other terms.

Table 1-3: Glossary

Term	Meaning
-	-

2. Requirement overview

2.1. System context

We show the system context with DFD (Figure 2-1). The circle means this system, and the rectangles mean subjects transmitting information or services.

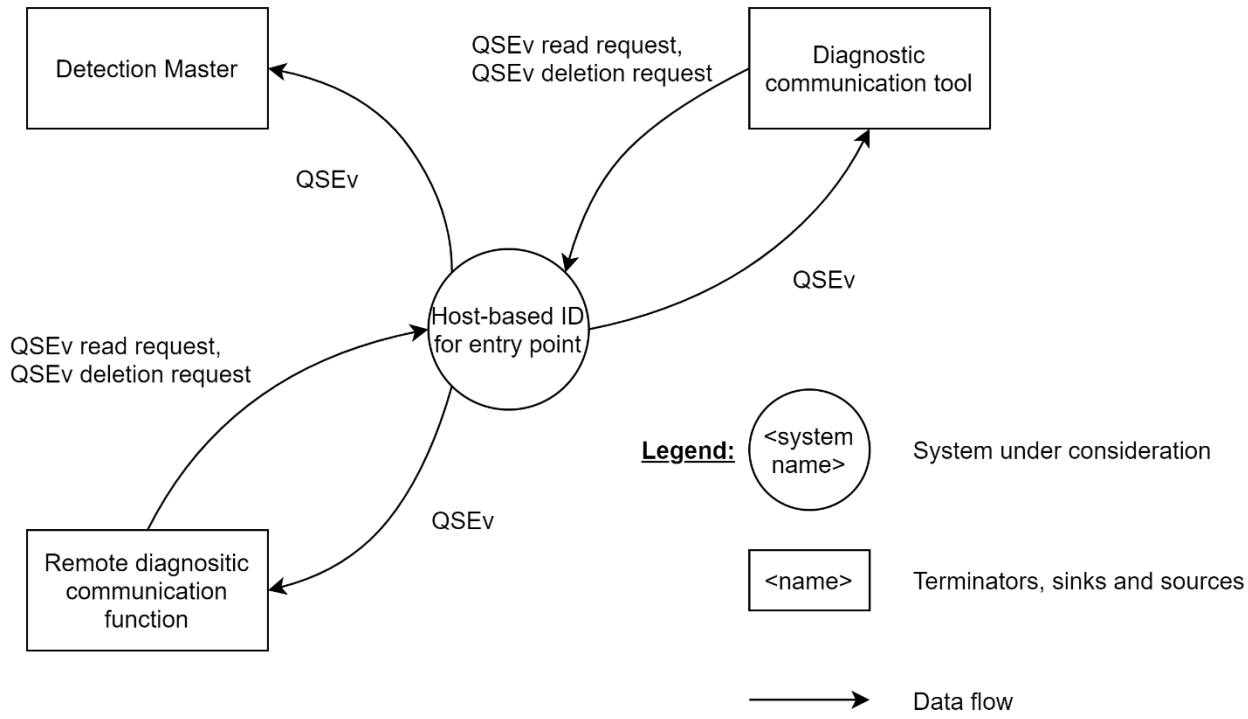

Figure 2-1: System context

2.2. System operation overview

This system operates as the UML activity diagram (Figure 2-1) when one of these events shown in (Table 2-1) happens.

Table 2-1: Events to start the operation

Event No.	Event that can be the starting point of the operation
①	Occurrence of intrusion on ECUs/VMs where this system is implemented.
②	Request to read QSEvs stored by this system
③	Request to delete QSEvs stored by this system

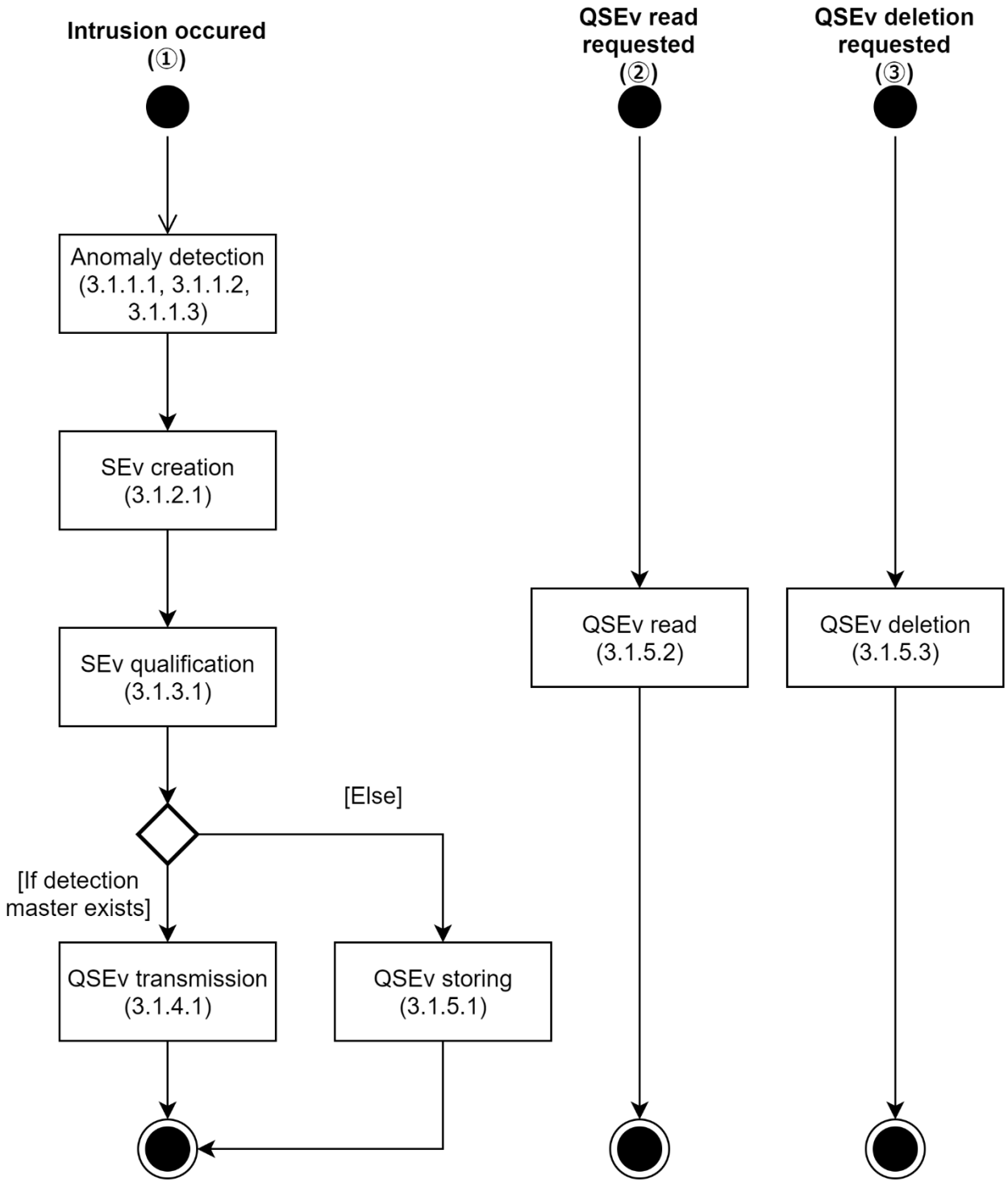

Figure 2-2: System operation

2.3. List of requirements

We show the list of all requirements defined in this document (Table 2-2).

Table 2-2: List of requirements

Category			Requirement ID	Hardware-related requirement
Functional requirements	Detection function	Detection of abort of a first layer protection function for communication from Out-Car	IDSHER_04101	No
		Detection of illegal operation of a function to terminate communication from Out-Car	IDSHER_01601	No
			IDSHER_01101	No
			IDSHER_01102	No
			IDSHER_01201	No
			IDSHER_01202	No
			IDSHER_01501	No
			IDSHER_01502	No
			IDSHER_01401	No
		Detection of manipulation of CSP/PSP or software in an entry point region	IDSHER_02101	No
			IDSHER_02301	No
	SEv creation function	SEv creation	IDSHER_07102	No
	QSEv creation function	SEv qualification	IDSHER_07103	No
	QSEv transmission function	QSEv transmission	IDSHER_07108	No
			IDSHER_07208	No
	QSEv storing function	QSEv storing	IDSHER_07109	No
			IDSHER_07111	No
		QSEv read	IDSHER_07110	No
		QSEv deletion	IDSHER_07204	No
Quality requirements			IDSHER_12201	No
Parameters			IDSHER_03401	No
			IDSHER_03402	No

3. System requirements

We define system requirements in this chapter.

3.1. Functional requirements

We define functional requirements in this section.

3.1.1. Detection function

There are mainly three kinds of detection.

1. *Detection of abort of a first layer protection function for communication from Out-Car* (No. 1 of Figure 3-1).
2. *Detection of illegal operation of a function to terminate communication from Out-Car* (No. 2 of Figure 3-1).
3. The third one is *detection of manipulation of CSP/PSP or Software in an entry point region* (No. 3 of Figure 3-1).

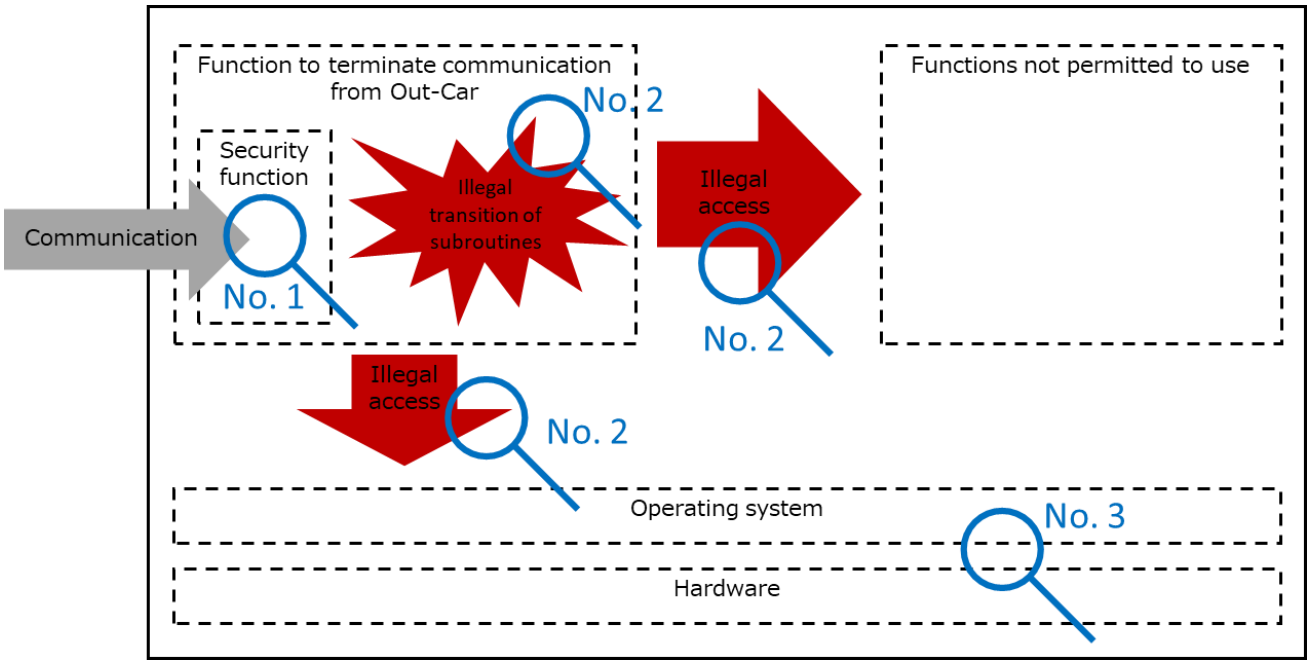

Figure 3-1: Entry-point ECU/VM model to terminate communication from Out-Car

Note that “a function to terminate communication from Out-Car” is software which directly communicate with a remote entity outside the car. Furthermore the function has a first layer protection function for communication from Out-Car.

As an example, TLS is “a first layer protection function for communication from Out-Car”,

software which terminates TLS communication is “a function to terminate communication from Out-Car”. Furthermore, as an example, “software which terminates TLS communication” is “process of an application comprised of shared libraries which realize TLS function (TLS libraries) and executable files which load TLS libraries”.

Note that when “other software” transmits and receives information from Out-Car indirectly through “software which terminates TLS communication”, if the processes are different, “other software” is not “a function to terminate communication from Out-Car”

In addition, this document has Appendix. A which shows examples of monitored target, monitored anomaly, and way to meet the requirement for eace requirement at the end of this document.

3.1.1.1. Detection of abort of a first layer protection function for communication from Out-Car

[Requirement: IDSHER_04101]

If a first layer protection function for communication from Out-Car is designed as residen software (resident process), this requirement shall be allocated. If such software does not work(*1) in a situation when the software should work, a detection function shall notify a SEv creation function of the anomaly. However, if IDSHMR_04101 in [10] is allocated, both a detection function of this system and a detection function defined in [10] shall not notify a SEv creation function of the anomaly for the same event. In other words, either the detection function of this system or the detection function defined in [10] shall notify a SEv creation function of the anomaly.

*1: “Software does not work” is a status that software is not running, is not a status that security functions of the software is invalid because of manipulation of the code or configuration files.

3.1.1.2. Detection of illegal operation of function to terminate communication from Out-Car

3.1.1.2.1. Detection of abnormal control flow

[Requirement: IDSHER_01601]

If a *transition between functions that shall not occur in an authenticated flow* (*1) occurs when software composing a function to terminate communication from Out-Car is running, or is attempted, a detection function shall notify a SEv creation function the anomaly.

*1: *transition*s *between functions that shall not occur in an authenticated flow* is indirect call (function call with a function pointer) and return by manipulation of a function pointer or return address on a stack. *transition*s *between functions that shall not occur in an authenticated flow* do not include events caused by manipulation of code.

3.1.1.2.2. Detection of illegal access to non-volatile memory

[Requirement: IDSHER_01101]

If software composing a function to terminate communication from Out-Car has architecture so that it can access to *non-volatile memory* (*1) in an entry point region by a path of a file system and the software is designed so that unnecessary access is not permitted, this requirement shall be allocated. If the software performs any operation shown below by a path to *non-volatile memory* in the entry point region where the operation is not permitted to be performed, or attempts to do so, a detection function shall notify a SEv creation function of the anomaly.

- Read access
- Write access
- Execute access
- Change *attribute* (*2)

*1: *non-volatile memory* means physical or logical devices that can keep data non-volatilely, regardless of implementation such as memory embedded in MCU/SoC or discrete memory. Note HSM is not non-volatile memory but IO (peripheral).

*2: *Attribute* means whether read/write/execute access is permitted or not.

[Requirement: IDSHER_01102]

If software composing a function to terminate communication from Out-Car has architecture so that it can access to *non-volatile memory* (*1) in a region of entry point by an address and the software is designed so that unnecessary access is not permitted, this requirement shall be allocated. If the software performs any operation shown below by a path to *non-volatile memory* in the entry point region where the operation is not permitted to be performed, or attempts to do so, a detection function shall notify a SEv creation function of the anomaly.

- Read access
- Write access
- Execute access
- Change *attribute* (*2)

*1: *non-volatile memory* means physical or logical devices that can keep data non-volatilely, regardless of implementation such as memory embedded in MCU/SoC or discrete memory. Note HSM is not non-volatile memory but IO (peripheral).

*2: *attribute* means whether read/write/execute access is permitted or not.

3.1.1.2.3. Detection of illegal access to volatile memory

[Requirement: IDSHER_01201]

If software composing a function to terminate communication from Out-Car has architecture so that it can access to *volatile memory* (*1) in an entry point region by a path of a file system and the

software is designed so that unnecessary access is not permitted, this requirement shall be allocated. If the software performs any operation shown below by a path to *volatile memory* in the entry point region where the operation is not permitted to be performed, or attempts to do so, a detection function shall notify a SEv creation function of the anomaly.

- Read access
- Write access
- Execute access
- Change *attribute* (*2)

*1: *volatile memory* means physical or logical devices that can keep data volatilely, regardless of implementation.
*2: *attribute* means whether read/write/execute access is permitted or not.

[Requirement: IDSHER_01202]
If software composing a function to terminate communication from Out-Car has architecture so that it can access to *volatile memory* (*1) in an entry point region by an address and the software is designed so that unnecessary access is not permitted, this requirement shall be allocated. If the software performs any operation shown below by a path to *volatile memory* in the entry point region where the operation is not permitted to be performed, or attempts to do so, a detection function shall notify a SEv creation function of the anomaly.

- Read access
- Write access
- Execute access
- Change *attribute* (*2)

*1: *volatile memory* means physical or logical devices that can keep data volatilely, regardless of implementation.
*2: *attribute* means whether read/write/execute access is permitted or not.

3.1.1.2.4. Detection of illegal access to IO (peripheral)

[Requirement: IDSHER_01501]
If software composing a function to terminate communication from Out-Car has architecture so that it can access to *IO (peripheral)* (*1) in an entry point region by a path of a file system and the software is designed so that unnecessary access is not permitted, this requirement shall be allocated. If the software performs any operation shown below by a path to *IO (peripheral)* memory in the entry point region where the operation is not permitted to be performed, or attempts to do so, a detection function shall notify a SEv creation function of the anomaly.

- Read access
- Write access

- Change *attribute* (*2)

*1: *IO (peripheral)* means physical or logical devices that can input/out data, regardless of implementation.

*2: *attribute* means whether read/write access is permitted or not.

[Requirement: IDSHER_01502]

If software composing a function to terminate communication from Out-Car has architecture so that it can access to *IO (peripheral)* (*1) in an entry point region by an address and the software is designed so that unnecessary access is not permitted, this requirement shall be allocated. If the software performs any operation shown below by a path to *IO (peripheral)* in the entry point region where the operation is not permitted to be performed, or attempts to do so, a detection function shall notify a SEv creation function of the anomaly.

- Read access
- Write access
- Change *attribute* (*2)

*1: *IO (peripheral)* means physical or logical devices that can input/out data, regardless of implementation.

*2: *attribute* means whether read/write access is permitted or not.

3.1.1.2.5. Detection of illegal usage of function

[Requirement: IDSHER_01401]

If an entry point region has a function that requires an authority to use and the software composing functions to terminate communication from Out-Car is designed so that use of unnecessary function is not permitted, this requirement shall be allocated. When the software uses a function unpermitted to use, or attempts to do so, a detection function shall notify a SEv creation function of the anomaly.

3.1.1.3. Detection of manipulation of CSP/PSP or software in entry point region

3.1.1.3.1. Detection of manipulation of CSP/PSP

[Requirement: IDSHER_02101]

If an entry point region has data fallen into CSP/PSP in *non-volatile memory* (*1), this requirement shall be allocated. If the data is manipulated *at the usage of it* (*2), a detection function shall notify a SEv creation function of the anomaly.

*1: *non-volatile memory* means physical or logical devices that can keep data non-volatilely, regardless of implementation such as memory embedded in MCU/SoC or discrete memory. Note HSM is not non-volatile memory but IO (peripheral).

*2: *at the usage of it* includes moment when the data stored in non-volatile, fallen into CSP/PSP, are allocated to volatile or HSM.

3.1.1.3.2. Detection of manipulation of software

[Requirement: IDSHER_02301]

If any of the following is manipulated at the startup of each software composing a function in an entry-point region, a detection function shall notify a SEv creation function of the anomaly. In addition, the detection of manipulation shall be performed from a region where integrity is guaranteed. However, if IDSHMR_01601 in [10] is allocated, both a detection function of this system and a detection function defined in [10] shall not notify a SEv creation function of the anomaly for the same event. In other words, either the detection function of this system or the detection function defined in [10] shall notify a SEv creation function of the anomaly.

- The code of the software
- The data controlling the behavior of the software which can be updeted only by reprogramming.

3.1.2. SEv creation function

3.1.2.1. SEv creation

[Requirement: IDSHER_07102]

When a SEv creation function is notified of an anomaly by a detection function, it shall create an SEv (Table 3-1), and notify a QSEv creation function of the SEv. Event Definition ID and Context Data shall be set in accordance Table 3-2. Context Data shall be set with big endian.

Table 3-1: Anomaly notification SEv

Field Name	Length	Description
Security Event ID	16 bit	This field shall be set to an identifier that identifies Event Definition ID and Sensor Instance ID that a QSEv creation function sets a QSEv to. - Event Definition ID shall be in accordance with an anomaly detected (Table 3-2). - Sensor Instance ID shall be fixed to 0. <Note> This field is implemented by an IdsMInternalEventId type parameter.
Context Data Size	8 or 32 bit	This field shall be set to a byte length of Context Data. One of them shall be chosen for each Event Definition ID in software design phase up to size of Context Data.
Context Data	Variable length	This field shall be set to a sequence of bytes about an anomaly detected, and shall be set depending on a requirement ID of a detection function that has notified an anomaly. Diagnostic timestamp of occurrence of anomaly shall be also set.

Table 3-2: Event Definition ID, Count, and Context Data for each requirement ID

Corresponding Requirement ID	Event Definition ID	Event sumary	Context Data		
			Field Name	Length [Byte]	Description
IDSHER_04101	0x8110	Detection of abort of a first layer protection function for communication from Out-Car	Format Version	1	Fixed vale: 0x01
			Diagnostic timestamp (*1)	7	Trip counter and time counter of diagnostic timestamp
			Diagnostic clock Information (*1)	6	Clock information of diagnostic timestamp
			Diagnostic vehicle odometer information (*1)	4	Odometer of diagnostic timestamp
			Software ID Size	1	This field shall be set to a byte length of Software ID (0~16).

			Software ID	Variable length	This field shall be set to the information (*3) to identify which software has been aborted unless it is difficult to obtain the information due to technical constraint (*2). (For example, the executable file name of the software, identifier related to the software, etc.)
IDSHER_01601	0x8120	Detection of abnormal control flow	Format Version	1	Fixed value: 0x01
			Diagnostic timestamp (*1)	7	Trip counter and time counter of diagnostic timestamp
			Diagnostic clock information (*1)	6	Clock information of diagnostic timestamp
			Diagnostic vehicle odometer information (*1)	4	Odometer of diagnostic timestamp
			Software ID Size	1	This field shall be set to a byte length of Software ID (0~16).
			Software ID	Variable length	This field shall be set to the information (*3) to identify which software an abnormal control flow occurred in unless it is difficult to obtain the information due to technical constraint (*2). (For example, the executable file name of the software, identifier related to the software, etc.)
			Function ID Size	1	This field shall be set to a byte length of Function ID (0~16).
			Function ID	Variable length	This field shall be set to the information (*3) to identify which function an abnormal transition occurred from unless it is difficult to obtain the information due to technical constraint (*2). (For example, the function name, identifier related to the function, etc.)
IDSHER_01101 IDSHER_01102 IDSHER_01201 IDSHER_01202 IDSHER_01501 IDSHER_01502	0x8130	Detection of illegal access to memory	Format Version	1	Fixed vale: 0x01
			Diagnostic timestamp (*1)	7	Trip counter and time counter of diagnostic timestamp
			Diagnostic clock information (*1)	6	Clock information of diagnostic timestamp
			Diagnostic vehicle odometer information (*1)	4	Odometer of diagnostic timestamp
			Software ID Size	1	This field shall be set to a byte length of Software ID (0~16).

			Software ID	Variable length	This field shall be set to the information (*3) to identify which software performed illegal access unless it is difficult to obtain the information due to technical constraint (*2). (For example, the executable file name of the software, identifier related to the software, etc.)
			Memory or IO ID Size	1	This field shall be set to a byte length of Memory or IO ID (0~16).
			Memory ID or IO ID	Variable length	This field shall be set to the information (*3) to identify which memory or IO (peripheral) was illegally accessed unless it is difficult to obtain the information due to technical constraint (*2). (For example, the memory address, the device name of the IO, identifier related to the IO, etc.)
IDSHER_01401	0x8140	Detection of illegal usage of function	Format Version	1	Fixed value: 0x01
			Diagnostic timestamp (*1)	7	Trip counter and time counter of diagnostic timestamp
			Diagnostic clock information (*1)	6	Clock information of diagnostic timestamp
			Diagnostic vehicle odometer information (*1)	4	Odometer of diagnostic timestamp
			Software ID Size	1	This field shall be set to a byte length of Software ID (0~16).
			Software ID	Variable length	This field shall be set to the information (*3) to identify which software used a function not permitted to do so unless it is difficult to obtain the information due to technical constraint (*2). (For example, the executable file name of the software, identifier related to the software, etc.)
			Function ID Size	1	This field shall be set to a byte length of Function ID (0~16).
			Function ID	Variable length	This field shall be set to the information (*3) to identify which function was used illegally unless it is difficult to obtain the information due to technical constraint (*2). (For example, the function name identifier related to the function, etc.)
IDSHER_02101 IDSHER_02301	0x8150	Detection of maninpulation of CSP/PSP or software	Format Version	1	Fixed value: 0x01
			Diagnostic timestamp (*1)	7	Trip counter and time counter of diagnostic tim1estamp
			Diagnostic clock information (*1)	6	Clock information of diagnostic timestamp

			Diagnostic vehicle odometer information (*1)	4	Odometer of diagnostic timestamp
			CSP/PSP or Software ID Size	1	This field shall be set to a byte length of CSP/PSP or Software ID (0~16).
			CSP/PSP ID or Software ID	Variable length	This field shall be set to the information (*3) to identify which CSP/PSP or software was manipulated unless it is difficult to obtain the information due to technical constraint (*2). (For example, the CSP/PSP name, the executable file name of the software, identifier related to the software, etc.)

*1: For details, see reference [8].

*2: For example, a case when OS or BSW is necessary to be modified to obtain the information.

*3: Developer (ECU software designer, supplier, etc.) defines data effective to identify the cause and the region where an event occurs.

3.1.3. QSEv creation function

3.1.3.1. SEv qualification

[Requirement: IDSHER_07103]

A QSEv creation function shall qualify notified SEvs to a QSEv for each Security Event ID, in accordance with [2], with parameters specified in [IDSHER_03402].

3.1.4. QSEv transmission function

3.1.4.1. QSEv transmission

[Requirement: IDSHER_07108]

If a detection master (reference [11]) exists on any ECU, this requirement shall be allocated. When a QSEv creation function creates a QSEv, a QSEv transmission function shall create a frame defined below for each protocol used for the transmission from the own ECU to the detection master, and send it to the detection master.

- **When an ECU sends the frame with CAN communication or CAN FD communication:**
  The frame which contains data Label defined by QSEV_DATA_[ECU node name(*1)] in [12]. In addition, the data defined in Figure 3-2 shall be set to the QSEV_DATA_[ECU node name].
- **When an ECU sends the frame with Ethernet communication:**
  The frame which defined by the Table 3-3. In addition, the data defined in Figure 3-2 shall be set to IDS Message in the Table 3-3.

Protocol Version	Protocol Header	IdsM Instance ID	Sensor Instance ID	Event Definition ID	Count	Reserved	Context Data (*2)
msb … lsb	msb … lsb	msb … lsb	msb … lsb	msb … lsb	msb … lsb	msb … lsb	msb … lsb

Figure 3-2: Data Structure

Table 3-3: Frame format (case of Ethernet)

Layer	Protocol	Description	Note
L2	Ethernet	The values of each field shall follow [13]. In addition, the MAC address of CEN2 which have the detection master shall be set to Destination MAC address.	-
L3	IPv4	The values of each field shall follow [13]. In addition, the IP address of CEN2 which have the detection master shall be set to Destination IP address.	-
L4	TCP	The values of each field shall follow [13] However, the follow values shall be set to Destination Port Number and Source Port Number. - Destination Port Number: 50004 (0xC354) - Source Port Number: 50004 (0xC354)	-
L5	IDS	The values of each field shall store and compose the follow data by big-endian method - Message ID (4Byte): ALL 0 - Length (4Byte): sum of data length of Message ID, Length, IDS Message - IDS Message (Variable): the data defined by the Figure 3-2.	For the details of the IDS protocol, see reference [3].

*1: [ECU node name] shall be replaced by a node name of an own ECU that this document is allocated to. If Data Label is not definedin [12], please contact us.
*2: Allocated to ECU/VMs on CAN-FD or Ethernet network but **NOT** allocated to ECU/VMs on CAN network.

[Requirement: IDSHER_07208]
If a QSEv transmission function transmits QSEvs to a detection master, this requirement shall be allocated. If QSEv transmission by the QSEv transmission function wakes up network or prevents network from sleeping, the QSEv transmission function shall not transmit QSEvs.
<Note>
This requirement has been defined to avoid running out of battery due to transmitting QSEv.

3.1.5. QSEv storing function

3.1.5.1. QSEv storing

[Requirement: IDSHER_07109]

If any detection master does not exist on any ECU, this requirement shall be allocated. A QSEv storing function shall store the latest QSEvs created by a QSEv creation function into non-volatile memory for each Event Definition ID where the number of QSEvs to be stored is [NumberOfQSEvs]. However, it may not store QSEvs at unexpected reset (e.g. power source instantaneous interruption, low voltage). In addition, QSEv storing function shall be designed considering the limit of number of writes to non-volatile memory.

[Note]

If a detection master exists on an ECU, it is optional to store QSEvs.

Buffering QSEvs in RAM during IG-ON, and then writing the QSEvs into non-volatile memory at IG-OFF can be an example of the implementation of storing QSEvs in non-volatile memory considering the maximum number of writes to non-volatile memory.

[Requirement: IDSHER_07111]

UserDefMemoryDTC and DID for QSEvs storing shall be in accordance with Table 3-4, Table 3-5, and Table 3-6.

UserDefMemoryDTC and DID are defined in accordance with the following policy.

・UserDefMemoryDTC: Defined for each Event Definition ID

・DID：Defined for whole QSEv, and common among all Event Definition IDs

Table 3-4: UserDefMemoryDTC Related Information

UserDefMemoryDTC	FTB	Event Definition ID corresponding to UserDefMemoryDTC	Memory Selection
U2B21	0x00	0x8110	0x14
U2B22	0x00	0x8120	0x14
U2B23	0x00	0x8130	0x14
U2B24	0x00	0x8140	0x14
U2B25	0x00	0x8150	0x14

Table 3-5: DID for QSEv storing

DID	Data	Length [Bit]
0xA910	Protocol Version	4
	Protocol Header	4

	IdsM Instance ID	10
	Sensor Instance ID	6
	Event Definition ID	16
	Count	16
	Reserved	8
	Context Data (*1)	Variable Length

*1: Allocated to ECU/VMs on CAN-FD or Ethernet network but **NOT** allocated to ECU/VMs on CAN network.

Table 3-6: Example of QSEv storage data(Store 5 QSEvs with Event Definition ID 0x8110)

UserDefMemoryDTC	FTB	UserDefDTC SnapshotRecordNumber	Description
U2B21	0x00	0x01	Oldest QSEv of the last [NumberOfQSEv] QSEvs (DID: 0xA910)
		0x02	Second oldest QSEv of the last [NumberOfQSEv] QSEvs (DID: 0xA910)
		0x03	Third oldest QSEv of the last [NumberOfQSEv] QSEvs (DID: 0xA910)
		0x04	Fourth oldest QSEv of the last [NumberOfQSEv] QSEvs (DID: 0xA910)
		0x05	Newest QSEv of the last [NumberOfQSEv] QSEvs (DID: 0xA910)

3.1.5.2. QSEv read

[Requirement: IDSHER_07110]

If QSEvs are stored in non-volatile memory, this requirement shall be allocated. QSEvs stored in non-volatile memory shall be able to be read from off-board client and on-board client by diagnostic communication with SID 0x19 (Sub Function 0x17/0x18). However, if the QSEvs are loaded on volatile memory, these QSEvs shall be able to be read.

For the details of the diagnostics communication, see reference [9].

3.1.5.3. QSEv deletion

[Requirement: IDSHER_07204]

If QSEvs are stored in non-volatile memory, this requirement shall be allocated. QSEvs stored in

non-volatile memory shall be able to be deleted from off-board client by diagnostic communication with SID 0x14 (QSEv output MemorySelection 0x14).
For the details of the diagnostics communication, see reference [9].

3.2. Quality requirements

We define quality requirements in this section.

[Requirement: IDSHER_12201]
Write access to this system and generated QSEv by a function to terminate communication from Out-Car shall be forbidden so that this system and generated QSEv are not manipulated by the function.
<Note>
- Method example: access control by using OS function
- Illegal write access to this system and generated QSEv is detected based on "Detection of illegal access to non-volatile memory/ volatile memory/ IO (peripheral)" (IDSHER_01101, IDSHER_01102, IDSHER_01201, IDSHER_01202, IDSHER_01501, IDSHER_01502).
- Manipulation of this system is not a target of this requirement because it is detected based on "Detection of manipulation of software" (IDSHER_02301).

3.3. Constraints

None.

3.4. Parameters

We define parameters in this section.

[Requirement: IDSHER_03401]
All parameters defined in this section shall be able to be changed under conditions defined in each requirement.

[Requirement: IDSHER_03402]
QSEvs shall be created and stored with parameters in Table 3-7 and the meta-information of the parameters shall be in accordance with Table 3-8 and Table 3-9.

Table 3-7: Parameters for QSEv creation and storing

Name	Event Definition ID	Sensor Instance ID	Value (*1)
IdsMEventAggregationTimeInterval	0x8110	0x0	0.3

	0x8120	0x0	0.3
	0x8130	0x0	0.3
	0x8140	0x0	0.3
	0x8150	0x0	0.3
IdsMContextDataSourceSelector	0x8110	0x0	IDSM_FILTERS_CTX_USE_FIRST
	0x8120	0x0	IDSM_FILTERS_CTX_USE_FIRST
	0x8130	0x0	IDSM_FILTERS_CTX_USE_FIRST
	0x8140	0x0	IDSM_FILTERS_CTX_USE_FIRST
	0x8150	0x0	IDSM_FILTERS_CTX_USE_FIRST
NumberOfQSEvs	0x8110	0x0	5
	0x8120	0x0	5
	0x8130	0x0	5
	0x8140	0x0	5
	0x8150	0x0	5

Table 3-8: Meta information of parameters for QSEv creation

Name	Unit	Type	Lower limit	Upper limit
IdsMEventAggregationTimeInterval (*2)	sec	EcucFloatParam Def	0.05	10.00
IdsMContextDataSourceSelector	-	EcucEnumeration ParamDef	IDSM_FILTERS_CTX_USE_ FIRST	IDSM_FILTERS_CTX_USE_ LAST

*1: That value of IdsMEventAggregationTimeInterval is hyphen means no aggregation.

*2: If it is not available to set the value specified in the value column, the biggest value among available values smaller than the value specified shall be adopted.

Table 3-9: Meta information of parameters for QSEv storing

Name	Description	Unit	Lower limit	Upper limit
NumberOfQSEvs	The number of QSEvs to be stored	-	0	10

Appendix. A. Examples of monitored target and detection method of requirements

We show exmaples of monitored target, monitored anomaly, and detection method in this chapter.

A.1. Detection of abort of a first layer protection function for communication from Out-Car [IDSHER_04101]

A.1.1. Monitored target

The monitored target is a first layer protection function for communication from Out-Car which is designed as a resident software (resident process).

As an example, a case when a first layer protection function for communication from Out-Car is designed as a resident software (resident process) means a case when TLS is designed as software which resides in a user space and provides a proxy function to integrate TLS communication from Out-Car.

A.1.2. Monitored anomaly

Monitored anomaly regarding resident process mentioned in A.1.1 is that the process is not working.

As an example, Detection function shall detect cases when the process does not work from the start up and when the process stops during running as an anomaly. Note that the case when the process does not work properly because of illegal change of the code or configuration of the software is not a monitored anomaly of this requirement because it is detected by detection of manipulation of software (IDSHER_02301).

A.1.3. Detection method

An example of detection method is alive monitoring of process. Methods below satisfy this requirement.

- The monitor checks regularly whether the monitored target process works or not.
- The monitored target process notifies the monitor of its status of running regularly.

A.2. Detection of abnormal control flow [IDSHER_01601]

A.2.1. Monitored target

The monitored target is software composing a function to terminate communication from Out-Car.

As an example, detection methods as to software to terminate TLS communication mentioned in A.2.3 is applied to a shared library to realize TLS function (TLS library) and executable files which load TLS library.

A.2.2. Monitored anomaly

Monitored anomaly is a transition between functions that shall not occur in an authenticated flow.

We show the examples below.

- Indirect call by manipulation of a function pointer

  The program shown in the Figure 3-3 calls Func2 with a function pointer from Func1. Here, transition to any place except a head of a function by manipulation of the function pointer shall be detected.

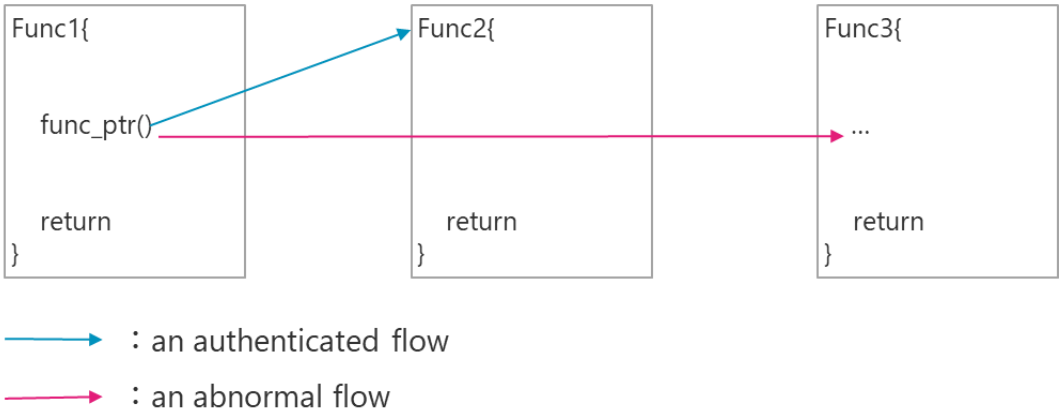

Figure 3-3: Anomaly of indirect call transition

- Return by manipulation of return address on a stack

  The program shown in the Figure 3-4 calls Func2 from Func1 and returns to Func1 after Func2 is executed. Here, transition to any place other than the place of function call source caused by manipulation of the return address on the stack shall be detected.

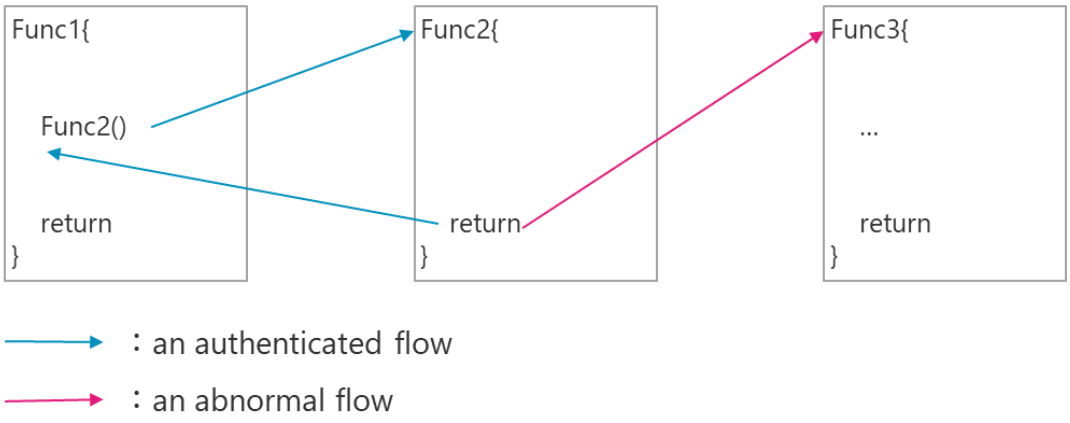

Figure 3-4: Anomaly of function return transition

Note that manipulation of the code is not a monitored anomaly of this requirement because it is detected by detection of manipulation of software (IDSHER_02301).

A.2.3. Detection method

Examples of detection method are Control Flow Integrity (CFI) and shadow stack.

- CFI

  As an example, CFI inserts code that vefries target address of indrect transition.Verifying this code during transition detects a transition between functions that shall not occur in an authenticated flow.

- Shadow stack

  As an example, shadow stack stores return address to a specific area in addition to a normal stack at the time of function call. Verifying whether both return addresses are the same, detects a transition between functions that shall not occur in an authenticated flow.

A.3. Detection of illegal access with a path [IDSHER_01101, IDSHER_01201, IDSHER_01501]

A.3.1. Monitored target

The monitored target is software that satisfy all conditions below.

- It is software composing a function to terminate communication from Out-Car.
- It can access resources by a path of a file system.
- It can access non-volatile memory, volatile memory and IO (peripheral).

Path is one of the identifier to indicate the place of resource. As an example, the string "/path/to/data" to indicate the place of resource is a path.

Examples of IO (peripheral) in this requirement are CAN controller, Ethernet controller, USB controller, Wi-Fi module, Bluetooth module and HSM.

A.3.2. Monitored anomaly

The monitored anomaly is an access to unpermitted resource by the software above. We show an example of an illegal access with a path when the software works as one process.

- An illegal access with a path

  The software has writable access in directories and files under /foo/bar/ but does not have writable access in directories and files under /foo/baz/ in non-volatile memory of EP region shown in the Figure 3-5 Here, when the software accesses or attempts to access /foo/baz/file_6, which is not permitted to write, this event shall be detected. Illegal access to volatile memory and IO (peripheral) shall be also detected.

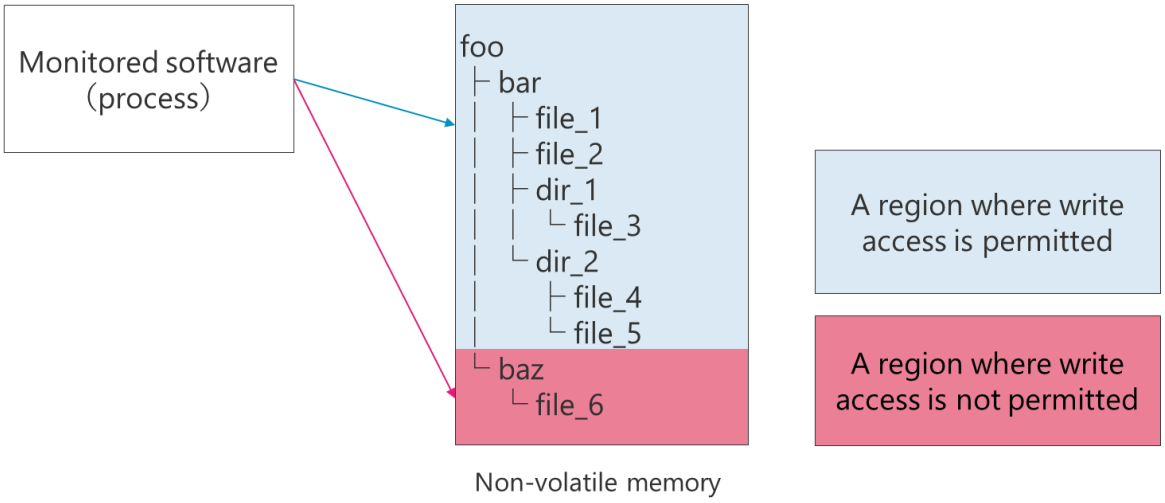

Figure 3-5: Illegal access with a path

Note that as a prerequisite, the software shall be designed to prevent from unnecessary access.

A.3.3. Detection method

An example of detection method is monitoring file access. As an example, permission of read access, write access, and execution access can be set for each file in Linux. Unpermitted operation shall be detected as permission violation.

A.4. Detection of illegal access with an address [IDSHER_01102, IDSHER_01202, IDSHER_01502]

A.4.1. Monitored target

The monitored target is software that satisfy all conditions below.

- It is software composing a function to terminate communication from Out-Car.
- It can access resources by an address.
- It can access non-volatile memory, volatile memory and IO (peripheral).

Address is one of the identifier to indicate the place of resource. As an example, the value “0x1000” to indicate the place of the device is an address.

Examples of IO (peripheral) in this requirement are CAN controller, Ethernet controller, USB controller, Wi-Fi module, Bluetooth module and HSM.

A.4.2. Monitored anomaly

The monitored anomaly is an unpermitted access to resource by the software above. We show an example of an illegal access with an address when the software works as one process.

・ An illegal access with an address

The software has some areas where the software may write and other areas where the software may not write in volatile memory of EP region shown in the Figure 3-6. Here, when the software accesses or attempts to access unpermitted area 0x1000 to write, this event shall be detected. Illegal access to non-volatile memory and IO (peripheral) shall be also detected.

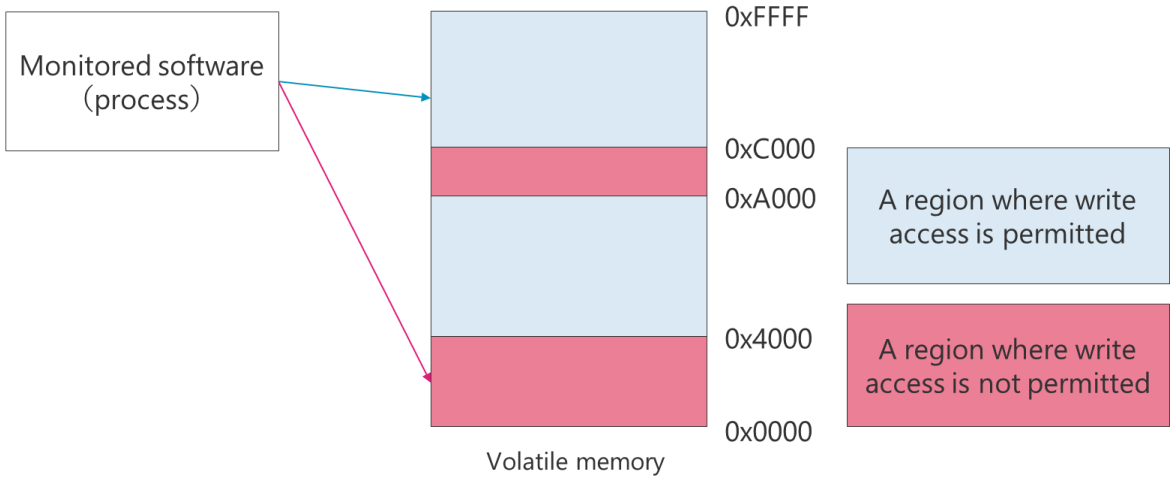

Figure 3-6: Illegal access with an address

Note that as a prerequisite, the software shall be designed to prevent from unnecessary access.

A.4.3. Detection method

An example of detection method is monitoring memory access. As an example, Linux manages its memory with MMU (Memory Management Unit). Unpermitted memory access shall be detected as segmentation violation.

A.5. Detection of illegal usage of function [IDSHER_01401]

A.5.1. Monitored target

The monitoring target is software composing a function to terminate communication from Out-Car.

A.5.2. Monitored anomaly

An example of a function that requires an authority to use is system call.

When the monitored target software uses or attempts to use unpermitted system call, the event shall be detected. Note that as a prerequisite, the software shall be designed to prevent from using unnecessary function.

A.5.3. Detection method

An example of detection method is monitoring use of system call. In this method, system calls which

the software can use are defined as an allow list in advance per software. When the software uses any system call, this method checks whether the use is permitted or not with the allow list. If the software is not permitted to use the system call, the method prevents the software from using the system call.

A.6. Detection of manipulation of CSP/PSP [IDSHER_02101]

A.6.1. Monitored target

The monitored target is CSP/PSP an entry point region has in non-volatile memory.
The examples of CSP are secret key, random number seed used for generating cryptographic key, password. And the examples of PSP are public cryptographic keys, public key certificates, self-signed certificate. Note that the definition of CSP/PSP is in accoradance with reference[6].

A.6.2. Monitored anomaly

The monitored anomaly is that the data is manipulated at the usage of CSP/PSP. Here, the usage of CSP/PSP indicates any operation that uses the data. Examples of the usage of CSP/PSP are, encryption, decryption, authentication and loading the data in non-volatile memory into volatile memory and HSM to perform the operantions.

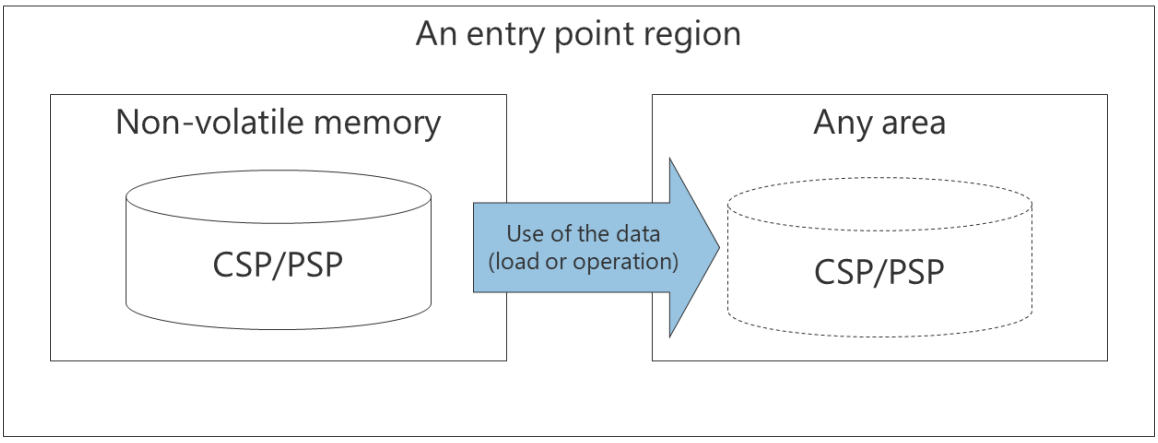

Figure 3-7: Detection of manipulation of CSP/PSP

A.6.3. Detection method

An example of detection method is detection of file manipulation. In the method of detection of file manipulation, MAC of the monitored data is generated in advance and the MAC is verified at the usage of it. If the MAC values are different, the event shall be detected as an anomaly.
Here, a key to generate MAC shall be stored properly like it is stored in the HSM.

A.7. Detection of manipulation of software [IDSHER_02301]

A.7.1. Monitored target

The monitored target is the code of software composing a function in an entry-point region. In addition, the data controlling the behavior of the software which can be updeted only by reprogramming is also the monitored target. As an example, the data controlling the behavior of the software means configuration data (configuration file) of the software.

A.7.2. Monitored anomaly

The Monitored anomaly is a manipulation of the monitored target when the software starts up. As an example, regarding to an ECU/VM whose software can be added and updated only by reprogramming, when the software is manipulated at the startup of this ECU/VM, the detection function shall notify a SEv creation function of the anomaly.

However, regarding to an ECU/VM whose software can be added and updated by other ways, the monitored anomaly and the detection method shall be considered depending on the way.

A.7.3. Detection method

An example of detection method is secure boot.

- Secure boot

  Secure boot verifies the integrity of the monitored software at the startup of the ECU/VM.

  Verifying the integrity prevents the manipulated software from working.