HEAVENS

HEAling Vulnerabilities to Enhance Software Security and Safety

Document Title	Security models
Document Type	Deliverable
Document Number	D2
Document Responsible	Aljoscha Lautenbach, aljoscha@chalmers.se, Chalmers Mafijul Islam, mafijul.islam@volvo.com, Volvo AB
Document Version	2.0
Document Status	Released (March 18, 2016)
Dissemination Level	Public
Last Change	March 18, 2016

Project Acronym	HEAVENS
Project Title	HEAling Vulnerabilities to ENhance Software Security and Safety
Research Program	Vinnova/FFI (Fordonsutveckling/Vehicle Development), Sweden
Diary Number	2012-04625
Project Duration	April 2013 – March 2016
Project Coordinator	Mats Olsson, mats.olsson.2@volvo.com , Volvo AB

This page is intentionally left blank

Executive Summary

This deliverable (**D2 Security models, Release 2, Version 2.0**) presents the results and achievements of work package WP2 (Security models) of the HEAVENS project.

The goal of this deliverable is to present a systematic approach of deriving security requirements for the automotive Electrical and/or Electronic (E/E) systems. It suggests an adaption of generic security engineering process for the automotive domain. The deliverable presents state-of-the-art threat analysis and risk assessment methodologies, processes, frameworks and tools, considering various industrial domains, for example, IT security, telecommunications, software engineering and defense. It presents the results obtained from performing a critical review of the state-of-the-art threat analysis and risk assessment in the context of the automotive industry. Based on this, a new security model − **HEAVENS security model** − for the automotive industry is proposed to facilitate deriving security requirements for the automotive E/E systems. The model includes methods, processes and tool support with focus on threat analysis and risk assessment aspects of security engineering process. Then, this deliverable presents the results obtained from a proof-of-concept implementation and evaluation of the proposed HEAVENS security model by using a couple of automotive use cases. This deliverable also discusses the HEAVENS security model in the context of existing standards, for example, ISO 26262 for functional safety and Common Criteria for IT Security Evaluation.

This page is intentionally left blank

Contributors

Editors	Affiliation	Email
Aljoscha Lautenbach	Chalmers	aljoscha@chalmers.se
Mafijul Islam	ATR, GTT, Volvo AB	mafijul.islam@volvo.com
Contributors	**Affiliation**	**Email**
Mafijul Islam	ATR, GTT, Volvo AB	
Christian Sandberg	ATR, GTT, Volvo AB	
Andreas Bokesand	VE, GTT, Volvo AB	
Tomas Olovsson	Chalmers	
Pierre Kleberger	Chalmers	
Aljoscha Lautenbach	Chalmers	
Andrew Söderberg-Rivkin	Chalmers/Volvo AB	
Sathya Prakash Kadhirvelan	Chalmers/Volvo AB	
Anders Hansson	SECTRA AB	
Henrik Broberg	Volvo Car Corporation (VCC)	

HEAVENS Consortium

VOLVO VOLVO

ARC CORE COMBITECH omega point. SECTRA

CHALMERS SP your Science Partner

This page is intentionally left blank

Document Change History

Version	Date	Contributor	Description
0.1	May 09, 2014	Aljoscha Lautenbach	First draft of “D2 Security models” created.
1.0	Sep 29, 2014	Mafijul Islam	Version 1.0, Release 1 of D2.
2.0	Feb 23, 2016	Christian Sandberg	Moved material from D3.1 to this document. Draft version
2.0	Mar 18, 2016	Mafijul Islam	Updated with comments and references. Released version

This page is intentionally left blank

Table of Contents

This page is intentionally left blank

List of Figures

List of Tables

This page is intentionally left blank

List of Abbreviations

ASIL	Automotive Safety Integrity Level
AUTOSAR	AUTomotive Open System Architecture
CAN	Controller Area Network
C2C-CC	CAR 2 CAR Communication Consortium
CC	Common Criteria
CIA	Confidentiality, Integrity, Availability
CIAAA	Confidentiality, Integrity, Availability, Authenticity and Accountability
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
DFD	Data Flow Diagram
DREAD	Damage, Reproducibility, Exploitability, Affected users, Discoverability
DTC	Diagnostic Trouble Code
E2E	End-to-End
EAL	Evaluation Assurance Level
ECU	Electronic Control Unit
E/E	Electrical and/or Electronic
FMEA	Failure Mode and Effect Analysis
HARA	Hazard Analysis and Risk Assessment
IAS	Information Assurance and Security
IL	Impact Level
ITS	Intelligent Transportation System
OBD	On-Board Diagnostics
OEM	Original Equipment Manufacturer
OWASP	Open Web Application Security Project
PASTA	Process for Attack Simulation and Threat Analysis
PCI - DSS	Payment Card Industry Data Security Standard
QM	Quality Management
RSL	Road Speed Limit

SAE	Society of Automotive Engineers
SDL	Security Development Lifecycle
SDLC	Software Development LifeCycle
SHE	Secure Hardware Extension
SIL	Safety Integrity Level
SL	Security Level
STRIDE	Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege
TAL	Trust Assurance Level
TARA	Threat Analysis and Risk Assessment
TOE	Target of Evaluation
TL	Threat Level
TVRA	Threat Vulnerability and Risk Analysis
V2V	Vehicle to Vehicle
V2I	Vehicle to Infrastructure
WP	Work Package

1. Introduction

This document corresponds to deliverable "**D2 Security models"** of the HEAVENS project. The current version (Version 2.0) is the final release of this deliverable. The deliverable summarizes the results from the activities performed within the scope of work package **WP2 (Security models)** of the project.

1.1 Background

Safety is traditionally regarded as one of the most important attributes in the automotive industry. In contrast, security has hardly been addressed in software-intensive automotive systems. This is where the HEAVENS project comes into the picture: the project aims to identify security vulnerabilities in automotive E/E systems and define methodologies along with tools to perform security evaluation. A common way of assessing security will improve the industry's ability to deliver safe and secure vehicles. The target of HEAVENS is to equip the "owners" with "countermeasures" to facilitate protecting their "assets" by minimizing the "risk" associated with the "vulnerabilities" that can be exploited by the "threats" originating from the "threat agents". Furthermore, the project intends to investigate the interplay of safety and security in the context of automotive Electrical/Electronic (E/E) systems. Please refer to the HEAVENS project proposal [15] for more information.

1.2 Objective

The primary objective of the HEAVENS security model is to outline a framework for identifying security requirements in the context of the automotive E/E systems. This is similar to the notion of functional safety requirements as described in the ISO 26262. To accomplish this objective, we identify assets and threats associated with the assets. We then map the threats with the security attributes and derive a security level for each asset-threat pair by estimating threat level along with impact level. Consequently, the HEAVENS security model as described in this deliverable focuses on methods, processes and tool support for threat analysis and risk assessment with respect to the automotive E/E systems. Table 1-1 shows how the deliverable D2 contributes to the fulfillment of the HEAVENS project goals as stated in the project proposal [15].

1.3 Scope and limitation

This deliverable focuses on threat analysis and risk assessment for the automotive industry. However, an overview of a number of security models from other domains (e.g., IT security, web applications, software design, telecommunications and defense) is also provided, and their applicability to the automotive industry is discussed. On the other hand, a detailed account of all possible models is out of the scope of this document. A critical review of the most relevant existing threat analysis and risk assessment techniques is presented to provide background and rationale for proposing a new security model for the automotive E/E systems. The current version (Version 2.0) of the HEAVENS security model does not suggest countermeasures or security mechanisms to assist in fulfilling the derived security requirements. Also, the model currently does not establish explicit relationship between threats and vulnerabilities.

Table 1-1: Relevance and contributions of D2 in relation to the HEAVENS project goals.

HEAVENS project goals (Relevant WP)	Deliverable D2 Security models	
	Relevance	Contributions
▪ Identify needs and requirements of security in the automotive industry. (WP1)	Low	Current version of D2 provides feedback to the next release of D1.1 Needs and requirements.
▪ Study and identify state-of-the-art in security in the automotive industry. (WP1)	Medium	State-of-the-art in threat analysis and risk assessment is presented, which provides input to the next release of D1.2 A stat-of-the-art report on vehicular security.
▪ Identify potential threats, threat agents and vulnerabilities to construct security models. (WP2) ▪ Security issues from software engineering and traditional networking as well as from other domains will be considered to map those in the context of the automotive domain. (WP2)	High	Primary goal of D2. ▪ Introduces a new security model: threat analysis and risk assessment method, process and tool support. ▪ Investigates security models from IT security, telecommunication, defense, and software engineering domains in the context of the automotive E/E systems.
▪ Define methodologies and identify tool support for evaluating software security. (WP3)	Medium	Some methodologies and tools for modeling have been evaluated already, and the models are expected to guide the choices in WP3.
▪ Investigate the interplay of safety and security in the E/E architecture, considering ISO 26262, AUTOSAR and other relevant standards. (WP4)	Low	Relationship between functional safety and cyber-security has been discussed to some extent: hazard analysis and risk assessment VS. threat analysis and risk assessment, assuming concept phase of product development lifecycle.
▪ Demonstrate proof of concepts. (WP5)	Not relevant	

1.4 Relation to other project activities

The findings of WP1 (Needs and Requirements) establish the foundation for the activities that we have performed in WP2 (Security models). This then establishes the foundations for work in WP3 (Security testing and evaluation) and WP4 (Safety, security and E/E architecture). The relationship of the work packages is shown in Figure 1-1. It is expected that there will be a continuous stream of feedback across the first four WPs (WP1 – WP4), i.e., that results achieved in one of the WPs can influence, guide and refine the results of the other WPs.

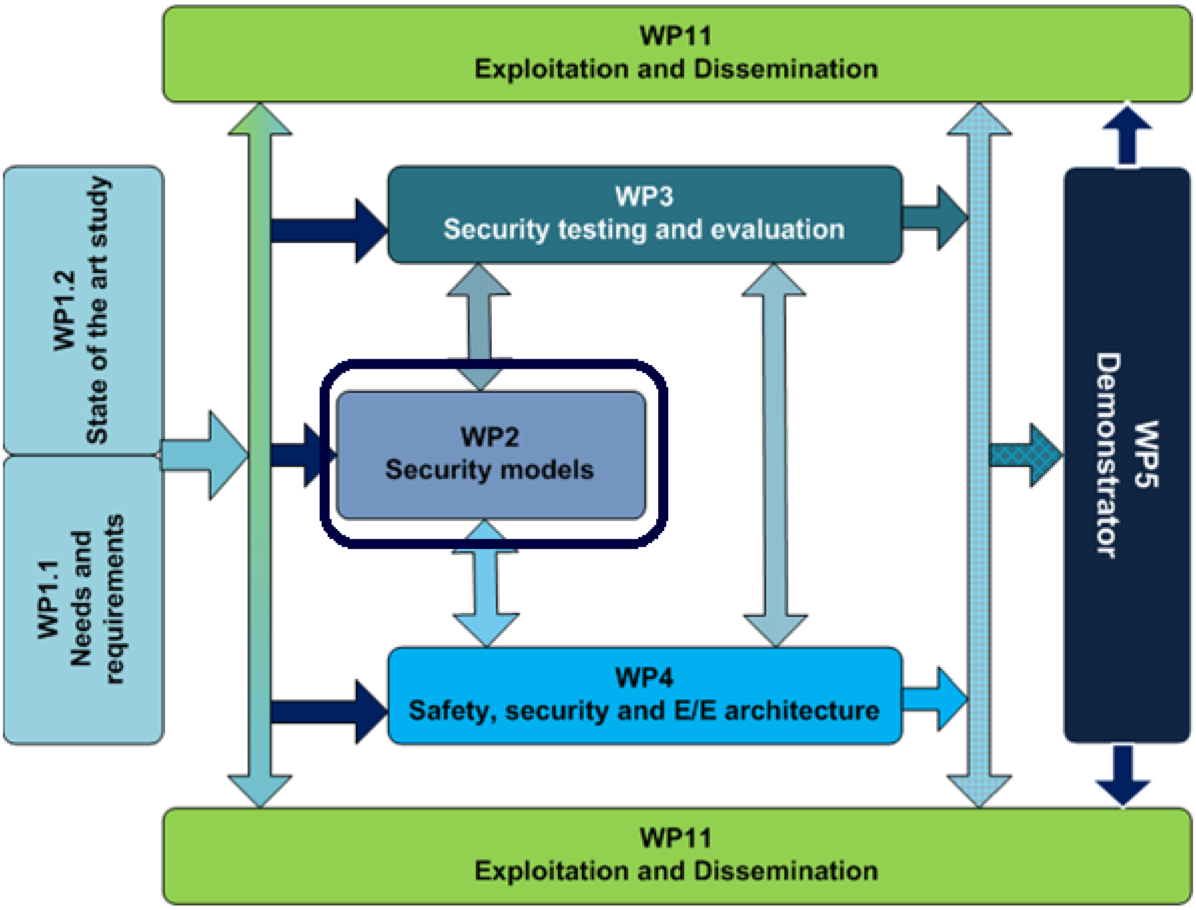

WP11
Exploitation and Dissemination
WP1.2
State of the art study
WP1.1
Needs and requirements
WP3
Security testing and evaluation
WP2
Security models
WP4
Safety, security and E/E architecture
WP5
Demonstrator
WP11
Exploitation and Dissemination

Figure 1-1: Work packages (WPs) of the HEAVENS project and their relationship.

1.5 Document outline

The remainder of the deliverable is structured as follows. Chapter 2 provides an overview of different approaches towards security engineering and modeling, and a short discussion on security metrics. Chapter 3 presents a number of threat analysis and risk assessment approaches, including frameworks, processes, methods and tools. The HEAVENS security model is presented in Chapter 4. This is followed by a proof-of-concept implementation and evaluation of the model based on a couple of automotive use cases in Chapter 5. Finally, Chapter 6 summarizes the contributions and points to potential future works.

This page is intentionally left blank

2. Approaches to security modeling

> *"Models are deliberate, purposeful simplifications of reality. They are the stock-in trade of technologists and empirical scientists [...]. Models are the means of obtaining useful generalizations and creating predictive theories."*
>
> — Agnes A. Kaposi [23]

This chapter provides basic definitions and a short overview of security concepts, safety and security engineering and different approaches towards security modeling and security metrics.

In the context of system development, a *model* is a system abstraction with the goal to simplify the design or comprehension of a complex system. It should allow quantitative or qualitative reasoning about the system. *Security models* are therefore meant as a tool to reason about the security of a system. One of the main challenges is to find the right level of abstraction. In order to discuss security models further, we first need to give a definition of security. Security is traditionally defined by a number of attributes which constitute its core. The security attributes which are used in the HEAVENS project have been defined in Deliverable D1.1 Needs and Requirements [16], but will be reproduced in Section 2.1 for convenience.

The remainder of this chapter is structured as follows. Section 2.1 provides a definition and short discussion of security, Section 2.2 provides a rough overview over the security and safety engineering processes, Section 2.3 discusses a number of different security models, and finally security metrics and model evaluation are discussed in Section 2.4.

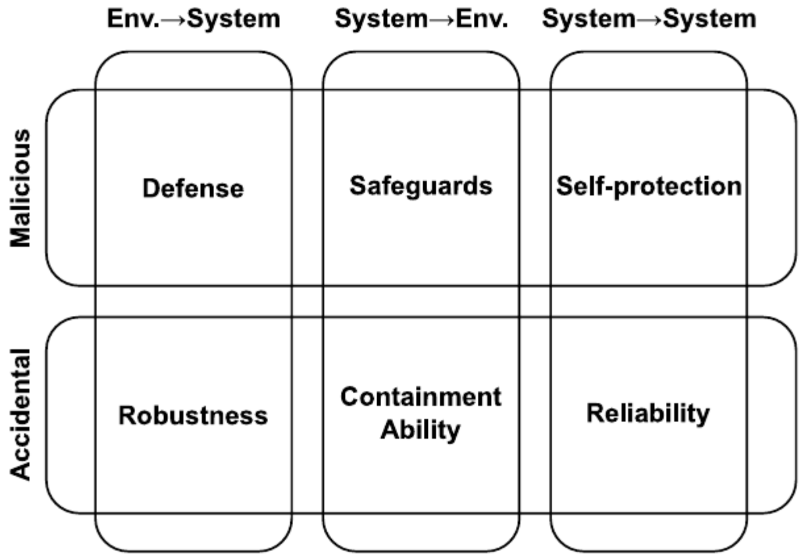


Figure 2-1: SEMA referential framework [36].

2.1 What is security?

Piètre-Cambacédès and Chaudet examined the ambiguities of the terms "security" and "safety" in different industries [36], most notably in the critical infrastructure protection domain. According to their findings there are many overlapping definitions, but Firesmith [14] and Line et al. [25] provided clear and exclusive definitions. Firesmiths's definitions of safety and security are mostly based on intent, i.e. he differentiates between accidental and malicious events [14]. Line et al. on the other hand distinguish the terms based on the directionality of impact, i.e. if the environment affects the system or vice versa [25].

Security according to Firesmith - “The degree to which malicious harm is prevented, reduced and properly reacted to [14]”. **Security according to Line et al.** - “The inability of the environment to affect the system in an undesirable way [25]”. Based on these two main dimensions, Piètre-Cambacédès and Chaudet developed the SEMA referential framework [36], depicted in Figure 2-1.

For the most part we are concerned with what Piètre-Cambacédès and Chaudet call Defense, i.e. the external influence of a system with malicious intent. However, the other domains should not be forgotten. We will refine our definition of security in the next two subsections which discuss security attributes and security objectives.

2.1.1 Security attributes

The CIA triad (Confidentiality, Integrity, Availability) has been one of the core principles of IT security for many years. Consequently, confidentiality, integrity and availability are often referred to as primary security attributes. However, in recent years, IT security has evolved drastically and reached beyond the realm of traditional computer and network security. Nowadays security is one of the central concerns in virtually every industrial domain such as automotive. This has motivated security researchers and practitioners to extend the classic CIA triad to embrace new security attributes.

In line with this, based on extensive literature analysis, the Information Assurance & Security (IAS) Octave has been developed and proposed as an extension of the CIA triad in 2013. The IAS Octave includes confidentiality, integrity, availability, privacy, authenticity & trustworthiness, non-repudiation, accountability and auditability [55]. In the automotive domain, several research projects (e.g., EVITA [11], PRESERVE [39], OVERSEE [32], SEVECOM [43]) have adopted a slightly modified view on security attributes. Accordingly, in the HEAVENS project, we limit ourselves to eight security attributes: confidentiality, integrity, availability, authenticity, authorization, non-repudiation, privacy, and freshness. The adopted security attributes are introduced in the subsequent sub-sections.

2.1.1.1 Confidentiality

Confidentiality refers to “absence of unauthorized disclosure of information” [3]. In ISO/IEC 27000 [52] confidentiality is defined as the property that information is not made available or disclosed to unauthorized individuals, entities, or processes. It is one of the primary security attributes. Privacy relies on confidentiality and can be considered as a special case of confidentiality [53]. The confidentiality attribute can be viewed as a property that consists of two components – a set of information and a set of authorized entities, individuals or processes.

2.1.1.2 Integrity

Integrity refers to “absence of improper system alterations” [3]. In ISO/IEC 27000 [52] integrity is defined as the property of protecting the accuracy and completeness of assets. It is one of the primary security attributes.

2.1.1.3 Availability

Availability refers to "readiness for usage" [3]. In ISO/IEC 27000 [52], availability is defined as the property of being accessible and usable upon demand by an authorized entity. It is one of the primary security attributes.

2.1.1.4 Authenticity

In ISO/IEC 27000 [52], authenticity is defined as the property that an entity is what it claims to be. Authenticity is the property of being genuine and being able to be verified and trusted [54]. Consequently, authentication is the process of verifying the identity or other attributes claimed by or assumed of an entity (user, process, or device), or to verify the source and integrity of data [54]. Authentication includes [32], [43]:

- ID authentication: receiver is able to verify a unique ID of the sender.
- Property authentication: receiver is able to verify that the sender has certain properties, e.g., sender is a car, a traffic sign, etc.
- Location authentication: receiver is able to verify that the sender is actually at the claimed position or that message location claim is valid.

  Additionally, authentication may include:

- Time authentication: receiver is able to verify that the quantum of information has been created/sent by the sender at claimed time. This is related to freshness attribute of security.

2.1.1.5 Authorization

Authorization is defined as access privileges granted to a user, program, or process or the act of granting those privileges [54]. It is an access control property that can incorporate three components:

1. a set of assets (e.g., vehicular networks, computing power), actions (e.g., read access, write access) and or/information (e.g., vehicular data)
2. a set of authorized entities (stakeholder, user, program or process)
3. the duration (time period) of the authorization.

Access rules shall be defined in the corresponding security policy derived during a security requirements engineering process, which determines the access rights for each authorized entity [32]. This is required to ensure that stakeholders only have access to assets that they are authorized to access, in accordance with their expected activities and only for the time period as required to complete them.

2.1.1.6 Non-repudiation

In ISO/IEC 27000 [52], non-repudiation (also known as auditability [32], [43]) is defined as the ability to prove the occurrence of a claimed event or action and its originating entities. This attribute is an assurance that the sender of information is provided with proof of delivery and the recipient is provided with proof of the sender's identity, so neither can later deny having processed the information [54]. It provides protection against an individual falsely denying having performed a particular action. It further provides the capability to determine whether a given individual took a particular action, such as creating information, sending a message, approving information, and receiving a message [54]. Non-repudiation is a property that consist of three components – a set of actions (e.g., message sent, message received), a set of entities (e.g., sender, receiver) and a time limit (e.g., time limited, permanent).

2.1.1.7 Privacy

Privacy and anonymity can be viewed as special forms of the confidentiality attribute. Privacy applies to an entity and a set of information [53]. This property is guaranteed if the relation between the entity and the set of information is confidential. Anonymity, for instance, is the property that the relation between an entity and its identity is confidential [53]. Privacy is frequently a major concern when the entity involved is an individual or a vehicle owned by an individual [53]. For example, an adversary constantly recording the location of a vehicle and knowing the identity of the driver may be considered as violating the driver's privacy with respect to her movements.

Privacy requirements are needed to ensure that the anonymity of stakeholders and confidentiality of their sensitive information are assured. Sensitive information introduced by the application shall be identified [53]. For users, sensitive information may include current location of a specific vehicle and/or driver whereas for vehicle manufacturers, sensitive information may include design information and performance data [53]. However, privacy requirements can potentially conflict with other security requirements, for example, non-repudiation.

2.1.1.8 Freshness

Freshness applies to a quantum of information, a receiving entity and a given time [53]. Essentially this means that every message sent includes a timestamp to uniquely identify the message. This is done to enable the identification of messages which have been received and processed earlier. Ensuring freshness can be used to prevent replay attacks, for example.

2.1.2 Security objectives

According to Common Criteria [6], a security objective is a statement of intent to counter identified threats and/or satisfy identified organization security policies and/or assumptions. Security objectives are closely related to the stakeholders of the system under investigation or the Target Of Evaluation (TOE). The Open Web Application Security Project (OWASP) [34] suggests breaking down the application's security objectives into the following categories: Identity, Financial, Reputation, Privacy and Regulatory, and Availability Guarantees. According to OWASP, other sources may as well have impact on establishing security objectives [34]: Laws, Regulations, Standards, Legal Agreements, Corporate Information Security Policy, etc. However, OWASP specified security objectives primarily intended to be applicable to web applications and focus quite a lot on the privacy attribute of security. Hence, this needs to be customized to be applied in the HEAVENS project in the context of the automotive E/E systems. In contrast, the E-safety Vehicle Intrusion Protected Applications Project (EVITA) project [41] proposes high-level security objectives and links generic security threats with those objectives. This is more suitable for the HEAVENS project. The security objectives mentioned in D2.3 of EVITA [41] are as follows: Operational, Safety, Privacy, and Financial.

Based on the above mentioned security objectives, we adopt the following security objectives in the HEAVENS project:

Safety: To ensure the safety of vehicle occupants, other road users and infrastructures, i.e.

- prevent unauthorized modification of vehicle functions and features that can affect safety.
- prevent denial of use/service that can cause an accident.

Financial: To prevent negative financial impact, i.e.

- to prevent fraudulent commercial transactions.
- to prevent theft of vehicles.
- to prevent Original Equipment Manufacturer (OEM) intellectual property infringement.
- to prevent damage to OEM reputation.
- to prevent insurance and warranty fraud.

Operational: To maintain the intended operational performance of all vehicle, Intelligent Transportation System (ITS) functions and related infrastructures, i.e.

- prevent unauthorized modification of functions and features that can affect expected operations of vehicles and infrastructures.
- prevent users from expected vehicle services and functionalities, i.e., denial of service/use.

Privacy and legislation: To ensure the privacy of all relevant parties and to fulfill all relevant legislations, i.e.

- to protect the privacy of vehicle drivers, vehicle owners and fleet owners.
- to protect the intellectual property of vehicle manufacturers and their suppliers.
- to protect user identities (e.g. vehicle owners, transportation service providers) from abuse, that is, impersonation of a victim to perform actions with stolen identities must be prevented.
- to fulfill requirements of relevant privacy legislations.
- to fulfill requirements of relevant driving and environmental related legislations.
- to fulfill requirements of relevant standards and laws.

2.2 Security and safety engineering

Security engineering is an engineering discipline concerned with securing a system and it encompasses the entire process from system design to deployment and maintenance [1], [30]. There are significant similarities between security engineering and safety engineering.

Jonsson defined a basic combined model for security and dependability in terms of system inputs and outputs [22], a refined version of which is depicted in Figure 2-2. This graphical representation should be intuitively understandable for both safety and security engineers. Security and safety engineering have essentially the same core processes. Especially in safety-critical environments, like the automotive industry, security and safety engineering are risk-based.

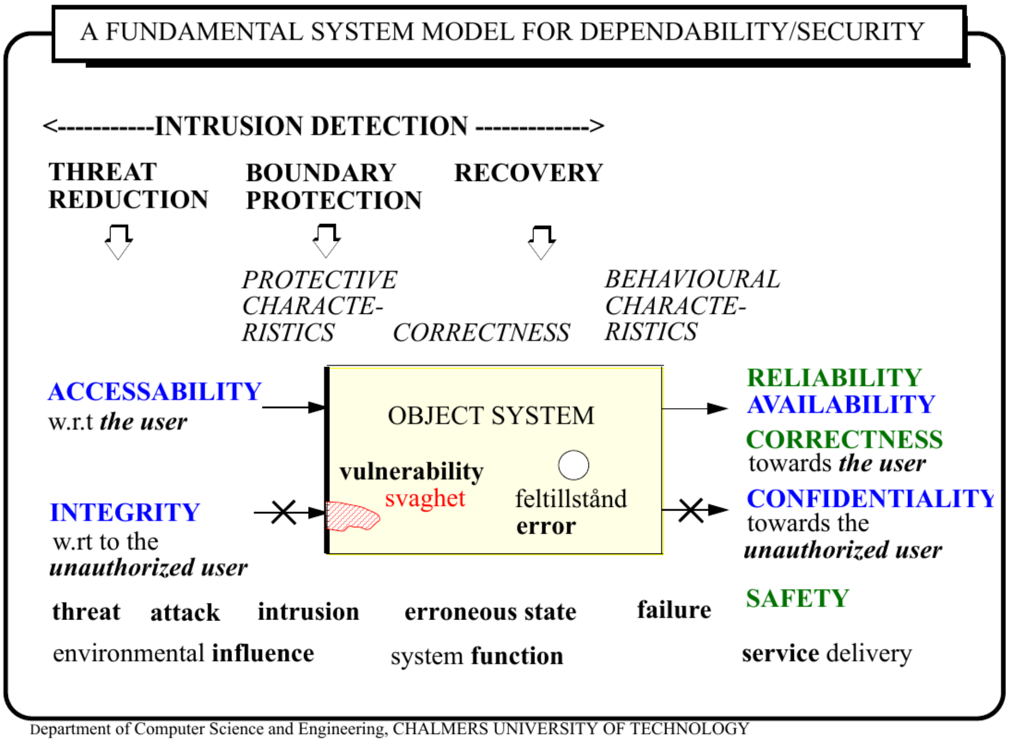


Figure 2-2: Basic security and dependability system model [21], [22].

Figure 2-3 shows a generic risk management process, as defined in ISO 31000:2009 Risk Management [18]. *Context establishment* is a very broadly defined activity, but in this specific context it can be seen as the establishment of a basic system model or system characterization [30], which is required for all subsequent steps. If there is no system model, there is no base for a risk assessment.

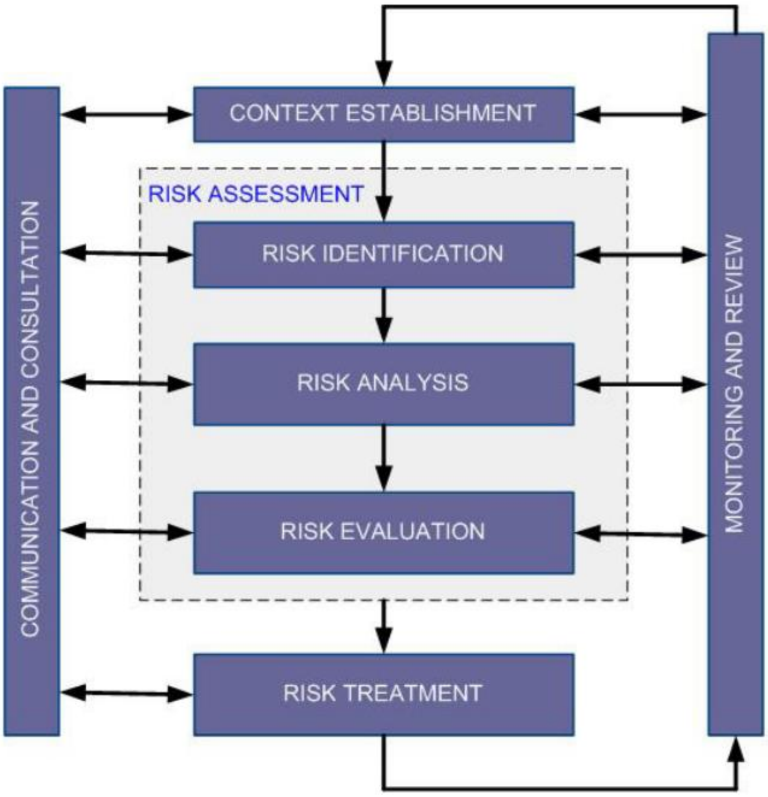


Figure 2-3: A generic risk management process [17], [18].

Risk identification is a crucial activity which is concerned with identifying events which have the potential to cause harm. According to ISO 27005:2013 Information security risk management [17], encompasses the following sub-steps:

1. Identification of assets
2. Identification of threats
3. Identification of existing controls
4. Identification of vulnerabilities
5. Identification of consequences

Risk analysis is the process of assigning each identified risk a severity and a corresponding likelihood of occurrence [35], [37], [49]. This can be a qualitative analysis, a quantitative analysis, or a mixture of both. Together, the severity and likelihood determine the risk level. During the *risk evaluation* phase, the determined risk levels will be evaluated based on the risk evaluation and risk acceptance criteria. This will result in a ranking for risk treatment priorities. Finally, *risk treatment* (see Figure 2-4) determines to which degree and in which way a risk needs to be modified in order to be deemed acceptable. For a more detailed explanation of each of these steps, please consult ISO 27005 [17].

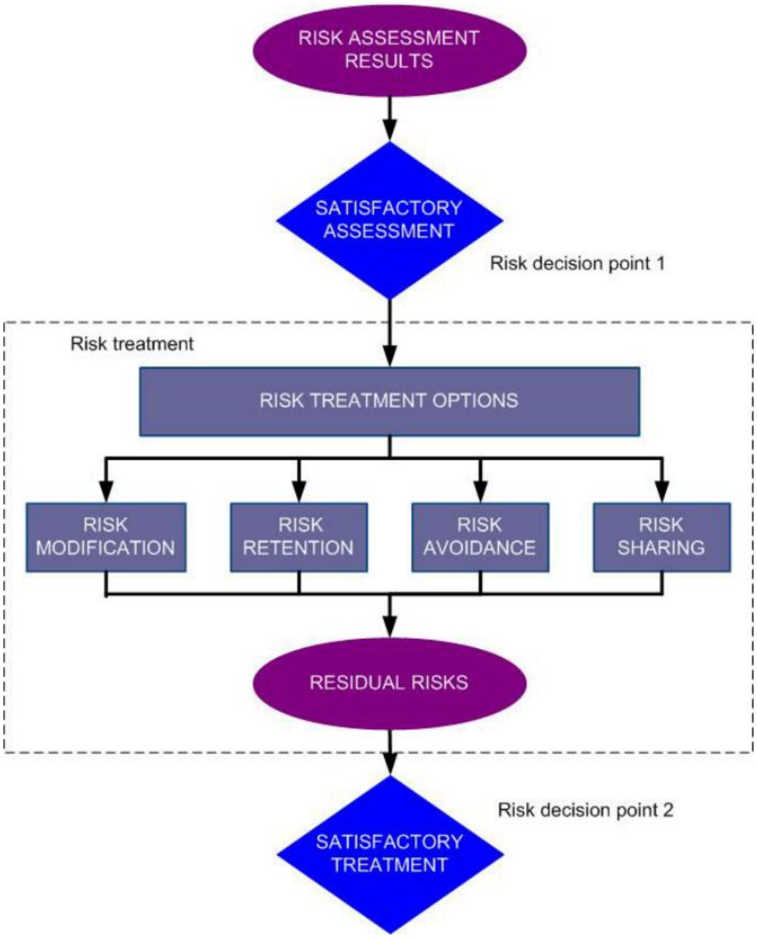


Figure 2-4: The risk treatment activity [17].

Now that we have introduced a (mostly generic) risk management approach, we can point out the general similarities in the security and safety engineering processes. That both processes require a basic system model before they can be applied should be obvious.

In ISO 26262 [19] the activities risk identification, risk analysis and risk evaluation are aggregated under the term Hazard Analysis and Risk Assessment (HARA). Consider Figure 2-5 for an overview of the functional safety requirements elicitation process in ISO 26262. HARA has a sister concept in security engineering, Threat Analysis and Risk Assessment (TARA). Piètre-Cambacédès and Bouissou in the article on “Cross-fertilization between safety and security engineering” [35], which discusses the similarities and differences of safety and security engineering in quite some detail. They also provide an overview of how the techniques of the two fields have influenced each other.

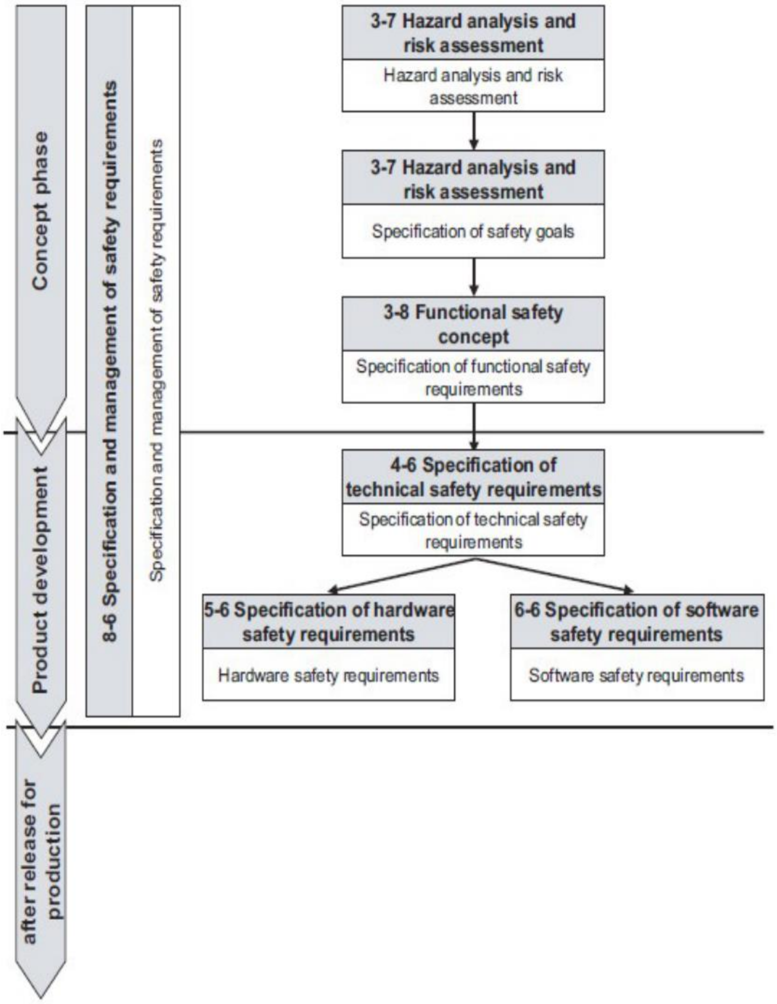

Concept phase
Product development
after release for production
8-6 Specification and management of safety requirements
Specification and management of safety requirements
3-7 Hazard analysis and risk assessment
Hazard analysis and risk assessment
3-7 Hazard analysis and risk assessment
Specification of safety goals
3-8 Functional safety concept
Specification of functional safety requirements
4-6 Specification of technical safety requirements
Specification of technical safety requirements
5-6 Specification of hardware safety requirements
Hardware safety requirements
6-6 Specification of software safety requirements
Software safety requirements

Figure 2-5: Hierarchy of safety goals and functional safety requirements in ISO 26262 [20].

To demonstrate how close security and safety engineering are in terms of processes, consider that simply replacing “hazard” with “threat” and “safety” with “security” in Figure 2-5 may yield a usable process for security engineering, too. It should be noted that, analogous to safety engineering, the desired outcome of the security engineering process is a risk-based threat rating with corresponding functional security requirements per asset.

As a final side note, it should be mentioned that the SAE International is currently working on Automotive Security Guidelines which are in-line with the ISO 26262 development processes. Recently, Committee Chair of the SAE Security Guidelines and Risk Management Task Force delivered a lecture on “Automotive Cyber-Security” [66]. Interestingly and importantly, the goal and direction of the SAE Committee activities seem to be closely aligned with ours.

2.3 Security modeling

As mentioned previously, security models are an abstraction which should allow to reason about the system. As one can expect, this allows for a very large and diverse range of security models. Figure 2-2 depicts one of the most high-level, abstract security models possible. There are also different models for almost every kind of security attribute, each of which has a different goal. However, one of the most common meanings of “security model” is a threat model, i.e. a model which helps to identify all possible threats to the system. Another common instantiation is that of access control models (e.g. Bell-LaPadula, the Chinese Wall model, etc.). In general, there are two main types of models:

- Quantitative Models (mostly stochastic models)
- Qualitative Models (descriptive models)

In this version (Version 2.0) of the deliverable, we only point to a number of survey papers of different security models, instead of summarizing the most important models here.

Fabian et al. [12] discuss several security requirements engineering methods. Felderer et al. [13] provide a survey of different methods related to security engineering, considering the evolution of software. Trivedi et al. [47] came up with an extension of the dependability model of Avivzienis et al. [3], and discussed several stochastic models, such as reliability block diagrams, reliability graphs, fault trees, attack trees, Markov chains, stochastic petri nets, etc. Another survey of quantitative security models was performed by Verendel [48]. Roudier et al. [40] and Apvrille and Roudier [2] created the modeling language SysML-Sec, which extends SysML with security extensions. Madan et al. [26] used knowledge from fault-tolerance models to derive intrusion-tolerant models. An early attempt to provide a security concept similar to Safety-Integrity Levels (SILs) for ECU classification was done by Nilsson et al. [31]. Attacker or threat agent modeling is a popular form of threat modeling because it identifies attacker capabilities. Tariq et al. [46] adapted a framework which is well known in usability engineering to attacker modeling. Similarly, Sindre and Opdahl [44] extended the notion of use-cases with so called “**misuse cases**”.

2.4 Security metrics

This section will provide a brief discussion of security metrics and how to evaluate security models.

A model in technical literature can have two meanings as nicely stated by [23]:

1. Movement from the general to the particular, when the model is the manifestation of a theory in practice and a ’concrete exemplification of general stated principles’.
2. Movement from the particular to the general by abstraction

Thus, the purpose of a security model is to have a general model that can be used to draw conclusions of a particular system before such a system has even been built. However, to reach that point, it seems that extensive efforts need to be spent on analyzing particular instances of security systems to be able to find a general abstraction of a secure system and security metrics.

Current approaches have mainly tried to target two different areas of security metrics, that of management security and operational security [48], [51]. In management security, standard processes are used for maintaining and establishing policies at the management level and thereby handle security (e.g. [14]). For operational security, the aim is to measure the security of implemented systems that are during operation, hence, the level of threats to which such a system is exposed to and how to implement correct protection mechanisms to divert these threats.

Much effort has already been spent in security metrics of operational security [48]. The problem of defining security metrics, which seems to be very hard, lies in that a secure system is a system without vulnerabilities. Thus, we need to prove the *absence* of security vulnerabilities to have a secure system. Furthermore, when measuring security an independent measure must provide reproducible results, i.e. it needs to come to identical conclusions for the same settings; otherwise the measurement is not measuring the right attribute(s) [48]. Also, with the constantly changing Internet and computers that are patched frequently, a system is far from stationary [48].

Metrics can yield either qualitative or quantitative results. In a qualitative model, the security system reaches a certain level of security, cf. common criteria's EAL [6], [7]. The problem with such evaluation is that analysis of the combinations of such security approaches hardly lead to any useful conclusion — What does it mean to combine an EAL2 with an EAL3 [7] ? In a quantitative approach using a *rational scale*, the addition of two values (hence security approaches) should have a meaningful conclusion. The sum of the height of two persons can be illustrated by one person standing on the shoulder of the other resulting in the total length of them, for instance [23], [50].

In relation to model evaluation, formal verification should be mentioned. The verifiability of models can be qualified in three different categories (also see corresponding definitions in ISO 26262-1):

- Formal Models: full formal verification of the model is possible.
- Semi-Formal Models: only parts of the model are formally verifiable.
- Informal Models: formal verification is not possible.

Many formal models use some kind of model checking, which provides proofs by exhaustive search, i.e. all known branches are expanded and checked for correctness under the given model.

3. Threat modeling and risk assessment – processes, methods and tools

This chapter provides an overview of state-of-the-art frameworks, processes, methodologies and tools for security engineering and security evaluation with focus on threat modeling and risk assessment. First, we present concepts, frameworks and processes for threat analysis and risk assessment. Next, we present a number of threat analysis and risk assessment methodologies. Finally, we briefly discuss about several open-source tools that can potentially be used for security modeling.

3.1 Concepts, frameworks and processes

In this sub-section, we first provide a brief overview of generic system security engineering process. Next, we present several known framework and processes for threat modeling and security evaluation.

3.1.1 Generic system security engineering

Designing system security is best done by utilizing a systematic engineering approach [30]. Systems security engineering is concerned with identifying security risks, requirements and recovery strategies. It involves well defined processes through which designers develop security mechanisms. Ideally, security engineering should be incorporated into the system design process as early as possible, from the initial architecture specification, if possible [30]. Figure 3-1 shows a view of security engineering process as suggested by Myagmar et al [30].

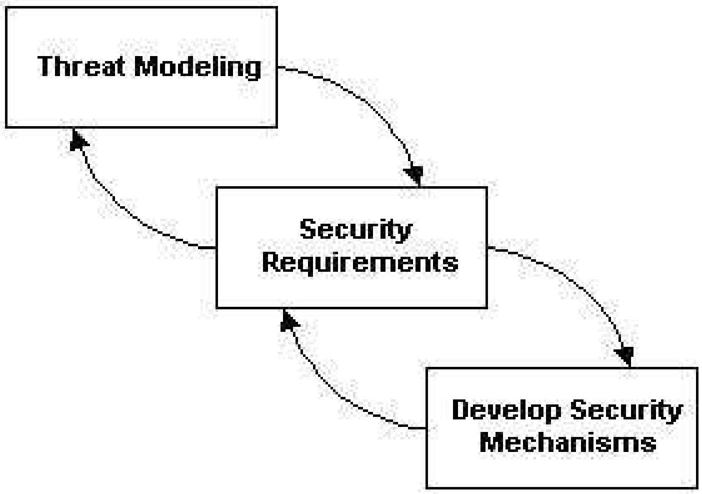


Figure 3-1: System security engineering [30].

A brief overview of the proposed security engineering process is as follows:

- Threat modeling involves (a) characterizing the system, (b) identifying assets and access points, and (c) identifying threats. The system under evaluation can be modeled using either Data Flow Diagram (DFD) or Network Model.
- Specifying security requirements require analyzing the identified threats based on their criticality and likelihood, and a decision is made whether to mitigate the threat or accept the risk associated with the threat.

- Developing security mechanisms follow the outcome of the risk assessment. The threats selected for mitigation must be addressed by some countermeasures.

Once the required security mechanisms are identified, the development of these mechanisms follows traditional product development lifecycle. Each stage of security engineering feeds back to the proceeding stage and through that stage to all earlier stages.

3.1.2 Microsoft − threat modeling and security development lifecycle

Threat modeling is an approach to systematically identify and rate the threats that are most likely to affect the system under evaluation [45]. Also, it should be an iterative process that starts during the early phases of the design of your application and continues throughout the application life cycle [45]. Figure 3-2 shows an overview of threat modeling process proposed by Microsoft [45].

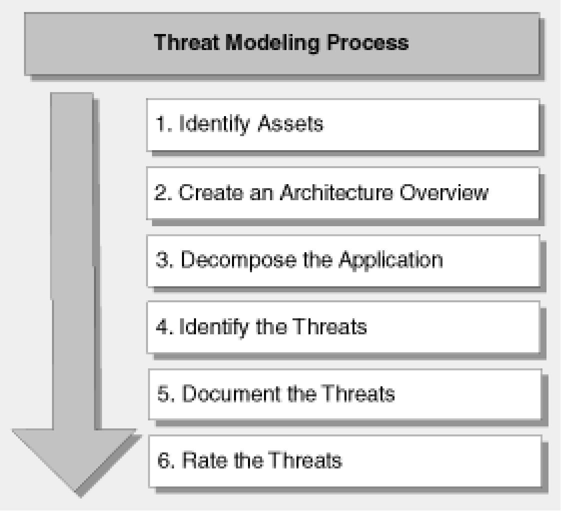


Figure 3-2: An overview of threat modeling process.

The steps of the overall process are as follows:

1. **Identify assets**: Identify the valuable assets that the systems under evaluation must protect.
2. **Create an architecture overview**: Use simple diagrams and tables to document the architecture of the application, including subsystems, trust boundaries, and data flow.
3. **Decompose the application**: Decompose the architecture of the application, including the underlying network and host infrastructure design, to create a security profile for the application. The aim of the security profile is to uncover vulnerabilities in the design, implementation, or deployment configuration of your application.
4. **Identify the threats**: Keeping the goals of an attacker in mind, and with knowledge of the architecture and potential vulnerabilities of the application, identify the threats that could affect the application.
5. **Document the threats**: Document each threat using a common threat template that defines a core set of attributes to capture for each threat.
6. **Rate the threats**: Rate the threats to prioritize and address the most significant threats first. The rating process weighs the probability of the threat against damage that could result should

an attack occur. It might turn out that certain threats do not warrant any action when you compare the risk posed by the threat with the resulting mitigation costs.

The output from the threat modeling process is a document that helps to clearly understand the threats that need to be addressed and how to address them. Threat models consist of a definition of the architecture of the application and a list of threats for the application scenario.

Furthermore, Microsoft proposes **Security Development Lifecycle (SDL)** which is a security assurance process with focus on software development [29].The SDL aims to reduce the number and severity of vulnerabilities in software. The SDL introduces security and privacy throughout all phases of the development process [29]. Figure 3-3 shows the simplified process methodology. Simply put, the Microsoft SDL is a collection of mandatory security activities, presented in the order they should occur and grouped by the phases of the traditional software development life cycle (SDLC) [29].

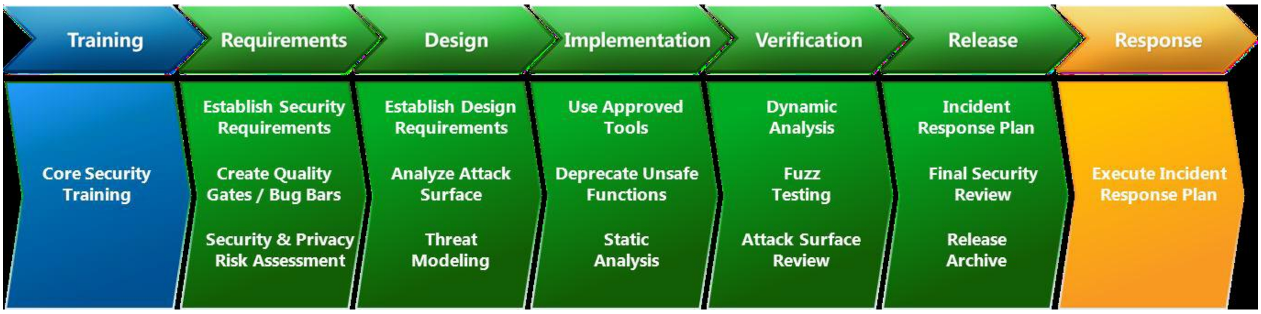


Figure 3-3: The Microsoft Security Development Lifecycle - Simplified.

3.1.3 Trike

Trike acts as a unified conceptual framework for security auditing from a risk management perspective through the generation of threat models, with an associated tool [61]. It is an open source threat modeling methodology and tool. A security auditing team can use it to completely and accurately describe the security characteristics of a system from its high-level architecture to its low-level implementation details [61]. Trike also enables communication among security team members and between security teams and other stakeholders by providing a consistent conceptual framework [61]. Trike uses four specific models:

1. **Requirements Model**
   a. Actors
   b. Assets
   c. Intended Actions
   d. Rules
   e. Actor-Asset-Action Matrix
2. **Implementation Model**
   a. Intended Actions vs. Supporting Operations and the State Machine
   b. Data Flow Diagrams
   c. Use Flows
3. **Threat Model**
   a. Threat Generation

b. Attacks, Attack Trees, and the Attack
c. Weaknesses
d. Vulnerabilities
e. Mitigations
f. Attack Libraries

4. **Risk Model**
   a. Asset Values, Role Risks, Asset-Action Risks, and Threat Exposures
   b. Weakness Probabilities and Mitigations
   c. Vulnerability Probabilities and Exposures
   d. Threat Risks
   e. Using the Risk Model

3.1.4 OCTAVE

OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) is a risk based strategic assessment and planning technique for security [62]. It is mainly known for being self-directed. While most assessments of a system is focused on technology (targeted at technological risk and focused on tactical issues), OCTAVE targets organizational risk and concentrates mainly on strategic, practice-related issues. It also utilizes people not only from the information technology department but also from the operational (business) departments to address the security needs of the organization as a whole. By doing so, OCTAVE assists organizations to balance three key aspects applied to any network infrastructure: operational risk, security practices, and technology.

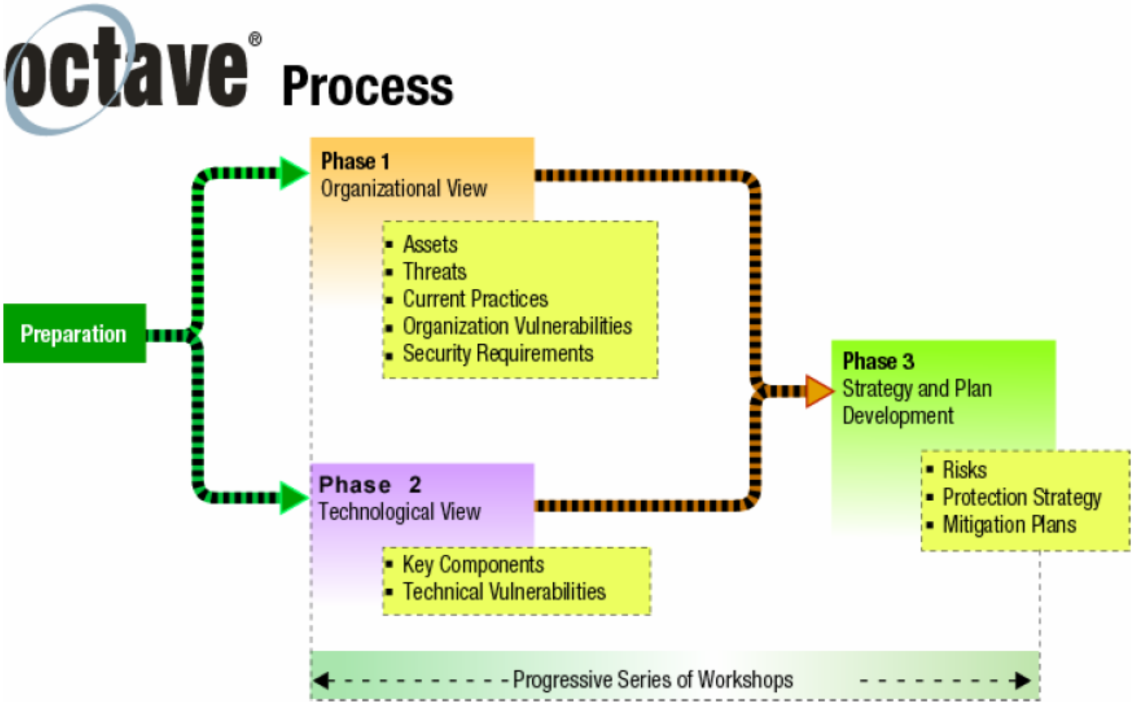


Figure 3-4: OCTAVE phases [62].

OCTAVE is an asset-driven evaluation approach. Analysis teams:

- identify information-related assets (e.g., information and systems) that are important to the organization
- focus risk analysis activities on those assets judged to be most critical to the organization
- consider the relationships among critical assets, the threats to those assets, and vulnerabilities (both organizational and technological) that can expose assets to threats
- evaluate risks in an operational context - how they are used to conduct an organization’s business and how those assets are at risk due to security threats
- create a practice-based protection strategy for organizational improvement as well as risk mitigation plans to reduce the risk to the organization’s critical assets

The organizational, technological, and analysis aspects of an information security risk evaluation are complemented by a three-phased approach. OCTAVE is organized around these three basic aspects (illustrated in Figure 3-4).

3.1.5 Miscellaneous models

A plethora of other frameworks, methodologies and processes are proposed in the literature. In this section, we present a couple of relevant and interesting security analysis approaches.

Wolf and Scheibel [71] refined the ideas by Henniger et al. [72], and also combine existing techniques into a risk rating framework for automotive systems. Our framework has many similarities with Wolf and Scheibel's work, but there are also significant differences. The approach and terminology in [71] is closely aligned with Common Criteria, whereas our approach aims to be compatible with ISO 26262 to ease industry adoption [73]. In addition, we stress the modularity and adaptability of our framework [73]. Another major difference is how threats are identified: Wolf and Scheibel [71] use per-asset security objectives to define attack trees based on security questionnaires, whereas we propose to use STRIDE due to its easier use for non-security experts [73]. Furthermore, in the attack potential calculation (Threat Level) we deviate more from Common Criteria to adapt it for the automotive industry, and we settled on more intuitive names [73]. They also do not consider the privacy and legislative impact [73].

Schmidt et al. [56] proposes a security analysis approach referred to as “**Security In Networked Automotive (SINA)**”. The SINA methodology consists of three major steps:

1. **Data flow modeling:** A system use case must be represented as Data Flow Diagram (DFD). DFDs are hierarchical in such a way that processes can be iteratively refined. SINA recommends starting with a very abstract top-level view. As a result, top-level processes can often be directly formulated from the stakeholders in the use case description.
2. **Threat analysis:** SINA proposes keyword-based threat identification, which is based on the DFD. It enumerates applicable security threats according to specific threat classes. This method is similar to the STRIDE approach, but differs in the threat classification being used.

3. **Risk assessment:** The identification of threats and their potential effects is the basis for the subsequent risk assessment. SINA suggests building attack trees for the most severe threats directly from the DFD. Investigation of the attack tree of a specific security threat reveals the different risk-levels of each attack path and thus gives an overview of the minimum, maximum, or average risk.

The purpose of PASTA (Process for Attack Simulation and Threat Analysis) is to provide a process for simulating attacks to applications, analyzing cyber threats that originate them and mitigate cybercrime risks that these attacks and threats pose to organizations. PASTA consists of a seven stage process for simulating attacks and analyzing threats to an application environment with the objective of minimizing risk and associated impact to the business. By following this process, businesses can determine the adequate level of countermeasures that can be deployed to mitigate the risk from cyber threats and attacks to applications. Figure 3-5 shows PASTA model [63] of threat and risk analysis.

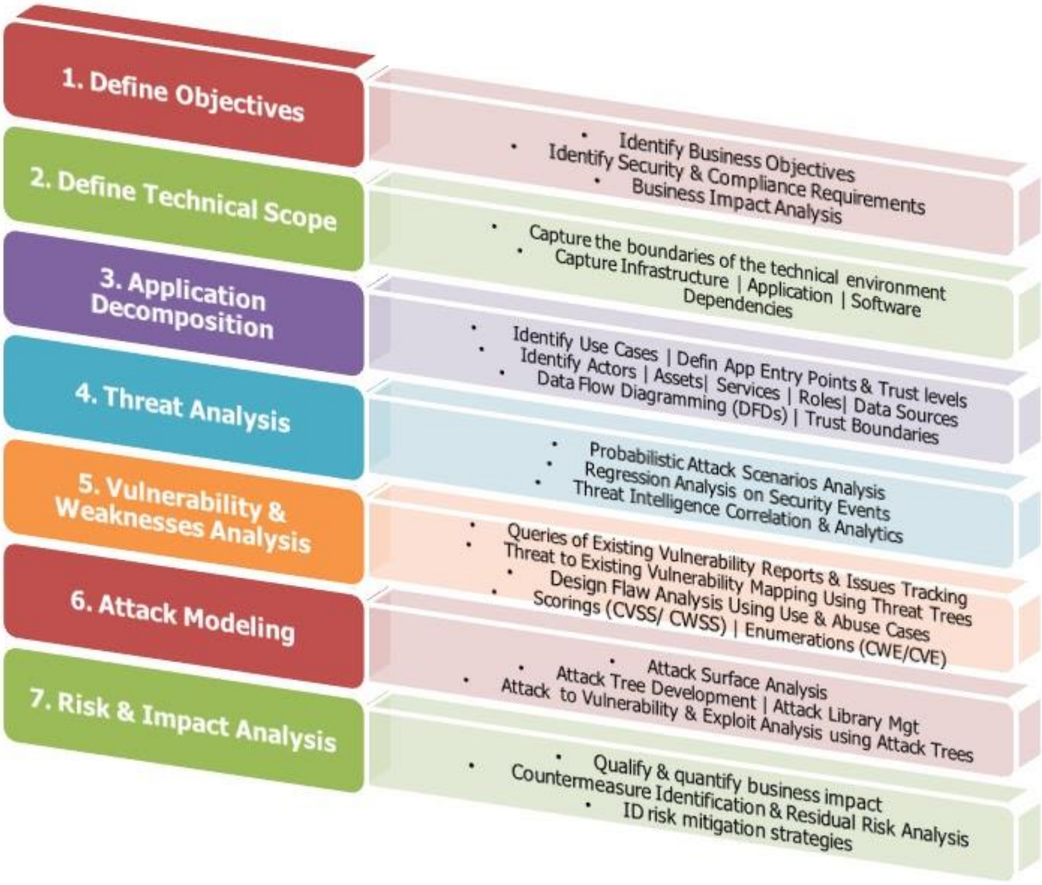

1. Define Objectives
• Identify Business Objectives
• Identify Security & Compliance Requirements
• Business Impact Analysis
2. Define Technical Scope
• Capture the boundaries of the technical environment
• Capture Infrastructure | Application | Software Dependencies
3. Application Decomposition
• Identify Use Cases | Defin App Entry Points & Trust levels
• Identify Actors | Assets| Services | Roles| Data Sources
• Data Flow Diagramming (DFDs) | Trust Boundaries
4. Threat Analysis
• Probabilistic Attack Scenarios Analysis
• Regression Analysis on Security Events
• Threat Intelligence Correlation & Analytics
5. Vulnerability & Weaknesses Analysis
• Queries of Existing Vulnerability Reports & Issues Tracking
• Threat to Existing Vulnerability Mapping Using Threat Trees
• Design Flaw Analysis Using Use & Abuse Cases
• Scorings (CVSS/ CWSS) | Enumerations (CWE/CVE)
6. Attack Modeling
• Attack Surface Analysis
• Attack Tree Development | Attack Library Mgt
• Attack to Vulnerability & Exploit Analysis using Attack Trees
7. Risk & Impact Analysis
• Qualify & quantify business impact
• Countermeasure Identification & Residual Risk Analysis
• ID risk mitigation strategies

Figure 3-5: P.A.S.T.A. model of threat and risk analysis.

3.2 Methods and models for threat analysis and risk assessment

A threat model in general describes security aspects with respect to a particular system under investigation by associating a set of potential vulnerabilities and threats with a potential set of assets. The output of threat identification process is a threat profile for a system, describing all the potential attacks, each of which needs to be mitigated or accepted [30]. Risk assessment are followed by threat analysis during security system engineering to identify whether to map each threat either into a countermeasure (mitigation mechanism) or an assumption that it is not worth worrying about [30].

3.2.1 CIA model

One approach of threat analysis is to apply the well-known CIA model (Confidentiality, Integrity, Availability) as threats to a particular asset can lead to the violation of the security attributes (Confidentiality, Integrity, Availability) associated with the asset.

Table 3-1 shows one example of using the CIA for threat analysis. Security requirement with respect to each row of the table (asset, threat and security attribute) can be derived. This approach can be considered as asset-centric as threats and security attributes are identified with respect to the each identified asset. Each asset may be associated with multiple threats and each threat may be associated with multiple security attributes.

Table 3-1: Threat analysis based on the CIA model.

Asset	Security Attribute as per CIA	Threat
In-Vehicle Network	Availability	Denial of Service – flooding, jamming
ECU Software	Integrity	Tampering – improper software modification
Log data	Confidentiality	Reading sensitive data using interface, e.g., OBD

3.2.2 Microsoft – STRIDE and DREAD

Microsoft proposes STRIDE model to identify threats [45]. In the STRIDE model, threats can be categorized based on the goals and purposes of the attacks. A working knowledge of these categories of threats can help to organize a security strategy to have planned responses to threats [45]. The term STRIDE stems from the initial letters of different possible threats:

1. **S**poofing - attackers pretend to be someone or something else
2. **T**ampering - attackers change data in transit or in a data store
3. **R**epudiation - attackers perform actions that cannot be traced back to them
4. **I**nformation disclosure - attackers get access to data in transit or in a data store
5. **D**enial of service - attackers interrupt a system’s legitimate operation
6. **E**levation of privilege - attackers perform actions they are not authorized to perform

STRIDE model can be used as shown in Table 3-2 to combine threats, assets and security attributes. This model can be considered as threat-centric or attacker-centric since each threat is associated with a particular asset from attacker perspective. One advantage of STRIDE model is that this changes focus from the identification of every specific attack to focusing on the end results of possible attacks. Moreover, STRIDE model actually extends the original CIA model by correlating threats with security attributes (authenticity, integrity, non-repudiation, confidentiality, availability, and authorization).

Table 3-2: Threat analysis based on STRIDE.

STRIDE threat	Asset	Security attribute
Spoofing	Sensor	Authenticity
Tampering	ECU Software	Integrity
Repudiation	Encryption/cryptographic keys	Non-repudiation
Information Disclosure	Log data	Confidentiality
Denial of Service	In-Vehicle Network	Availability
Elevation of Privilege	Features (enabling/disabling)	Authorization

Once the threats associated with assets and security attributes are identified using STRIDE or alternative model, the next step is to perform risk assessment to prioritize the risks associated with the threats. DREAD is a classification scheme for quantifying, comparing and prioritizing the amount of risk presented by each evaluated threat [45]. The DREAD acronym is formed from the first letter of each category below:

1. **D**amage potential - which assets are affected? Is damage with serious consequences possible?
2. **R**eproducibility - easiness of bringing the attack about, are there technical or time critical constraints?
3. **E**xploitability - is the threat valuable, are there many attackers with different objectives?
4. **A**ffected users - how many installations are subject to the attack: 5% or 90%?
5. **D**iscoverability - is the attack path discoverable through logical thought or by luck?

DREAD modeling influences the thinking behind setting the risk rating, and is also used directly to sort the risks. The DREAD algorithm, shown below, is used to compute a risk value, which is an average of all five categories [45]. The calculation always produces a number between 0 and 10; the higher the number, the more serious the risk. Each category is assigned a value between 0 and 10.

Risk_DREAD = (DAMAGE + REPRODUCIBILITY + EXPLOITABILITY + AFFECTED USERS + DISCOVERABILITY) / 5;

3.2.3 CVSS model

Common Vulnerability Scoring System (CVSS) (Finalized in 2007, Version 2) is known as being a specification for measuring the relative severity of software vulnerabilities and can be applied to a plethora of systems including those that belong to federal agencies in the United States [27] [28].

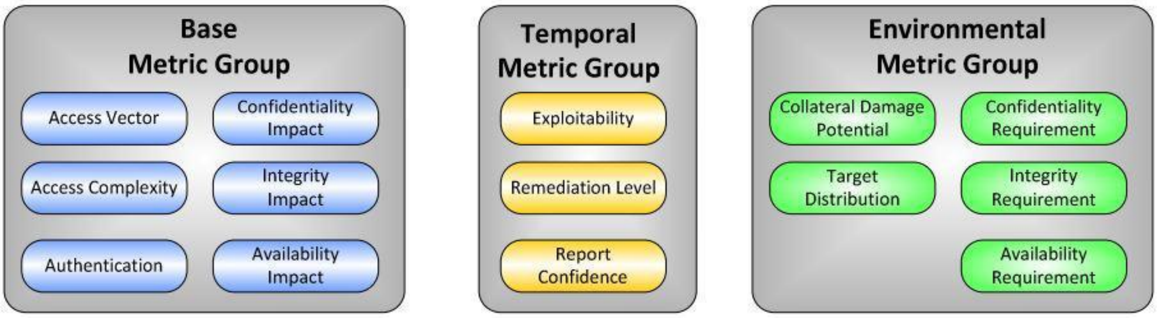


Figure 3-6: Metric groups of the CVSS [28].

CVSS is comprised of three different metric groups: Base, Temporal, and Environmental. Each one consists of their own set of metrics as shown in Figure 3-6. The base metric group can be used for most situations, but other values can be assigned to the other metric groups in order to provide additional context for a specific vulnerability. These metric groups can be described as follows:

1. **Base** represents the characteristics of a vulnerability that are constant over time and user environments
2. **Temporal** represents characteristics of a vulnerability over time but makes no mention of the user environments
3. **Environmental** represents characteristics of a vulnerability that are relevant and/or unique to a user's particular environment

Once each of these base metrics is assigned a value, the base equation calculates a score that ranges from 0 to 10. From all the factors, a vector is created from the equation and the vector is outputted as a string of text that contains values assigned to each metric in order to communicate exactly how the score, for each vulnerability found, is derived. The process is shown in Figure 3-7.

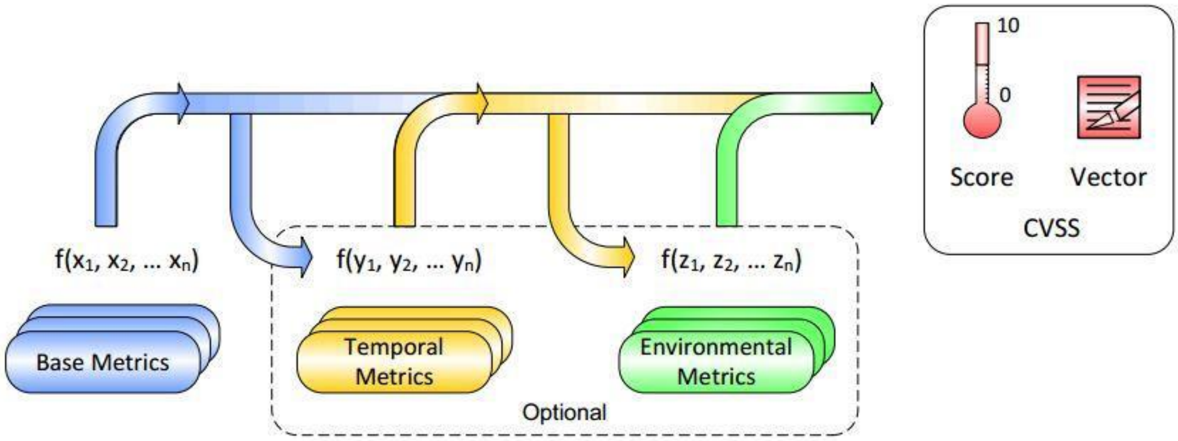


Figure 3-7: Metrics and equations of the CVSS being combined to create vector [27].

3.2.4 OWASP model

The Open Web Application Security Project (OWASP) is a worldwide not-for-profit charitable organization focused on improving the security of software. OWASP proposes a risk rating methodology that is based on the standard risk model: *Risk = Likelihood * Impact*. The OWASP methodology consists of six steps that include the factors that make up the "likelihood" and "impact" components. The six steps are as follows [33]:

Step 1: Identify Risk

Step 2: Factors for estimating likelihood

- Threat Agent Factors: Skill level, motive, opportunity, size
- Vulnerability Factors: Ease of discovery, ease of exploit, awareness, intrusion detection

Step 3: Factors for estimating impact

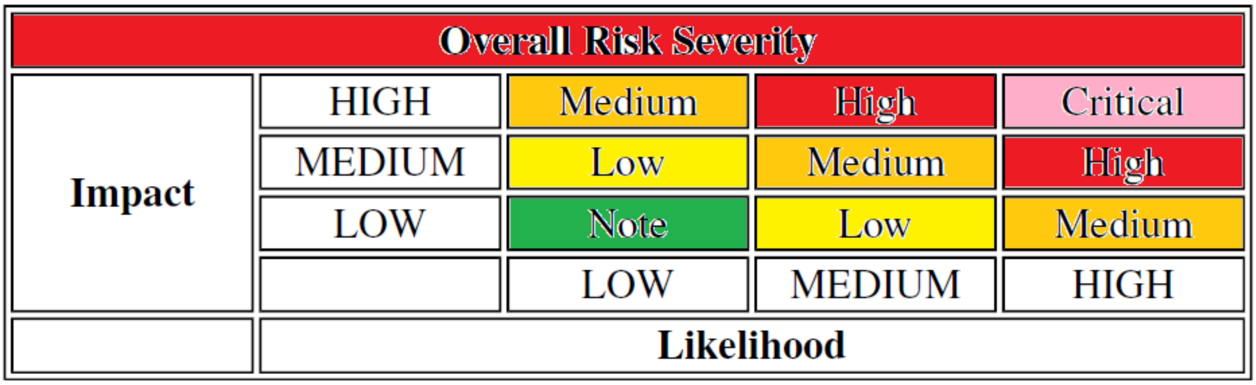
- Technical Impact Factors: Loss of confidentiality, integrity, availability and accountability
- Business Impact Factors: Financial damage, reputation damage, non-compliance, privacy violation

Step 4: Determining severity of risk

Step 5: Deciding what to fix

Step 6: Customizing your risk rating model

Figure 3-8 shows how OWASP methodology provides overall risk severity by combining likelihood and impact. The detailed information is available in the OWASP methodology [33].



Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Figure 3-8: Risk severity based on the OWASP methodology.

3.2.5 EVITA model

The EVITA project [11] has proposed a model of performing threat and risk analysis. As per this model, to assess the “risk” associated with an attack it is necessary to assess the “severity” of the possible outcome for the stakeholders, and the “probability” that such an attack can be successfully mounted. Hence, this model adopts attacker-centric approach of risk analysis. EVITA identifies four high-level security objectives: (a) operational, (b) privacy, (c) financial, and (d) safety.

To perform risk assessment by considering safety as security objective, EVITA adopts the ASIL determination approach as proposed in ISO 26262 [20]. The proposed model derives security risk graph for safety-related security threats by augmenting the notions of “controllability” and “severity” with “combined attack probability”. Generally speaking, “combined attack probability” used in EVITA is analogous to “exposure” as used in ISO 26262 [20] for ASIL determination. However, to perform risk assessment for the remaining security objectives (privacy, operational and financial) and derive security risk graph, only “severity” and “combined attack probability” are considered whereas “controllability” is excluded. A more detailed description of this process can be found in Appendix C – Threat and risk analysis of the deliverable D2.3 [53] of the EVITA project.

3.2.6 Threat, Vulnerability, and Risk Analysis (TVRA)

European Telecommunications Standards Institute (ETSI) proposes Threat, Vulnerability, and Risk Analysis (TVRA) that is an assessment method originally developed for their standards developers to analyze security in telecommunication systems [10]. The method has already been applied in the vehicular setting, notably while deriving the new ITS standard platform for European Vehicle-to-Vehicle (V2V) communication [9].

The TVRA method [10] can briefly be summarized as follows (see Figure 3-9); The TOE is identified and the assets within are described together with the goals of the evaluation. *Security objectives* are then identified and classified based on the five security attributes: confidentiality, integrity, availability, authenticity, and accountability (CIAAA). These security objectives are then used to derive the *functional security requirements*. Functional security requirements are more detailed requirements than the security objectives, e.g., authentication should be implemented by means of passwords. Then, an *inventory of assets* is done. Possible *vulnerabilities* are then identified and classified together with their *corresponding threats* and their unwanted outcome. These threats are classified based on the following four categories: interception, manipulation, denial of service, and repudiation. *Risks* are then calculated depending on the *likelihood* of these threats and their unwanted outcome. Finally, a set of *countermeasures* are derived and a *cost-benefit analysis* is performed to select the most suitable ones to *reduce the risks* of the identified threats. These results are then used to design the *security services*.

Note that the new security services may add new assets to the TOE, thus new analyses are needed to also ensure the security of them. A database structure is provided to help security analysts to collect all data during the assessment [10]. The use of an expert for the assessment is required as the standard only gives the steps to be performed, not how to perform them.

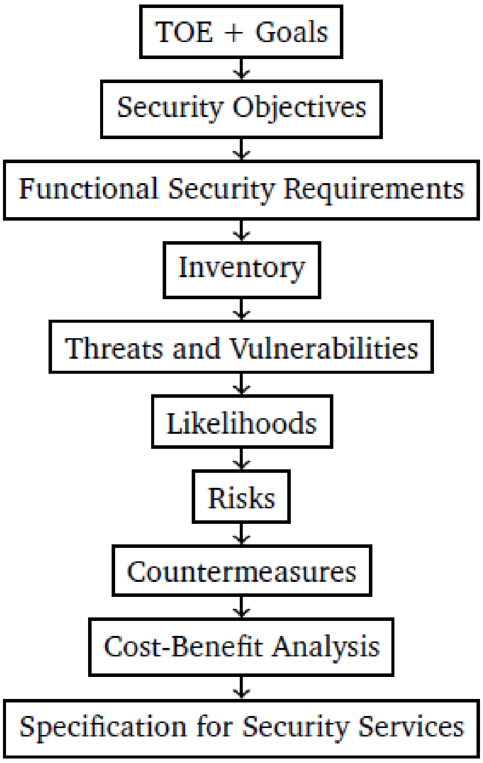

TOE + Goals
Security Objectives
Functional Security Requirements
Inventory
Threats and Vulnerabilities
Likelihoods
Risks
Countermeasures
Cost-Benefit Analysis
Specification for Security Services

Figure 3-9: Overview of the TVRA analysis methods.

3.2.7 SECTRA model

The SECTRA model [64] is an asset centric model as it defines required strength levels for security mechanisms needed to protect assets. The characteristics of the environment are defined by organizations that host threat agents. A threat agent has a motivation to develop or use attacks targeting one or more of the assets of the target system. The level of protection needed for an asset is defined by the ability to develop attacks, the motivation for attacks and the severity of a successful attack. The level of protection is called the required strength level and is defined individually for each of the security relevant assets within the system boundary. Strength levels are mapped to different kind of security mechanisms depending of the asset and type of protection needed. Security mechanisms may also need to be protected and this typically cases an iteration of the modeling. The steps involved to define the strength level for an asset is:

1. Determination of the **attack potential**. Factors that are taken into consideration are expertise, availability of resources and knowledge of the target with respect to identified threat agents and their associated organization.
2. Definition of **opportunity** to perform attacks. This is the level of exposure of the asset in terms of physical or logical access and the time of exposure.
3. The **threat level** depends of the attack potential and the level of opportunity to perform attacks. There are defined threat levels for each threat agent for each of the threats.
4. The **impact level** with respect to safety, privacy, operation and financial aspects are considered for each of the identified use cases. All security relevant assets for the use case are identified and corresponding need for protection (CIA) is identified.
5. The **motivation** parameter to perform attacks, the impact level and the threat level is combined to define the required strength level. All assets are assigned a required strength level.
6. **Security mechanisms** (named protection mechanisms in Figure 3-10) with respect to CIA are mapped to the assets at corresponding strength levels.

Figure 3-10 shows a more detailed view of the workflow using the SECTRA model.

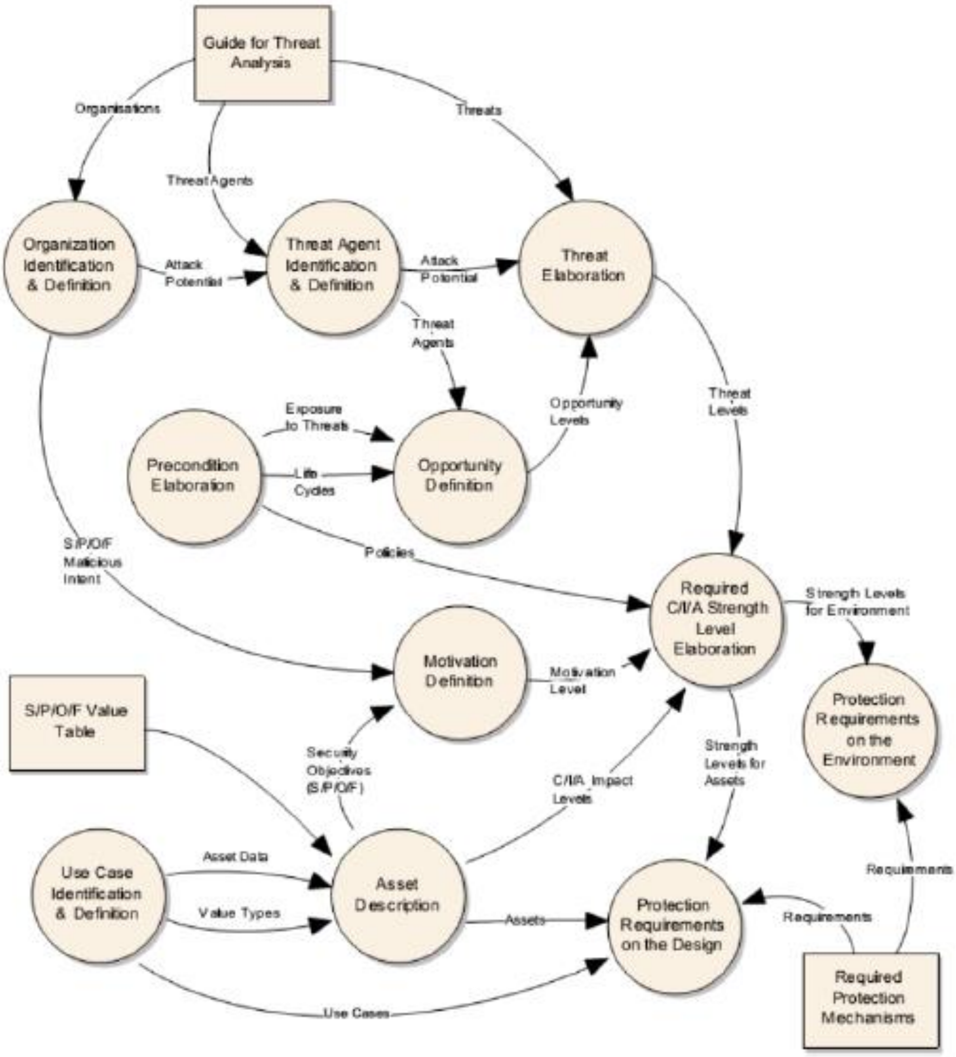


Figure 3-10: The SECTRA security model.

3.3 Tool support

Microsoft provides a threat modeling tool kit [38] to realize STRIDE threat model. The tool (Version 3.1.8) is freely available and has dependency on Microsoft Visio. A view of the tool is shown in Figure 3-11. The tool uses a simple drag and drop action in order to build the DFD of a specific use case. There are six main objects available for use to make a model: a process, multiple processes, an external interactor, a data store, data flow and trust boundaries (see Figure 3-12).

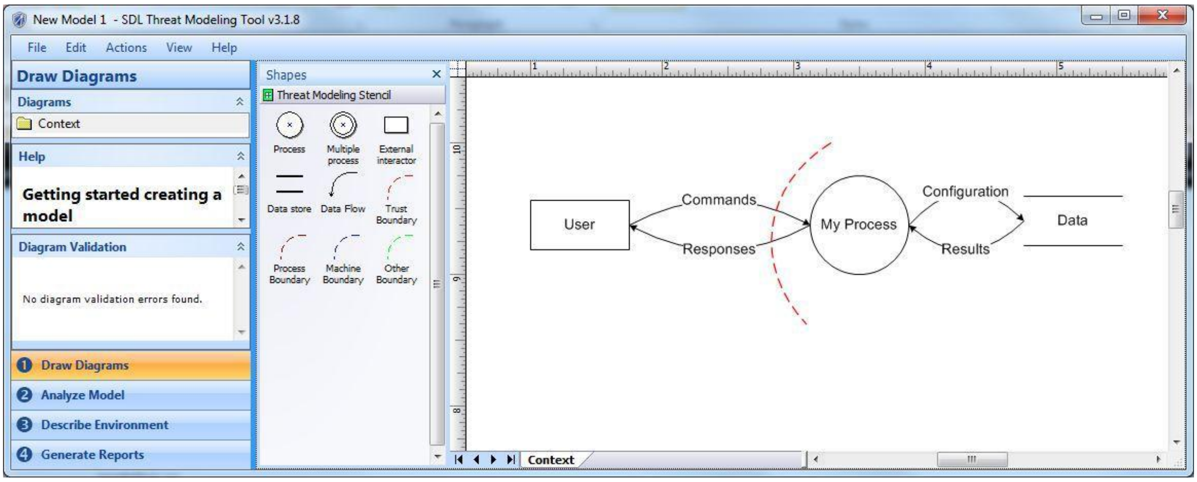


Figure 3-11: A view of the Microsoft Threat Modeling Tool.

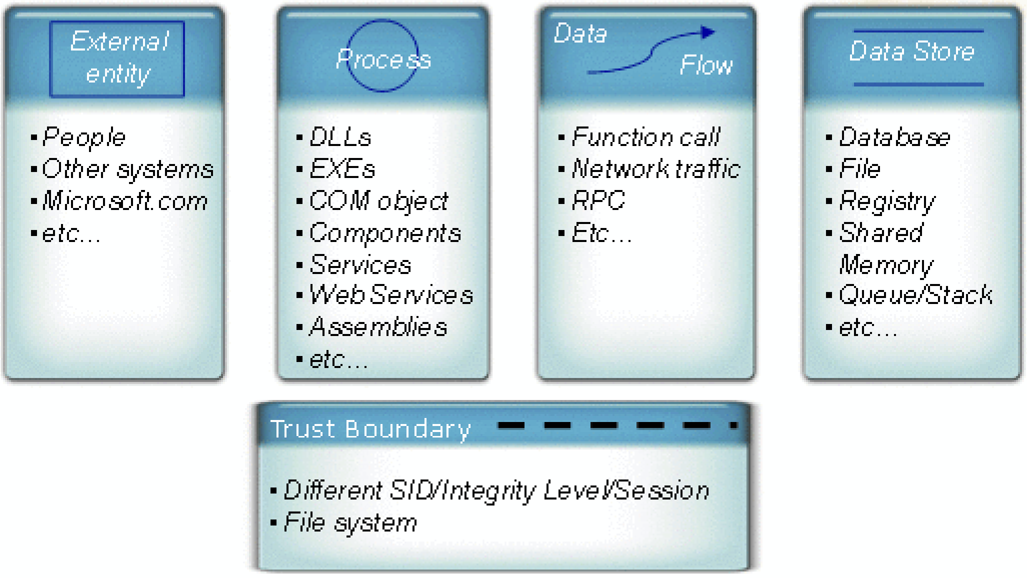


Figure 3-12: Objects of the Microsoft Threat Modeling Tool.

Once the DFD is complete, the tool can automatically validate the DFD and analyze it to identify the relevant threats for the DFD of the system under evaluation. The SDL tool also allows for multiple layers, depending on how in-depth the designer and security experts wish to go into a particular system. The context diagram is the highest level in which we view the entire component, product or system. From there, it is possible to go down into multiple levels to produce more detailed view.

Trike framework [61] provides tool support to realize the proposed conceptual framework. There are two implementations of Trike: one is a spreadsheet, and the other is a standalone desktop tool. The publicly available spreadsheet is most compatible with Excel 2011 and other versions of Excel. All versions of the spreadsheet are using Trike methodology version 1.5, which is an interim methodology bridging the very large gap between version 1 and version 2. The standalone desktop tool is written in Smalltalk, specifically Squeak. It implements methodology version 1, which is now fairly outdated.

SeaMonster is a security modeling tool [65]. The purpose of creating SeaMonster was to develop a free, common platform for modeling security that can be used by security experts as well as developers and that will facilitate reuse of models. SeaMonster is based on Eclipse, which is basically an application platform where a very large set of plugins can be added to suit the needs of the users. The three main Eclipse plugin frameworks SeaMonster benefits from are the Graphical Modeling Framework (GMF), the Eclipse Modeling Framework (EMF) and the Graphical Editing Framework (GEF). SeaMonster currently supports the following models:

- Misuse cases (what are the main threats to the system)
- Attack trees (how can the system be attacked)
- Security Activity Graphs (describes how to perform a security activity, notation to be updated)
- Security model (experimental notation for connecting various security model diagrams)

The GOAT modeling tool is one of the results of the SHIELDS EU Project [65]. GOAT is a cross-platform standalone modeling tool supporting vulnerability cause graphs and security activity graphs. GOAT consists of a framework application and a set of plugins. Each plugin can provide model types, editors, user interface elements, and import/export functionality.

4. HEAVENS security model

4.1 Background

HEAVENS project aims at identifying owners, assets, risks, vulnerabilities, countermeasures, threat agents and threats as shown in Figure 4-1, and putting all the aspects together to establish security model for the automotive industry. Also, we aim at investigating security models from other domains (e.g., IT security, telecommunications, and defense) in the context of the automotive E/E systems. Accordingly, in the HEAVENS security model, the first step is to identify the use cases from stakeholders (i.e., owners) perspective. This is described in HEAVENS Deliverable D1.1 Needs and requirements [16]. The use cases have then been used to identify assets and threats, including the security attributes that can be affected by a particular threat. During the risk assessment, the roles of threat agents (attackers) as well as the impact of an attack on a particular asset have been considered to rate the threats, i.e., identify the risks. This then facilitates identifying security requirements and possible countermeasures to address a particular threat for a particular asset.

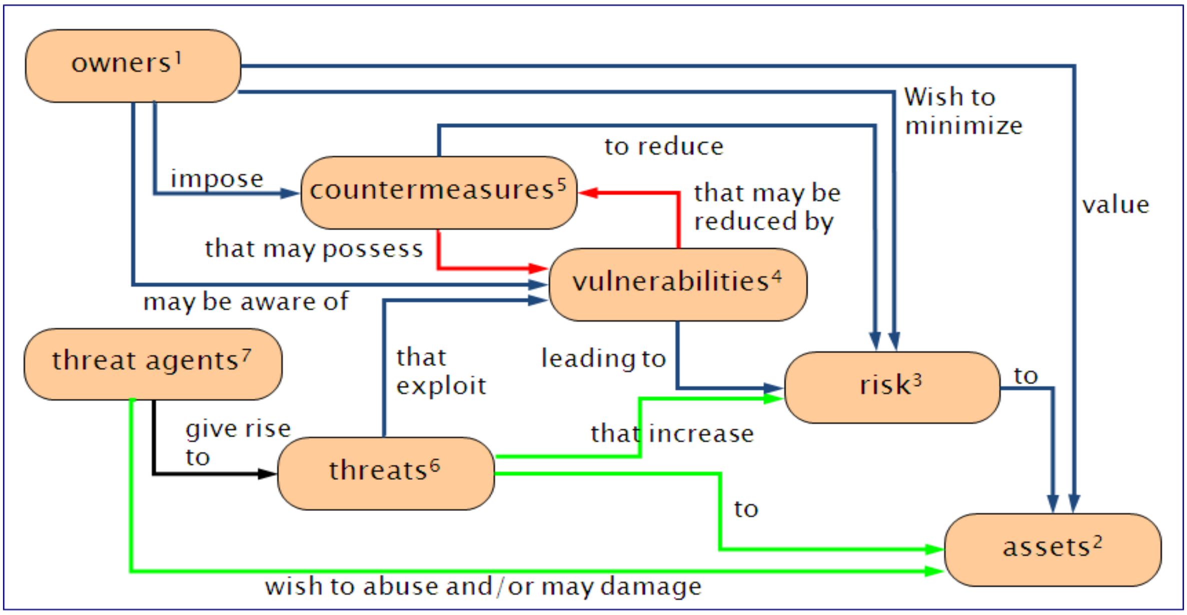


Figure 4-1: Basis for threat analysis and risk assessment [15].

4.2 Motivation, objective and scope

In this sub-section, we first review the existing threat analysis and risk assessment methodologies and establish rationale for developing HEAVENS security model. Then, we present the objectives and limitations of the HEAVENS security model. Finally, we present the workflow of the model.

4.2.1 Why another security model

Several security models encompassing methods and processes for threat analysis and risk assessment have already been proposed [e.g., Common Criteria, TVRA, Microsoft, SECTRA, OWASP, CVSS, OCTAVE] and these establish the state-of-the-art in the field of security evaluation. Standards include Common Criteria [5] for security evaluation in the IT industry. CVSS [28] provides an open and standardized method for rating IT vulnerabilities. OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) [62] is a suite of tools, techniques, and methods for risk-based information security strategic assessment and planning. OCTAVE is an approach used to assess an organization's information security needs [62]. ETSI [10] provides threat, vulnerability and risk analysis methodology to deal with security issues in the telecommunications industry. SECTRA [64] proposes a security model for the defense industry. OWASP (Open Web Application Security Project) [33] [34] aims at improving software security and provides risk rating methodology which is primarily suitable for web application security. However, none of these is readily applicable to the automotive E/E systems and fulfills the requirements of the automotive industry. However, several aspects of the aforementioned methodologies can potentially be customized and modified to meet the requirements of the automotive E/E systems. Accordingly, we have considered the state-of-the-art threat analysis and risk assessment methodologies while developing the HEAVENS security model.

Microsoft suggests STRIDE methodology [45] for threat analysis along with tool support and this mainly targets software system. Furthermore, Microsoft suggests DREAD methodology for risk assessment. While STRIDE approach can potentially be used to identify threats in the automotive E/E systems, DREAD model cannot be considered as a potential candidate for risk assessment in the context of the automotive industry. The ratings suggested by DREAD risk assessment model are not consistent and are highly subjective. As a result, Microsoft abandoned the model in 2008. Hence, we consider STRIDE but not DREAD in the HEAVENS security model.

Finally, the EVITA project [11] has proposed a risk rating methodology [53] for the automotive E/E systems. This indeed is the pioneering risk rating approach for the automotive industry and is well-known in the automotive cyber-security community. As a result, we have closely looked into the EVITA methodology while deriving the HEAVENS security model. In the EVITA approach, the parameters that are used to estimate attack potential is inspired by Common Criteria [8]. However, the usage of a couple of parameters (elapsed time and opportunity) suffers from a few anomalies. For example, EVITA does not distinguish among various access types (e.g., physical, logical) while rating opportunity. EVITA considers four parameters for risk rating – safety, financial, operational and privacy. The safety component is well defined and aligned with ISO 26262. Conversely, the other three components do not provide a suitable guideline to effectively evaluate the impact of those on the overall risk rating. For example, to estimate financial damages, EVITA suggests as follows: low-level loss (~10Euro), moderate loss (~100Euro) and heavy loss (~1000Euro). Furthermore, the EVITA approach does not take legislation aspects into account for risk rating. However, several legislations related to environment and driver behavior are already enforced for the commercial vehicles and there exist threats that can potentially lead to the violation of the legislations. The EVITA approach seems more focused around the views of an owner of a passenger car and considers a limited number of stakeholders. For examples, EVITA does not consider fleet owner as stakeholder as well as does not distinguish between vehicle owner and driver. Last but not the least, the EVITA approach is attack-centric and focuses on identifying all possible attacks against a TOE although attacks are dynamic and practically, possible attacks and attack techniques are virtually endless.

Based on the findings of our review of the state-of-the-art threat analysis and risk assessment methodologies, we have proposed the HEAVENS security model to address the limitations of the existing methodologies. The advantages of the HEAVENS security model are as follows:

- The proposed model is equally applicable to a wide range of road vehicles, for example, passenger cars and commercial vehicles. The model considers a wide range of stakeholders (e.g., OEM, Fleet owner, Vehicle owner, Driver, Passenger, etc.).
- Threat centric model realized by applying Microsoft's STRIDE approach in the context of the automotive E/E systems. It supports better understanding of the effects of possible attacks by using only a handful of generic threat categories instead of thinking about virtually endless possibilities of attacks and attack techniques that can be related to an asset.
- The model establishes a direct mapping between security attributes and threats during threat analysis. This facilitates visualizing and making early estimation of the technical impact (confidentiality, integrity, availability) of a particular threat on a particular asset.
- The model maps security objectives (safety, financial, operational, privacy and legislations) with impact level estimation during risk assessment. This assists in understanding the potential business impacts of a particular threat for the relevant stakeholders, for example, OEMs.
- The model provides estimation of impact level parameters (safety, operational, financial, privacy and legislation) based on industry standards. For example, the safety parameter is aligned with the ISO 26262 [19], financial parameter is based on the BSI-Standard [59], and operational parameter is based on the Failure Mode and Effect Analysis (FMEA) proposed by the Automotive Industry Action Group (AIAG) [60].
- The model is aligned with well-established industry standards and initiatives. For example, Common Criteria, ISO 26262, Car 2 Car Communication Consortium. This facilitates reusing the processes that are already in place for another field of studies, for example, functional safety. It also provides an opportunity of understanding the cyber-security issues across safety and security domains.
- The model defines a systematic approach of deriving security requirements by connecting asset, threat, security level and security attribute.

4.2.2 Objective and limitation

The primary objective of the HEAVENS security model (as presented in this deliverable) is to derive security requirements for the TOE, i.e., the assets of the TOE, similar to the notion of functional safety requirements as described in the ISO 26262. To achieve this, we establish a security level for each of the identified threats in relation to the assets constituting the TOE. The HEAVENS security model thus includes both threat analysis and risk assessment. Hence, the "**HEAVENS security model**" refers to threat analysis and risk assessment to facilitate deriving security requirements for a particular TOE by applying the HEAVENS methodology and tool support.

The current version (Version 2.0) of the HEAVENS security model does not suggest countermeasures or security mechanisms to assist in fulfilling the derived security requirements. Also, the model currently does not establish explicit relationship between threats and vulnerabilities.

4.2.3 Workflow

Figure 4-2 shows the workflow of the HEAVENS security model. It consists of three components – threat analysis, risk assessment and security requirements.

- Threat analysis – Description of the functional use cases (In_01 in the figure) is the input to the threat analysis process. Threat analysis produces two outputs: (a) a mapping between threats and assets (Out_01 in the figure) for each asset in the context of the use case, and (b) a mapping between threats and security attributes (Out_02 in the figure) to establish which security attributes are affected due to a particular threat in the context of an asset.
- Risk assessment – Once the threats for the relevant assets are identified, the next step is to rank the threats. This is what is done during risk assessment. The mapping between threats and assets are used as input along with threat level (TL) (In_03 in the figure) and impact level (IL) (In_04 in the figure) parameters. Threat level parameters are presented in Section 4.4.1 (Threat Level (TL)) and impact level parameters are presented in Section 4.4.2 (Impact Level (IL)). As an end result of risk assessment, we identify security level (Out_03 in the figure) for each threat associated with each asset of the TOE/use case.
- Security requirements – Finally, we consider both the mapping between threat and asset (Out_02 in the figure as a result of threat analysis) as well as security level (Out_03 in the figure as a result of risk assessment) to formulate security requirements for the asset and the TOE. Security requirement is a function of asset, threat, security level and security attribute. Note that security level considers the potential business impact in terms of security objectives of a particular threat associated with a particular asset. The derived security requirements are at the level of the functional safety requirements of the ISO 26262 and belong to the concept phase. Later, during product development phase, software security requirements and hardware security requirements need to be derived based on the high-level security requirements.

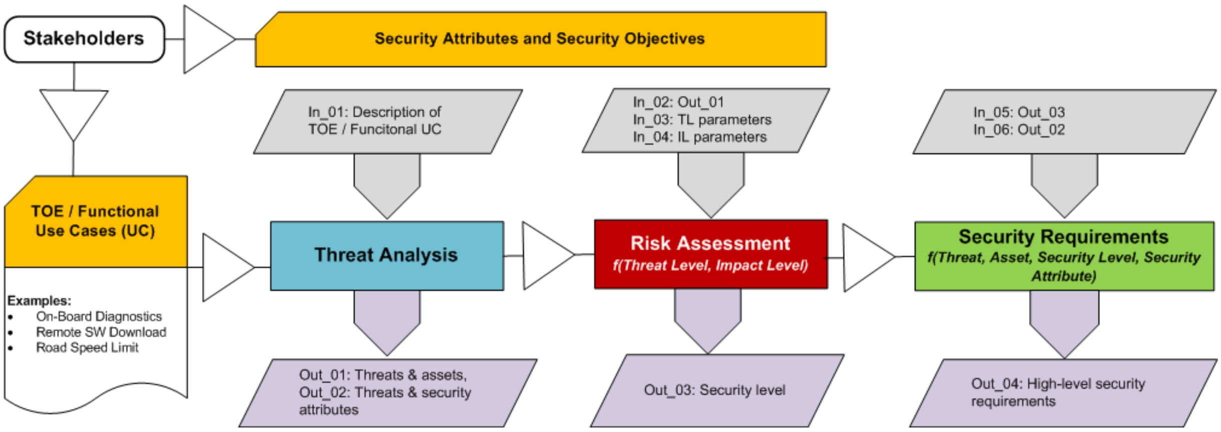

Stakeholders
Security Attributes and Security Objectives
In_01: Description of TOE / Funcitonal UC
In_02: Out_01
In_03: TL parameters
In_04: IL parameters
In_05: Out_03
In_06: Out_02
TOE / Functional Use Cases (UC)
Examples:
On-Board Diagnostics
Remote SW Download
Road Speed Limit
Threat Analysis
Risk Assessment
f(Threat Level, Impact Level)
Security Requirements
f(Threat, Asset, Security Level, Security Attribute)
Out_01: Threats & assets,
Out_02: Threats & security attributes
Out_03: Security level
Out_04: High-level security requirements

Figure 4-2: Workflow of the HEAVENS security model.

4.3 Threat analysis

In the HEAVENS security model, threat analysis refers to the identification of the threats associated with the assets of the TOE and/or the use case under evaluation, and mapping of the threats with the security attributes. We have adopted Microsoft's STRIDE approach [45] for threat analysis. While STRIDE is a structured and qualitative security approach for discovery and enumeration of threats present in a software system [56], we extend the applicability of the STRIDE approach to the automotive E/E systems.

4.3.1 Why STRIDE

STRIDE can be considered as threat-centric or attacker-centric approach since each threat is associated with a particular asset from attacker's perspective. One advantage of such an approach is that this changes focus from the identification of every specific attack to focusing on the end results of possible attacks [45]. The reason is that it is useful to think about threats in terms of what the attacker is trying to achieve instead of thinking about virtually endless variations of specific attacks and attack techniques. This has inspired us to adopt the STRIDE approach in the HEAVENS project.

We have analyzed a large number of threats associated with different assets as presented in the deliverables of the EVITA [53] and the PRESERVE [39] projects, and observed that all those threats can actually be represented using STRIDE as shown in Table 4-1. The threats that are presented in EVITA and PRESERVE are more "specific" (e.g., insert fake data, corrupt data or code, corrupt messages) with respect to an asset whereas the STRIDE threats are relatively "generic" (e.g., tampering). In this way, STRIDE provides a structured approach of categorizing the threats into a small number of groups which makes it easier to understand and apply. STRIDE can be seen as a structured keyword-based method [56]. It has been argued that STRIDE addresses the issues of completeness and repeatability, whereas other threat modeling activities are organized around unstructured brainstorming sessions and informal group discussions [56]. Furthermore, STRIDE suggests that it is possible to associate a set of countermeasure techniques with respect to each threat category to reduce potential risks [56]. Consequently, STRIDE approach appears to be more suitable and applicable in the industrial context.

Table 4-1: Mapping of specific threats to STRIDE.

STRIDE threats	Example of threats from EVITA and PRESERVE
Spoofing	Fake messages
Tampering	Corrupt data or code, Malware flashed, Alter, Inject, Corrupt messages, Manipulate, Configuration change
Repudiation	Replay, Fake messages, Repudiation of message transmission and receipt
Information disclosure	Listen, Intercept, Eavesdropping, Illegal acquisition
Denial of service	Disable, Denial of service, Jamming
Elevation of privilege	Illegal acquisition (keys, certificates, back-end databases), Gain root access

4.3.2 STRIDE and security attributes

STRIDE provides the opportunity of extending the original CIA model by correlating threats with security attributes (authenticity, integrity, non-repudiation, confidentiality, availability, freshness and authorization). We have mapped each category of the STRIDE threats to a set of security attributes. This mapping is static and is used to formulate security requirements as soon as the security level of a particular threat-asset pair is determined during the risk assessment. The mapping between the STRIDE threats and the security attributes is shown below (Table 4-2).

Table 4-2: Mapping between STRIDE threats and security attributes.

STRIDE threats	Explanation	Security attribute
Spoofing	attackers pretend to be someone or something else	Authenticity, *Freshness*
Tampering	attackers change data in transit or in a data store, attackers may change functions as well – implemented in software, firmware or hardware	Integrity
Repudiation	attackers perform actions that cannot be traced back to them	Non-repudiation, *Freshness*
Information disclosure	attackers get access to data in transit or in a data store	Confidentiality, Privacy
Denial of service	attackers interrupt a system’s legitimate operation	Availability
Elevation of privilege	attackers perform actions they are not authorized to perform	Authorization

4.3.3 Threats and vulnerabilities

“Attackers” give rise to “threats” that exploit “vulnerabilities” leading to “risks” to “assets”. However, nearly all of the existing threat analysis and risk assessment methodologies do not consider vulnerabilities explicitly. Furthermore, the existing approaches often apply the terms “vulnerability”, “threat”, and “attack” inconsistently. For example, during the identification of a vulnerability, TVRA suggests as follows: “Possible attack interfaces need to be identified and all possible attacks need to be elaborated [9].” Common Criteria [6] suggests preparing a list of potential vulnerabilities applicable to the TOE in its operational environment, which can be used as an input into penetration testing activities. This activity is to be performed during vulnerability assessment of the TOE. Appendix B of [8] provides informative guidelines about vulnerability analysis and assessment.

We suggest considering threats prior to vulnerabilities during threat analysis and risk assessment (TARA). The reason is as follows. TARA is performed during the concept phase and we may not have detailed information (e.g., implementation) about the TOE. On the other hand, vulnerabilities may be introduced during product development and implementation (i.e., implementation vulnerabilities) and configuration of a system (e.g., configuration vulnerabilities). As a result, it is nearly impossible during the concept phase to argue about vulnerabilities that are related to the specific implementation. However, in the context of the automotive industry, it is worthy of considering available vulnerability data, for example, Mitre's Common Vulnerabilities and Exposures (CVE) and the USA Government's National Vulnerability Database (NVD). Also, numerous vulnerabilities referred to as WIFFs (Weakness, Idiosyncrasies, Faults, and Flaws) are grouped into a number of generic categories and are explained with examples by researchers [57]. In fact, Common Criteria [8] suggests that "The evaluator examines the sources of information publicly available to support the identification of possible potential vulnerabilities in the TOE." Such an analysis of vulnerabilities may provide a better understanding of the consequences of the known vulnerabilities and how those were exploited as well as mitigated. This can potentially lead to more effective prioritization of threats as well as identification of countermeasures.

However, it is almost impossible to establish a direct one-to-one mapping between vulnerabilities and threats because a particular vulnerability may lead to several threats and a particular threat may exploit several vulnerabilities. Generally, vulnerabilities do not directly define threats, and threats do not impose a specific kind of vulnerability being exploited [58]. On the other hand, it is possible to use the WIFFs categories [57] and Common Criteria guidelines presented in Appendix B.2.1 of [8] to establish generic mapping between threats and vulnerabilities. However, in the current version (Version 1.0) of the deliverable, we do not aim at establishing threat-vulnerability mapping.

4.4 Risk assessment

Risk assessment refers to ranking of the threats. After identifying the threat-asset pairs for a particular use case based on STRIDE approach, we proceed with risk assessment to rank the threats, i.e., to derive security level for each threat-asset pair. Security Level (SL) is a measure of the needed strength of security mechanisms for a security relevant asset to meet a certain level of security. The risks are balanced by usage of security levels for a defined environment including threats and attackers. Risk assessment consists of three steps: (a) determination of threat level (TL): this corresponds to the estimation of the "likelihood" component of risk, (b) determination of impact level (IL): this corresponds to the estimation of the "impact" component of risk, and (c) determination of security level (SL): this corresponds to the final risk rating.

4.4.1 Threat Level (TL)

In this sub-section, we first present threat level (TL) parameters that have been presented in state-of-the-art risk assessment methodologies and attempt to justify the selection of the TL parameters in the HEAVENS security model. Next, we elaborate on the selected TL parameters. Finally, we provide guideline about selecting values for each of the TL parameters to estimate threat level.

4.4.1.1 State-of-the-art and HEAVENS

Table 4-3 provides a comparative view of the parameters that are proposed in the literature to estimate threat level/attack potential/likelihood. It is notable that

- Common Criteria [8], TVRA [9], and EVITA [53] use the same set of five parameters to estimate the attack potential. However, CC and TVRA, ETSI apply different scale for setting the values of each parameter.

- OWASP uses eight parameters to estimate likelihood component of the risk: four (skill level, motive, opportunity, size) are grouped as threat agent factors and the remaining four (ease of discovery, ease of exploit, awareness, intrusion detection) as vulnerability factors [33].
- SECTRA model uses four parameters to estimate threat level [64]. SECTRA model first applies three parameters (Expertise, Knowledge of the target, Availability of resources) to estimate the attack potential. Then, it augments the estimated attack potential with "Opportunity" to identify the threat level.

Table 4-3: Comparative view of the parameters to determine threat level.

CC	TVRA, ETSI	OWASP	EVITA	SECTRA	HEAVENS
Elapsed Time	Time		Elapsed Time		
Expertise	Expertise	Skill Level	Expertise	Expertise	Expertise
Knowledge of TOE	Knowledge	Awareness	Knowledge of System	Knowledge of the target	Knowledge about TOE
Window of Opportunity	Opportunity	Opportunity	Window of Opportunity	Opportunity	Window of opportunity
Equipment	Equipment		Equipment	Availability of resources	Equipment
		• Motive • Size • Intrusion Detection • Ease of Discovery • Ease of Exploit			

Based on these models, we choose to consider four parameters to estimate threat level in the HEAVENS security model: (a) Expertise, (b) Knowledge about TOE, (c) Window of opportunity, and (d) Equipment. We do not consider the OWASP model for further analysis because the model is mainly suitable for web applications. Also, the model applies too many parameters which are often overlapping and may lead to inconsistencies. Then, we continue analyzing the parameters involved in the other models to justify the selection of the parameters for the HEAVENS security model. The principles and rationales include:

- In general, the HEAVENS security model attempts to consistently adopt the terms of the Common Criteria wherever applicable. The goal is to stay aligned with the existing and well-known standards as much as possible. However, the HEAVENS model applies different scale to select values for each parameter in contrast to the Common Criteria.
- The HEAVENS model excludes the "Elapsed Time" parameter because this is not a first-order parameter while deriving threat level. The elapsed time required to mount a particular attack on a particular asset is proportional to the values of the other parameters, namely, expertise, equipment, knowledge about TOE and window of opportunity. For example, depending on the

attacker's skill level and the availability of the required equipment to mount the attack, the elapsed time may vary significantly – from less than an hour to several months.

- The HEAVENS model does not consider the "Motivation" of the attackers (i.e., threat agents) as one of the parameters to estimate threat level. This is in line with the rationale presented in clause B.4.1.1 of the Common Criteria Evaluation methodology [8] and Annex B of ETSI TS 102 165-1 V4.2.3 (2011-03) [10]. For example, if a successful attack is highly rewarding (may depend on the value of the asset under consideration or may lead to personal gains, for example, better performance, lower cost, etc.) for an attacker, he/she will most likely spend time and money not only to increase own skill level and knowledge of target but also to gain access to expensive and sophisticated equipment to carry out the attack.
- Unlike the other models, the HEAVENS model applies a linear scale with four values (from zero to three) for each parameter and assigns equal weight to each of the parameters. This facilitates consistent reasoning about the different parameters while deriving the threat level for a particular threat-asset pair.

4.4.1.2 Overview of the threat level parameters

- The parameter "**Expertise**" refers to the level of generic knowledge of the underlying principles, product type or attack methods that are required to carry out an attack on the TOE. The identified levels are as follows:
  - ✓ "Layman" is unknowledgeable compared to experts or proficient persons, with no particular expertise; Examples may include persons who can only follow simple instructions that come with the available tools to mount simple attacks, but not capable of making progresses himself/herself if the instructions or the tools do not work as expected.
  - ✓ "Proficient" persons have general knowledge about the security field and are involved in the business, for example, workshop professionals. Proficient persons know about simple and popular attacks. They are capable of mounting attacks, for example, odometer tuning and installing counterfeit parts, by using available tools and if required, are capable of improvising to achieve the desired results.
  - ✓ "Experts" are familiar with the underlying algorithms, protocols, hardware, structures, security behavior, principles and concepts of security employed, techniques and tools for the definition of new attacks, cryptography, classical attacks for the product type, attack methods, etc. implemented in the product or system type.
  - ✓ The level "Multiple Experts" is introduced to allow for a situation, where different fields of expertise are required at an Expert level for distinct steps of an attack.
- The parameter "**Knowledge about TOE**" refers to the availability of information about the TOE and the community size that possesses knowledge about the TOE from an attacker perspective. This parameter points to the sources from where attackers can gain knowledge about the TOE and indicates how easy or difficult it can be for an attacker to acquire knowledge about the TOE. Identified levels are as follows:
  - ✓ "Public" information concerning the TOE (e.g. as gained from the Internet, bookstore, information shared without non-disclosure agreements);

- ✓ "Restricted" information concerning the TOE (e.g. knowledge that is controlled within the developer organization and shared with other organizations, for example, between suppliers and OEMs, under a non-disclosure agreement). Examples include requirements and design specifications, internal documentation.
- ✓ "Sensitive" information about the TOE (e.g. knowledge that is shared between discrete teams within the developer organization, access to which is constrained only to members of the specified teams). Examples include restricted ECU configuration parameters to enable/disable features in vehicles, vehicle configuration database, software source code.
- ✓ "Critical" information about the TOE (e.g. knowledge that is known by only a few individuals, access to which is very tightly controlled on a strict need to know basis and individual undertaking). Examples include secret root signing key.

Figure 4-3 shows one possible way of interpreting the different levels of the parameter "Knowledge about TOE". The value is always either "Sensitive" or "Critical" if the knowledge is available only within the organization, for example, an OEM. On the other hand, the value is always either "Restricted" or "Public" if the knowledge is available outside the organization, for example, suppliers, customers. The attack potential increases as we move from "Critical" to "Public" because of the increased easiness for an attacker to acquire information about the TOE and the increased size of the community that possesses knowledge about the TOE.

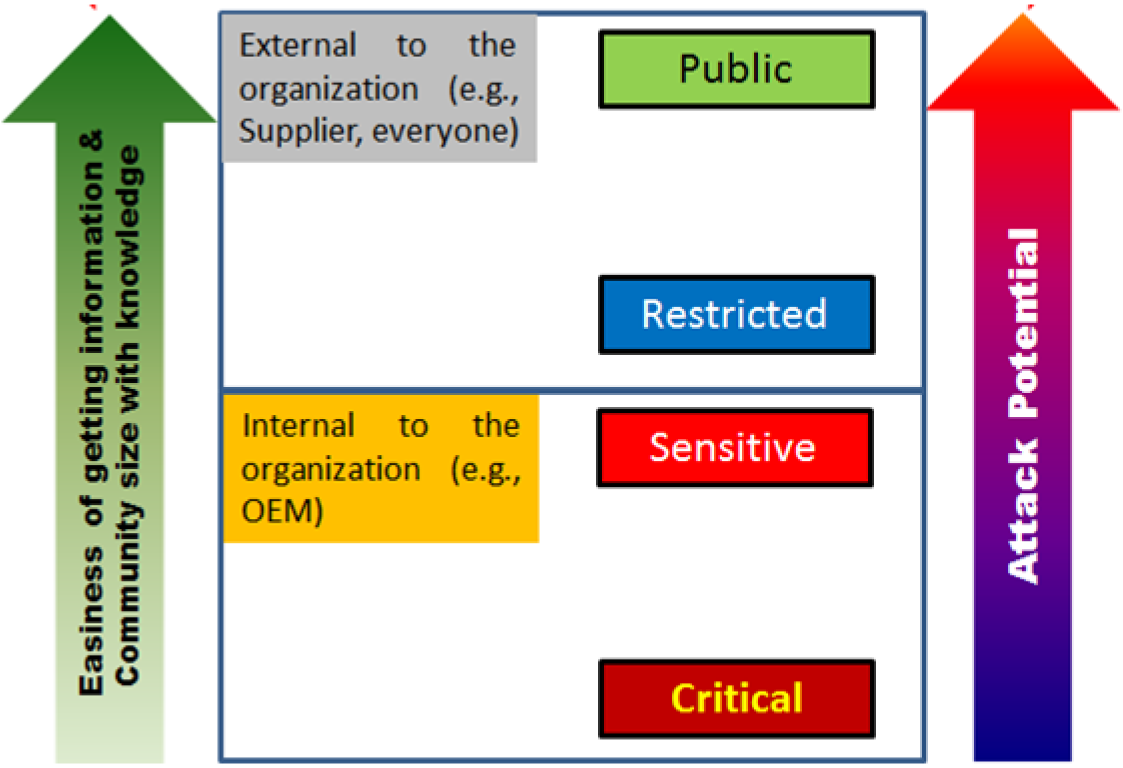

Easiness of getting information & Community size with knowledge
External to the organization (e.g., Supplier, everyone)
Public
Restricted
Internal to the organization (e.g., OEM)
Sensitive
Critical
Attack Potential

Figure 4-3: Knowledge about TOE.

- The parameter "**Equipment**" refers to the equipment required to identify or exploit vulnerability and/or mount an attack.
  - ✓ "Standard" equipment is readily available to the attacker, either for the identification of vulnerability or for an attack. This equipment may be a part of the TOE itself (e.g. a debugger in an operating system), or can be readily obtained (e.g. Internet downloads, protocol analyzer or simple attack scripts). Examples include simple OBD diagnostics devices, common IT device such as notebook.
  - ✓ "Specialized" equipment is not readily available to the attacker, but could be acquired without undue effort. This could include purchase of moderate amounts of equipment (e.g. power analysis tools, use of hundreds of PCs linked across the Internet would fall into this category), or development of more extensive attack scripts or programs. Examples include in-vehicle communication devices (e.g., CAN cards), costly workshop diagnosis devices. If clearly different test benches consisting of specialized equipment are required for distinct steps of an attack this would be rated as bespoke.
  - ✓ "Bespoke" equipment is not readily available to the public as it may need to be specially produced (e.g. very sophisticated software), or because the equipment is so specialized that its distribution is controlled, possibly even restricted. Alternatively, the equipment may be very expensive.
  - ✓ The level "Multiple Bespoke" is introduced to allow for a situation, where different types of bespoke equipment are required for distinct steps of an attack.
- The parameter "**Window of opportunity**" combines access type (e.g., logical, physical) and access duration (e.g., unlimited, limited) that are required to mount an attack on the TOE by an attacker. The different levels include:
  - ✓ "Low": Very low availability of the TOE. Physical access required to perform complex disassembly of vehicle parts to access internals to mount an attack on the TOE.
  - ✓ "Medium": Low availability of the TOE. Limited physical and/or logical access to the TOE. Physical access to vehicle interior or exterior without using any special tool (e.g., opening the hood to access wires).
  - ✓ "High": High availability and limited time. Logical or remote access without physical presence.
  - ✓ "Critical": High availability via public/untrusted network without any time limitation (i.e., TOE/asset is always accessible). Logical or remote access without physical presence and time limitation as well as unlimited physical access to the TOE/asset. Examples include wireless or via Internet (e.g., V2X or cellular interfaces).

4.4.1.3 Applying the threat level parameters

Table 4-4 presents the different parameters and the values to be used for each parameter.

Table 4-4: Applying the TL parameters to estimate threat level.

Parameter	Value	Explanation
Expertise		
Layman	0	Section 4.4.1.2 “Overview of the threat level parameters” of this deliverable.
Proficient	1	
Expert	2	
Multiple experts	3	
Knowledge about TOE		
Public	0	Section 4.4.1.2 “Overview of the threat level parameters” of this deliverable.
Restricted	1	
Sensitive	2	
Critical	3	
Window of opportunity		
Critical	0	Section 4.4.1.2 “Overview of the threat level parameters” of this deliverable.
High	1	
Medium	2	
Low	3	
Equipment		
Standard	0	Section 4.4.1.2 “Overview of the threat level parameters” of this deliverable.
Specialized	1	
Bespoke	2	
Multiple bespokes	3	

Finally, for each threat-asset pair, we sum the values of each of the parameters and define ranges to determine a threat level corresponding to each identified range. We adopt five different threat levels (None, Low, Medium, High, and Critical) as shown in Table 4-5.

Table 4-5: Estimating the Threat Level (TL).

Summation of the values of the TL parameters	Threat Level (TL)	TL Value
> 9	None	0
7 – 9	Low	1
4 – 6	Medium	2
2 – 3	High	3
0 – 1	Critical	4

4.4.2 Impact Level (IL)

In this sub-section, we first present impact level (IL) parameters that have been presented in state-of-the-art risk assessment methodologies and attempt to justify the selection of the IL parameters in the HEAVENS security model. Next, we elaborate on the selected IL parameters. Finally, we provide guideline about selecting values for each of the IL parameters to derive impact level.

4.4.2.1 State-of-the-art and HEAVENS

Common Criteria [6] does not apply risk assessment in line with the traditional approach of estimating "impact" component of risk. It only estimates attack potential which is conceptually similar to threat level estimation in the HEAVENS security model.

OWASP [33] considers eight impact parameters that are grouped into two categories: technical impact factors (loss of integrity, loss of availability, loss of confidentiality, loss of accountability) and business impact factors (financial damage, reputation damage, non-compliance, privacy violation). However, from the HEAVENS perspective, business impact originates from technical impact. Moreover, the technical impact factors are connected with the security attributes whereas the business impact factors are connected with the security objectives.

EVITA proposes five severity classes (0 – 4) where '0' corresponds to no impact and '4' corresponds to the most severe impact for each of the four aspects of the security threats (i.e., security objectives) – safety, privacy, financial and operational [53].

TVRA [10] estimates the impact of a threat with values from 1 to 3. It is the impact on the system from a successful attack on a specific asset that is particularly important when analyzing a TOE. Table 4-6 identifies the three levels of impact used to evaluate assets in the TVRA process [9].

Table 4-6: Asset impact [9].

Impact	Explanation	Value
Low	The concerned party is not harmed very strongly; the possible damage is low	1
Medium	The threat addresses the interests of providers/subscribers and cannot be neglected	2
High	A basis of business is threatened and severe damage might occur in this context	3

SECTRA security model [64] defines four impact levels (low, medium, high, high+) to estimate harm that a compromised asset can cause to the defined stakeholders.

In the HEAVENS security model, four parameters are used to estimate the impact level: (a) Safety, (b) Financial, (c) Operational, and (d) Privacy and legislation. The parameters consider the security objectives as defined in Section 5.1 of the HEAVENS Deliverable D1.1 Needs and requirements [16]. The parameters are used to estimate the possible consequences of successful attacks on the TOE from the viewpoint of security objectives as defined by the stakeholders. In the HEAVENS model, the impacts are estimated in relation to the stakeholders (e.g., OEM, Fleet owner, Vehicle owner, Road users, Service Providers, Driver, Workshop personnel, Dealers, Insurance company, Authority) as defined in Section 5.4 of the HEAVENS Deliverable D1.1 Needs and requirements [16]. We do not consider the potential benefits of an attacker while estimating the impact level. The attacker's perspective is included in the derivation of the threat level.

4.4.2.2 Overview of the impact level parameters

It is a first-order requirement to ensure safety of the vehicle occupants, road users and infrastructures. The "**Safety**" parameter to estimate the safety impact is adopted from the ISO 26262 [20]:

- No injury
- Light and moderate injuries
- Severe and life-threatening injuries (survival probable)
- Life-threatening injuries (survival uncertain), fatal injuries

The "**Financial**" category considers all financial losses or damages that can be either direct or indirect. Direct financial damages may include product liability issues (e.g., penalties, recalls), legislation issues (e.g., penalties due to nonconformance), product features (e.g., loss in business due to illicit activation of sellable features). On the other hand, indirect financial damages may include damage to OEM reputation, loss of market share, IP infringement, etc. Direct financial losses are relatively easier to calculate and categorize whereas it's harder to estimate numerical values corresponding to indirect financial damages. Also, safety issues may contribute to financial damages. For example, recent recalls of certain models of cars by several OEMs due to various safety issues have financial impact on each of the OEMs. To summarize, the financial damage is the sum of direct and indirect costs for the OEM and the root cause may originate from any of the stakeholders.

The "**Operational**" category includes operational damages caused by unwanted and unexpected incidents. Examples of such operational damages include loss of secondary (e.g., cruise control) and comfort/entertainment (e.g., cd-player, air-conditioning) functionalities of the vehicle. However, in certain situations, operational damages may cause safety and financial damages. For example, operational damages in the form of loss of primary and safety-related vehicle functionalities may affect safety of passengers and road users. Consequently, the impact of the operational category on the overall impact is relatively lower with respect to the safety and financial categories.

The “**Privacy and legislation**” category includes damages caused by privacy violation of stakeholders (e.g., fleet owner, vehicle owner, driver) and/or violation of legislations/regulations (e.g., environmental, driving). We merge privacy and legislation into one parameter because privacy may be enforced through legislation and there exist legislations that are not related to privacy. Usually, such damages do not have direct injury, financial and operational dimensions. However, in certain situations, privacy and legislation violations may cause financial (e.g., fine, loss of access to certain market) and operational damages to the stakeholders. Consequently, the impact of the privacy and legislation category is relatively lower with respect to the safety and financial categories.

4.4.2.3 Applying the impact level parameters

In the HEAVENS model, we assign different weights to the different impact parameters. The “Safety” and “Financial” parameters have equal weights while estimating the overall impact level. The impact of safety and financial parameters can lead to the most severe consequences for stakeholders, for example, vehicle occupants may not survive, organizations may bankrupt. On the other hand, the impact of “Operational” as well as “Privacy and legislation” parameters on the overall impact is relatively lower with respect to the safety and financial damages. To reflect this fact during impact level estimation, we reduce the corresponding factors by a magnitude of one in case of operational as well privacy and legislation with respect to the safety and financial parameters. The different safety levels and the corresponding values to estimate the impact of safety is shown in Table 4-7.

Table 4-7: Impact level parameter – safety.

Safety	Value	Explanation
No injury	0	ISO 26262-3
Light and moderate injuries	10	
Severe and life-threatening injuries (survival probable)	100	
Life-threatening injuries (survival uncertain), fatal injuries	1000	

The categorization of financial damages depends on the financial strength of an individual stakeholder. For example, a loss of €100,000 could be relatively trivial compared to the sales volume and budget of a large enterprise whereas even a loss of €10,000 could threaten the existence of a small enterprise. It may therefore be appropriate to express the limits as percentages of total sales, total profit, or on a similar base value as well as to classify the damages qualitatively into damage categories instead of calculating the damages quantitatively [59]. Table 8 shows a comparison of the damage categories and the protection requirement categories [59]. Based on this, Table 9 suggests one possible categorization of financial damages.

Table 4-8: Damage categories and protection requirements categories [59].

Damage categories		Protection requirements categories	
Category	**Explanation**	**Category**	**Explanation**
“low”	Failure has a minor, barely noticeable effect.		
“normal”	Failure has noticeable effects.	“normal”	The effects of the damage are limited and manageable.
“high”	Failure has serious effects.	“high”	The effects of damage can be considerable.
“very high”	Failure or impairment leads to effects that threaten the existence of the organization.	“very high”	The effects of the damage can reach a catastrophic level that threatens the existence of the organization

Table 4-9: Impact level parameter – financial.

BSI-Standard [59]	HEAVENS		Explanation based on BSI-Standard 100-4 [59]
Damage category	**Financial**	**Value**	
Low	No impact	0	• No discernible effect. No appreciable consequences
Normal	Low	10	• The financial damage remains tolerable to the organization
High	Medium	100	• The resulting damage leads to substantial financial losses, but does not threaten the existence of the organization
Very High	High	1000	• The financial damage threatens the existence of the organization

We adapt the vehicular defect severity categorization such as FMEA (Failure Mode and Effects Analysis) [60] to classify the operational damages. This is shown in Table 4-10.

Table 4-10: Impact level parameter – operational.

Severity of Effect on Product (Effect on Customer) [60]	Effect [60]	Severity Rank [60]	HEAVENS Value
No discernible effect	No effect	1	No Impact (0)
Appearance item or audible noise (vehicle still operates, but does not conform, annoys more than 25% of customers)	Minor disruption	2	Low (1)
Appearance item or audible noise (vehicle still operates, but does not conform, annoys more than 50% of customers)		3	
Appearance item or audible noise (vehicle still operates, but does not conform, annoys more than 75% of customers)	Moderate disruption	4	
Degradation of secondary function (vehicle still operable, but comfort or convenience functions work at a reduced level of performance)	Moderate disruption	5	Medium (10)
Loss of secondary function (vehicle still operable, but comfort or convenience functions do not work)		6	
Degradation of primary function (vehicle still operates, but at a reduced level of performance)	Significant disruption	7	
Loss of primary function (vehicle inoperable, but does not affect safe vehicle operation)	Major disruption	8	High (100)
Potential failure mode affects safe vehicle operation with some warning or noncompliance with government regulations		9	
Potential failure mode affects safe vehicle operation without warning or involves noncompliance with government regulations	Fails to meet safety or regulatory requirements	10	

It has already been mentioned that privacy and legislation category includes the damages caused by privacy violation of stakeholders (e.g., fleet owner, vehicle owner, driver) and/or violation of legislations/regulations (e.g., environmental, driving). Table 4-11 shows one possible way of assigning different values to this parameter. There is a possibility to align the privacy aspect with the “Privacy Impact Assessment Guideline” provided by BSI, Germany [67].

Table 4-11: Impact level parameter – privacy and legislation.

Privacy & legislation	Value	Explanation
No impact	0	• No discernible effects in relation to violations of privacy and legislation
Low	1	• Privacy violations of a particular stakeholder (e.g., vehicle owner, driver) which may not lead to abuses (e.g., impersonation of a victim to perform actions with stolen identities) • Violation of legislations without appreciable consequences for business operations and finance (e.g., warning without any significant financial penalty, limited media coverage) for any stakeholder (e.g., OEM, fleet owner, driver)
Medium	10	• Privacy violations of a particular stakeholder (e.g., vehicle owner, driver) leading to abuses (e.g., impersonation of a victim to perform actions with stolen identities) and media coverage • Violation of legislations with potential of consequences for business operations and finance (e.g., financial penalties, loss of market share, media coverage)
High	100	• Privacy violation of multiple stakeholders (e.g., fleet owners, multiple vehicle owners and multiple drivers) leading to abuses (e.g., impersonation of a victim to perform actions with stolen identities). Such a level of privacy violation may lead to extensive media coverage as well as severe consequences in terms of loss of market share, business operations, trust, reputation and finance for OEMs and fleet owners • Violation of legislations (e.g., environmental, driver) causing significant consequences for business operations and finance (e.g., huge financial penalties, loss of market share) as well as extensive media coverage

Finally, we sum the values of all the impact parameters to estimate the impact level (see Table 4-12).

Table 4-12: Estimating Impact Level (IL).

Summation of the values of the impact parameters	Impact Level (IL)	IL Value
0	No Impact	0
1 – 19	Low	1
20 – 99	Medium	2
100 – 999	High	3
>= 1000	Critical	4

We summarize the values associated with each of the impact parameters and the estimated impact level in Figure 4-4.

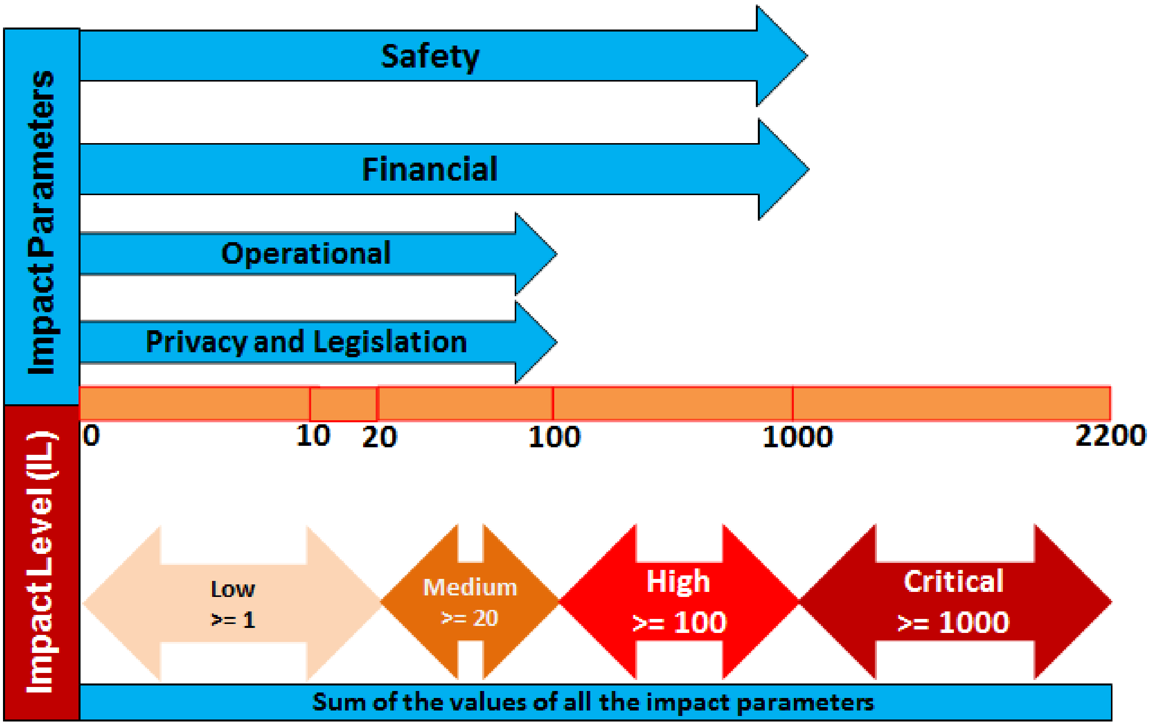


Figure 4-4: Impact parameters and impact level in the HEAVENS security model.

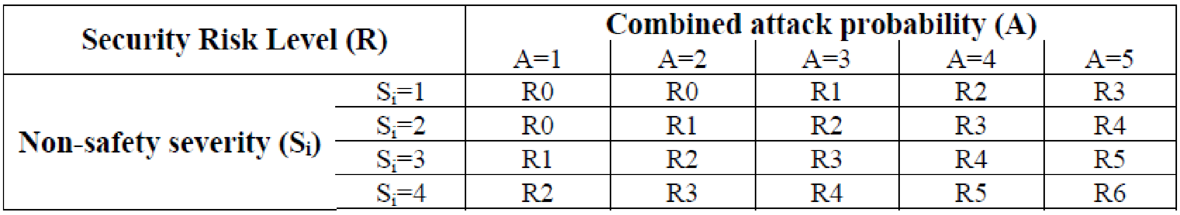
4.4.3 Security Level (SL)

In this sub-section, we first briefly present state-of-the-art methodologies. Next, we present how to estimate security level in relation to the HEAVENS security model.

The OWASP risk rating methodology [33] suggests summing up the values of the eight likelihood parameters and calculating arithmetic mean to define likelihood level. A similar approach is followed to define impact level [33] by using the values of the four technical impact factors and the four business impact factors. Finally, likelihood and impact levels are combined to estimate overall risk severity (none, low, medium, high, critical). OWASP [33] methodology attempts to estimate whether the likelihood is low, medium, or high and then do the same for impact. The 0 to 9 scale is split into three parts as shown below:

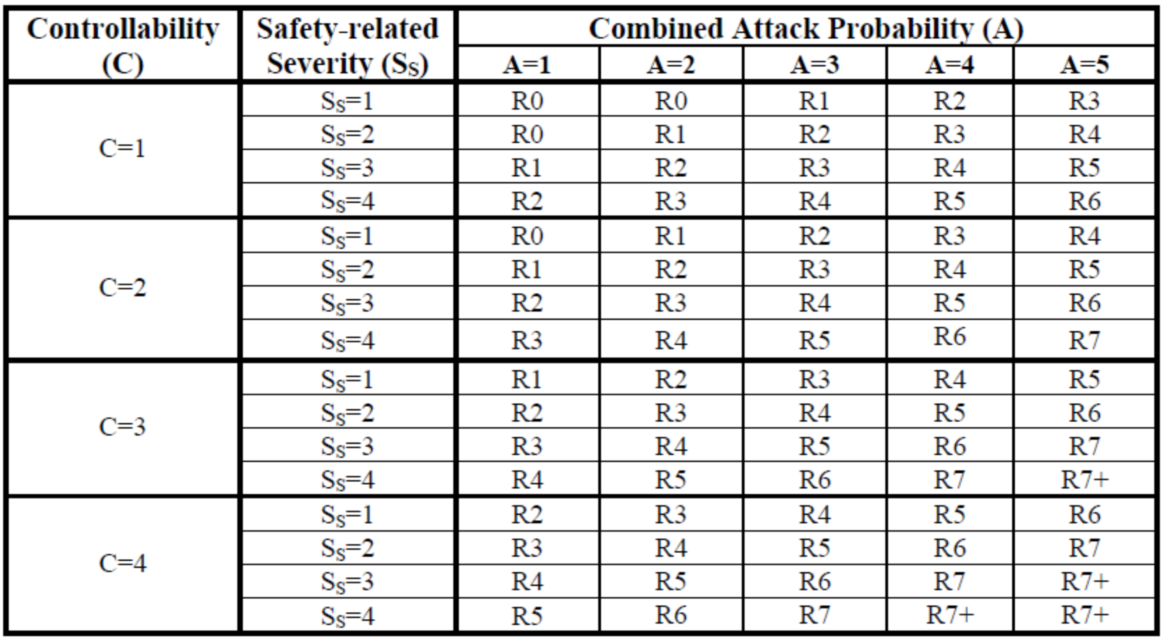
Likelihood and Impact Levels	
0 to <3	LOW
3 to <6	MEDIUM
6 to 9	HIGH

EVITA [53] suggests two different approaches to derive risk level. The risk level (**R**, a vector) is determined from the severity (**S**) associated with the attack objective and the combined attack probability (*A*) associated with a particular attack method. This is achieved by mapping the severity and attack probability to the risk using a "risk graph" approach [53]. For severity aspects that are not safety related (privacy, financial and operational) the risk graph maps two parameters (attack probability and severity) to a qualitative risk level. Combinations of severity and combined attack probability are mapped to a range of "security risk levels" (denoted *Ri*, where "*i*" is an integer). This is shown in Figure 4-5. In order to include the additional parameter (controllability) in the assessment of safety related security risks it is necessary to use of a different risk graph as proposed in Figure 4-6, which maps three parameters (severity, attack probability, controllability) to qualitative risk levels [53].



Security Risk Level (R)		Combined attack probability (A)				
		A=1	A=2	A=3	A=4	A=5
Non-safety severity (S_i)	S_i=1	R0	R0	R1	R2	R3
	S_i=2	R0	R1	R2	R3	R4
	S_i=3	R1	R2	R3	R4	R5
	S_i=4	R2	R3	R4	R5	R6

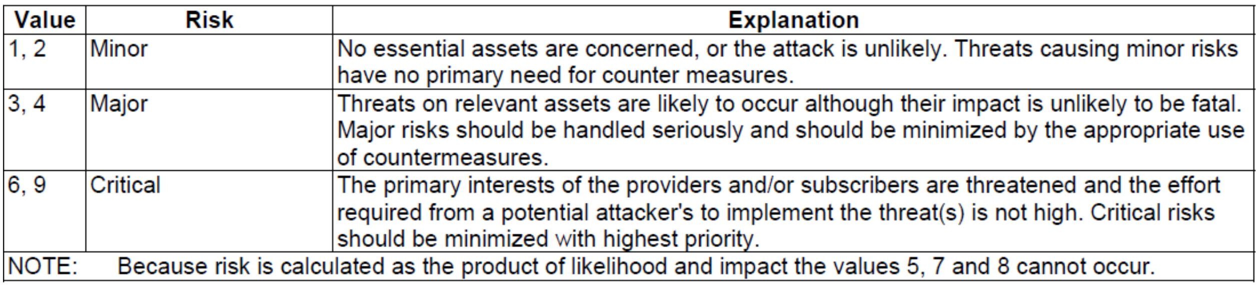
Figure 4-5: Security risk graph for non-safety related security threats [53].



Controllability (C)	Safety-related Severity (S_S)	Combined Attack Probability (A)				
		A=1	A=2	A=3	A=4	A=5
C=1	S_S=1	R0	R0	R1	R2	R3
	S_S=2	R0	R1	R2	R3	R4
	S_S=3	R1	R2	R3	R4	R5
	S_S=4	R2	R3	R4	R5	R6
C=2	S_S=1	R0	R1	R2	R3	R4
	S_S=2	R1	R2	R3	R4	R5
	S_S=3	R2	R3	R4	R5	R6
	S_S=4	R3	R4	R5	R6	R7
C=3	S_S=1	R1	R2	R3	R4	R5
	S_S=2	R2	R3	R4	R5	R6
	S_S=3	R3	R4	R5	R6	R7
	S_S=4	R4	R5	R6	R7	R7+
C=4	S_S=1	R2	R3	R4	R5	R6
	S_S=2	R3	R4	R5	R6	R7
	S_S=3	R4	R5	R6	R7	R7+
	S_S=4	R5	R6	R7	R7+	R7+

Figure 4-6: Security risk graph for safety related threats [53].

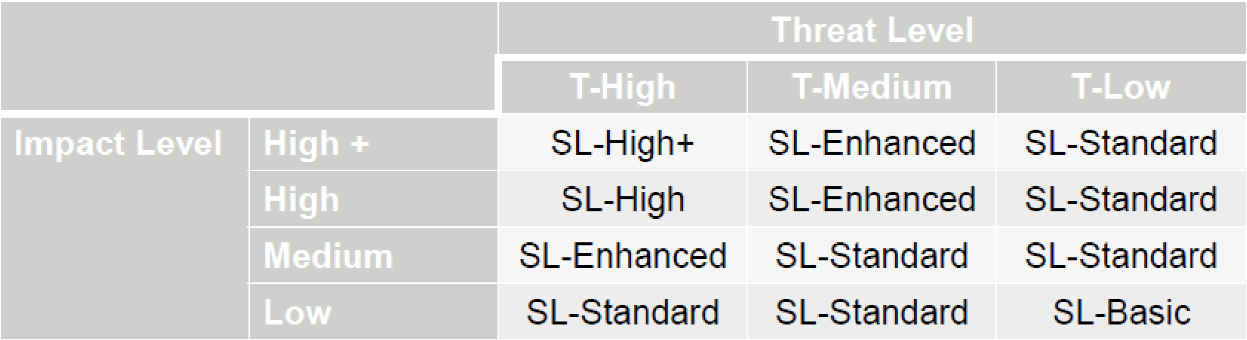
According to the TVRA methodology [9], the likelihood of a threat occurring may be estimated with values from 1 to 3 and the impact of a threat is also estimated with values from 1 to 3. The product of occurrence likelihood and impact value provides the risk which serves as a measurement for the risk that the concerned asset is compromised. The result is classified into three categories as shown in Figure 4-7.



Value	Risk	Explanation
1, 2	Minor	No essential assets are concerned, or the attack is unlikely. Threats causing minor risks have no primary need for counter measures.
3, 4	Major	Threats on relevant assets are likely to occur although their impact is unlikely to be fatal. Major risks should be handled seriously and should be minimized by the appropriate use of countermeasures.
6, 9	Critical	The primary interests of the providers and/or subscribers are threatened and the effort required from a potential attacker's to implement the threat(s) is not high. Critical risks should be minimized with highest priority.
NOTE: Because risk is calculated as the product of likelihood and impact the values 5, 7 and 8 cannot occur.		

Figure 4-7: Risk rating according to the TVRA [9].

SECTRA methodology [64] suggests combining impact level and threat level as shown in Figure 4-8 to determine security level.



		Threat Level		
		T-High	T-Medium	T-Low
Impact Level	High +	SL-High+	SL-Enhanced	SL-Standard
	High	SL-High	SL-Enhanced	SL-Standard
	Medium	SL-Enhanced	SL-Standard	SL-Standard
	Low	SL-Standard	SL-Standard	SL-Basic

Figure 4-8: Determining security level using SECTRA security model.

In the HEAVENS security model, we combine Threat Level (TL) and Impact Level (IL) to derive Security Level as shown in Table 4-13. The security level is set to “Critical” if and only if both TL and IL have the value 4 (“Critical”).

Table 4-13: Security level based on threat level and impact level.

Security Level (SL)		Impact Level (IL)				
		0	1	2	3	4
Threat Level (TL)	0	QM	QM	QM	QM	Low
	1	QM	Low	Low	Low	Medium
	2	QM	Low	Medium	Medium	High
	3	QM	Low	Medium	High	High
	4	Low	Medium	High	High	Critical

4.5 Security requirements

After performing threat analysis and risk assessment, the final part of the HEAVENS security model deals with deriving security requirements based on asset, threat, security attribute and security level. Consider the example shown in Table 4-14. As shown in the third row, the asset “CAN Signal X on Bus A” has a security level “QM”. Hence, we don’t need to formulate any additional security requirement for this asset to deal with spoofing threat and to enforce authenticity. On the other hand, we need to formulate security requirements for the other two cases.

Note that there might be several threats for one asset and as a result, we may have multiple security levels based on multiple threat levels for all the threats related to an asset. One approach of determining a security level for the asset as a whole is to consider the highest security level out of all the security levels for all the threats associated with the asset. An alternative is to consider the highest threat level together with the impact level to define the security level for the asset.

Table 4-14: Examples of deriving security requirements.

No.	Asset	Threat	Security Attribute	Security Level
1	Cryptographic key	Elevation of privilege	Authorization	Critical
2	ECU Software	Tampering	Integrity	Medium
3	CAN Signal X on Bus A	Spoofing	Authenticity	QM

4.6 Tool support

We have used the Microsoft SDL Threat Modeling tool [38] to carry out threat analysis. The tool is used to create a data flow diagram (DFD) for each of the selected use cases based on the available description. The use cases and description are presented in HEAVENS Deliverable D1.1 Needs and requirements [16]. Based on the DFD, the tool then automatically identifies the assets and generates the threats for each of the assets. The results are then used as input for the risk assessment tool. Figure 4-9 shows the HEAVENS tool chain that is used to apply the HEAVENS security model to facilitate deriving security requirements.

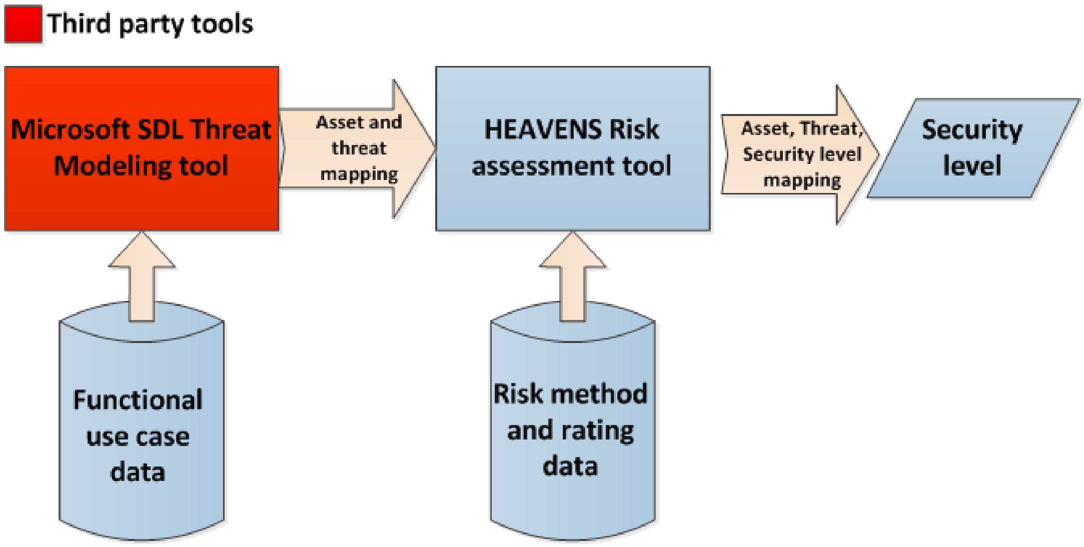

Third party tools
Microsoft SDL Threat Modeling tool
Asset and threat mapping
HEAVENS Risk assessment tool
Asset, Threat, Security level mapping
Security level
Functional use case data
Risk method and rating data

Figure 4-9: Tool support for the HEAVENS security model.

We have developed a risk assessment tool to rate the identified threats. The tool consists of two parts: the first part is a parser and the second part is used to perform risk rating as well as compare multiple risk assessment methodologies (e.g., EVITA, SECTRA, HEAVENS).

- The Microsoft SDL tool generates its output in XML format. The parser parses the assets and the threats generated by the SDL tool. Thus the parser passes a refined version of assets and threats to the next stage.
- The risk assessment tool implements several risk assessment methodologies. The list of the assets and the threats that are obtained from the parser is used in conjugation with a configuration file. The configuration file defines the parameters and the user needs to define the values for each of the parameters. The risk assessment tool then uses the user-defined values to implement the selected risk assessment methodology.

4.7 HEAVENS security model and functional safety

In this section, we present the HEAVENS security model in the context of the functional safety standard ISO 26262. The goal is to understand the cyber-security aspects with respect to the functional safety which is relatively well-known in the automotive industry.

4.7.1 Security Level and Automotive Safety Integrity Level (ASIL)

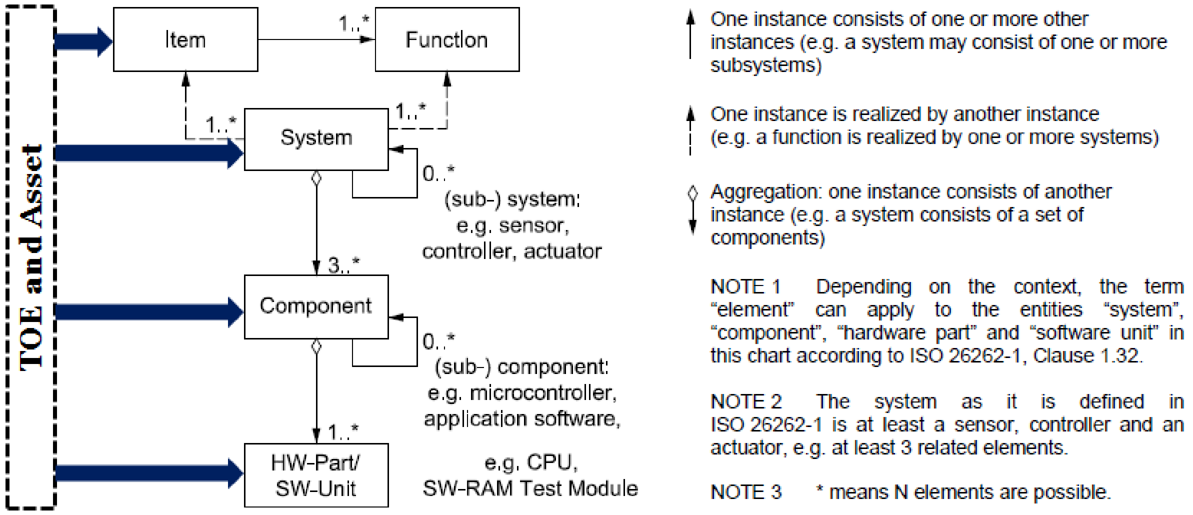
To visualize the HEAVENS security levels in relation to the existing functional safety standards, e.g., ISO 26262, we have mapped the HEAVENS Security Level to the ASIL as suggested in the ISO 26262. Note that functional safety and cyber-security of the automotive E/E systems are different fields of studies although safety and security can potentially be intertwined. As a result, the proposed mapping intends not to enforce any one-to-one correspondence between the HEAVENS Security Level and the ASIL from system/hardware/software development method, tool and process viewpoints. As functional safety is relatively established and well-known within the automotive industry, the mapping aims at establishing a reference point so that the implications of having different security levels on security requirements and security mechanisms can be better understood across functional safety and cyber-security domains. Table 4-15 shows the mapping.

Table 4-15: HEAVENS Security Level and ASIL of ISO 26262.

HEAVENS Security Level	ASIL, ISO 26262
QM	QM
Low	ASIL A
Medium	ASIL B
High	ASIL C
Critical	ASIL D

4.7.2 Target of Evaluation (TOE), Asset and Item

In the IT security domain, TOE is defined by the Common Criteria [6]. A TOE is a set of software, firmware and/or hardware possibly accompanied by guidance. The TOE may be an IT product, a part of an IT product, a set of IT products, a unique technology that may never be made into a product, or a combination of these. In the HEAVENS project, a TOE may consist of one or more assets. On the other hand, ISO 26262-1 defines item as system or array of systems to implement a function at the vehicle level, to which ISO 26262 is applied [19]. The terms item, system, component, hardware part and software unit are defined in ISO 26262-1 and the relationships among those terms are explained in ISO 26262-10. In the context of the HEAVENS security model, a TOE can be at any level (e.g., item, function, system, component, and HW-Part or SW-Unit) with respect to the ISO 26262. Figure 4-10 shows the possible relationship. The figure is adapted from the Part 10 of the ISO 26262.



One instance consists of one or more other instances (e.g. a system may consist of one or more subsystems)

One instance is realized by another instance (e.g. a function is realized by one or more systems)

Aggregation: one instance consists of another instance (e.g. a system consists of a set of components)

NOTE 1 Depending on the context, the term "element" can apply to the entities "system", "component", "hardware part" and "software unit" in this chart according to ISO 26262-1, Clause 1.32.

NOTE 2 The system as it is defined in ISO 26262-1 is at least a sensor, controller and an actuator, e.g. at least 3 related elements.

NOTE 3 * means N elements are possible.

Figure 4-10: Relationship of TOE and item, system, component, HW part and SW unit.

ISO 26262 requires that a safety goal shall be determined for each hazardous event with an ASIL evaluated in the hazard analysis. Also, the ASIL determined for the hazardous event shall be assigned to the corresponding safety goal and if similar safety goals are combined into a single one, the highest ASIL shall be assigned to the combined safety goal [20]. Similarly, if combined safety goals refer to the same hazard in different situations, then the resulting ASIL of the safety goal is the highest one of the considered safety goals of every situation [20]. According to the requirements of the ISO 26262, at least one functional safety requirement shall be specified for each safety goal.

In the automotive cyber-security domain, a TOE can consist of one or more assets and each asset may be related with one or more threats. As a result, we may have several security levels based on combination of different threat levels for different threats and impact level for each asset. However, we may need to define a security level for an asset as a whole. In this respect, similar to the ISO 26262 requirements, we can opt for the highest security level for an asset if multiple security levels are associated with the asset. The same principle can be applied to the TOE.

4.7.3 TARA and HARA

In the HEAVENS project, we perform threat analysis and risk assessment (TARA) to derive security level and security requirements for a "TOE" which may consist of one or more assets. The TOE may refer to an End-to-End (E2E) function, a vehicle function or feature, a sub-system of the E/E system, a software or hardware component and so on. On the other hand, in the functional safety standard ISO 26262, hazard analysis and risk assessment (HARA) are used to determine ASIL and the safety goals for an "item". Safety goals and their assigned ASIL are determined by a systematic evaluation of hazardous events [20]. The ASIL is determined by considering the estimate of the impact factors, i.e. severity, probability of exposure and controllability [20].

HARA is performed during the concept phase of develop functional safety concept that contains safety measures, including the safety mechanisms, to be implemented in the item's architectural elements and specified in the functional safety requirements [20]. Similar to concept of HARA as proposed in the ISO 26262, the HEAVENS security model suggest performing TARA during the concept phase to derive security level and corresponding security requirements. HARA is based on the item's functional behavior; therefore, the detailed design of the item does not necessarily need to be known [20]. Similarly, TARA is based on high-level functional description and behavior of the TOE. The detailed design of the TOE does not necessarily need to be available. Also, similar to the functional safety concept, the HEAVENS security model can suggest security mechanisms to fulfill the security requirements. However, in this version (Version 2.0, Release 2) of the deliverable, we exclude the security mechanisms or countermeasures from the HEAVENS security model. We aim at incorporating security mechanisms in the next revision of the HEAVENS Security Model.

Figure 4-11 shows the relationship between the TARA and the HARA in the context of the ISO 26262. The figure is adapted from the Part 3 of the ISO 26262. It is notable that the applicability of the HARA and the TARA during the concept phase opens up the opportunity of exploiting the existing functional safety processes in the context of the automotive cyber-security.

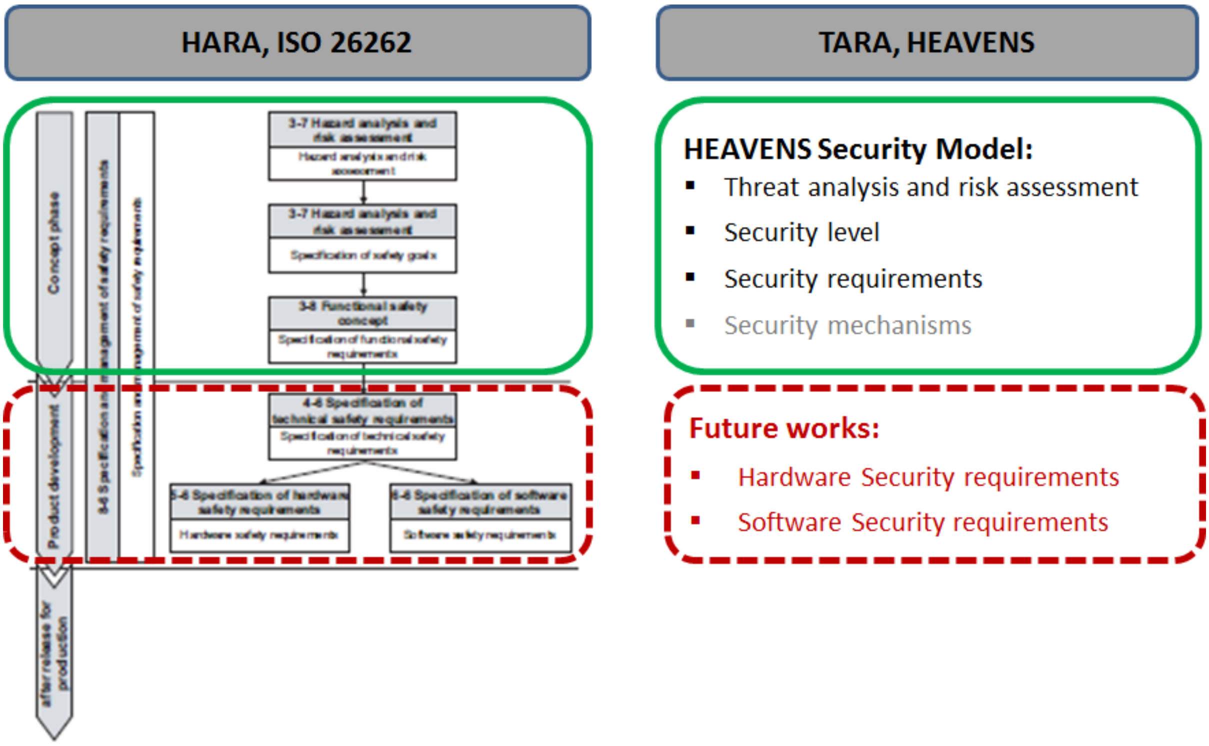

HARA, ISO 26262
Concept phase
Product development
after release for production
8-6 Specification and management of safety requirements
Specification and management of safety requirements
3-7 Hazard analysis and risk assessment
Hazard analysis and risk assessment
3-7 Hazard analysis and risk assessment
Specification of safety goals
3-8 Functional safety concept
Specification of functional safety requirements
4-6 Specification of technical safety requirements
Specification of technical safety requirements
5-6 Specification of hardware safety requirements
Hardware safety requirements
6-6 Specification of software safety requirements
Software safety requirements
TARA, HEAVENS
HEAVENS Security Model:
Threat analysis and risk assessment
Security level
Security requirements
Security mechanisms
Future works:
Hardware Security requirements
Software Security requirements

Figure 4-11: Relationship between HARA and TARA.

4.8 HEAVENS security model and Common Criteria (CC)

HEAVENS security model does not take assurance in effectiveness of implemented security functions into account. HEAVENS security model defines relative strength requirements for security mechanisms. This is two different aspects that should go hand in hand with each other. HEAVENS provides a balanced security level for a TOE and sort of assumes high confidence in the correctness of implemented security functions. The most natural is to align Common Criteria evaluation assurance levels (EAL) with HEAVENS security levels. A higher security level motivates a higher assurance level. However, HEAVENS apply required security level per asset, while Common Criteria applies the same evaluation assurance level for all security functions (SFRs) within the TOE. A TOE in Common Criteria is typically a system or a subsystem. And this does not fit directly into HEAVENS security model. A differentiated method like the safety standard ISO 26262 is needed to make Common Criteria more feasible to work in parallel with HEAVENS security model. In addition, is not motivated to apply one common EAL that correspond that would need to correspond with the highest security level as this would add very high costs and time for evaluation.

One possible solution can be that security mechanisms needed to solve the required security level according to HEAVENS security model is defined to fulfill security functions requirements according to Common Criteria. Evaluation assurance level is applied according to a fixed mapping between security levels and EALs. If a security function protects more than one asset, the strength of the security function corresponds to the required security level that is also matched with the EAL for the security function (SFR). The relationship between applied EALs and security levels for assets has to be investigated further. One other alternative is to define each use case as a TOE. The highest security level of the assets within the TOE defines the EAL for the use case. One complication is that different use cases can share assets. This is however taken care of as the security level for an asset that is shared between multiple use cases is set to the highest one. The advantage of this alternative compared to the first alternative depends on how the use case is realized. A use case that can be seen as a separate entity with well-defined interfaces and functionality, it can probably be defined as a TOE. But if the use case is an integrated function this advantage may not be true. It is likely that some security relevant functions are shared between several different use cases. With this in mind, it might be a better solution to apply EAL on assets instead of use cases. Because, how should a security function that is shared between different TOEs be evaluated without significant overhead?

Common Criteria define how various kinds of security functions shall be implemented in order to meet the security objectives as stated in a Protection Profile or a Security Target. Security requirements are getting more and more stringent with increasing EAL. New requirements are added with increasing EAL. The security requirements in Common Criteria serve as the basis for the definition of security mechanisms that aim to fulfill those requirements. Common Criteria does in other words not support the developer with any kind of security mechanisms like key management schemes or algorithm to fulfill a certain EAL.

In order for a security relevant system to meet a certain level of security we both need assurance in that security measures are effectively implemented and that they have a strength that meets the requirements with respect to actual impact and threat level. The assurance of effectiveness is what Common Criteria can add. Common Criteria defines a well-established process within the security industry that provides quality assurance for IT security solutions. The possibility of having Protection Profiles that is generic for a certain product type enable a more competitive market where customers can compare security solutions from different vendors. We have now two motivations for Common Criteria - market competition and assurance of effectiveness regarding implemented security functions. In order to enable market competition, Protection Profiles should be developed for common use cases. To make use of Protect Profiles for use cases, we also need the security and safety modeling of the use case. For effective specification, development and verification & validation of a use case we need to harmonize assurance requirements between the safety and security domains.

And safety processes is relatively established within the automotive industry, so it is natural to map Common Criteria to ISO 26262 instead of vice versa.

We have performed a mapping among the HEAVENS Security Level (SL), Evaluation Assurance Level (EAL) as proposed by Common Criteria [6], and Trust Assurance Level (TAL) as proposed by the C2C-CC [4] to align the HEAVENS security model with the existing standards (e.g., common criteria) and frameworks (e.g., C2C-CC). Table 4-16 shows one possible mapping. Note that the mapping presented in the table does not aim at establishing a one-to-one correspondence across security level (SL), evaluation assurance level (EAL) and trust assurance level (TAL).

Table 4-16: Mapping of the HEAVENS Security Level with EAL and TAL.

HEAVENS Security Level (SL)	Common Criteria: Minimum Evaluation Assurance Level (EAL)	C2C-CC: Trust Assurance Level (TAL)	Prevents CC Attacker Class
QM	0	0	None
Low	EAL 2	1 and 2	Basic, Enhanced-Basic
Medium	EAL 3	3	Moderate
High	EAL 4	4	High
Critical	EAL 4+	5	Beyond High

Car 2 Car Communication Consortium (C2C-CC) [4] defines Trust Assurance Level (TAL) to facilitate identifying the required security mechanisms to meet the requirements of a particular TAL. C2C-CC also proposes a mapping of TAL with the Evaluation Assurance Level (EAL) defined by Common Criteria. This is shown in Table 4-17.

Table 4-17: Trust Assurance Level (TAL) suggested by C2C-CC and Common Criteria (CC).

C2C-CC TAL	Prevents CC Attacker Class	HW/SW security	CC EAL
0	None	None	0
1	Basic	Software security (e.g., local cryptography)	1
2	Enhanced Basic	Enhanced software security (e.g., distributed and/or integrated multi-ECU software security [protocols])	2
3	Moderate	Simple hardware security (e.g., SHE)	3
4	High	Strong hardware security (e.g., SHE or EVITA with simple tamper-protection)	4
5	Beyond High	Maximum hardware security (e.g., strong smartcard-like tamper-protection)	4+

4.9 Summary and remarks

This chapter (Chapter 4) introduces the HEAVENS security model that includes method, process and tool support to perform threat analysis and risk assessment (TARA). However, stakeholders need to be cautious and well-judged about several factors while carrying out TARA. These are as follows:

- The HEAVENS security model provides guidelines about how to select values for threat level and impact level parameters. However, estimating values of the parameters is inherently subjective. As a result, a stakeholder (i.e., OEM as owner) needs to establish a context with respect to own organization, business and E/E systems while applying the guidelines. For example, categorization (No impact, low, medium, high) of financial damages to estimate impact level can largely be OEM specific. However, the most important aspect is to ensure consistency and apply the same principles for each parameter across the TOEs while determining security level.
- Impact level (estimated using impact level parameters: safety, financial, operational, privacy and legislation) component of the security level are primarily “stakeholder” (e.g., OEM) oriented and relatively “static” in nature – may not change significantly over time. Conversely, threat level parameters (Expertise, Knowledge about TOE, Window of opportunity, Equipment) are primarily “attack/attacker” oriented and relatively “dynamic” in nature – may change drastically over time. For example, availability of cheaper and better tools to mount attacks, sudden disclosure and media coverage of potential vulnerabilities of a system, etc. This should be kept in mind and thought of diligently while estimating threat level parameters during TARA.
- It is important to note that TL parameters are largely dependent on the technology that is used to implement TOE. For example, the usage of Ethernet as communication technology instead of traditional CAN may potentially increase threat level. On the other hand, IL parameters are largely technology independent. As a result, at the concept stage, one may think of only estimating the impact level parameters to perform risk assessment as the details of the design and technological solutions may not be known and/or available. However, at a later stage, when to-be-implemented technological solutions are known, a more rigorous TARA should be performed to look into TL parameters to estimate threat level along with IL parameters.
- Since TARA requires establishing a context in relation to a specific stakeholder, the required tool support to implement TARA methodology always need to be configured accordingly, for example, to choose a value for each impact level parameter for a threat-asset pair. Consequently, tool support to perform TARA to derive security level can at best be semi-automatic and may need customization to meet the requirements of a particular stakeholder.
- Cautions shall be exercised prior to adopting processes from functional safety standards to perform TARA. Although we see similarities between cyber-security and functional safety, the relationship is certainly not at least as straightforward as of replacing “hazard” with “threat” at any given phase of the product development lifecycle. In general, deriving “security level” is more multi-dimensional with respect to deriving its counterpart “ASIL” in functional safety. Also, the parameters (severity, exposure, controllability) used to determine ASIL are much more static in nature over time contrary to the threat level parameters used to determine security level.

5. Evaluation results

This section serves as proof-of-concept implementation of the HEAVENS security model that is presented in Chapter 4 (HEAVENS security model) of this deliverable. We present preliminary results of threat analysis and risk assessment (TARA) based on a couple of automotive use cases. First, we briefly introduce the use cases. Second, we provide results for threat analysis. Third, we present risk assessment results. Fourth, we show how to derive security requirements for the uses cases under evaluation based on TARA. Finally, we perform comparative analysis of several risk assessment methodologies and summarize the findings.

5.1 Use cases

We have used seven use cases to perform TARA. However, we restrict ourselves to two uses cases for this version of the deliverable D2 (Version 2.0). In this sub-section, we briefly introduce the use cases – on-board diagnostics (OBD) and road speed limit (RSL). Please refer to the next revision of Deliverable D1.1 Needs and requirements, Version 2.0 [16] for more information about the use cases.

5.1.1 On-board diagnostics (OBD)

It is a very common use case within vehicles today. A vehicle will perform its own diagnostics and reporting if it detects it is in a faulty state. In order to do this, the vehicular system has what is called on-board diagnostics (OBD). The system basically has the ability to use its instrument cluster to request and present information. This is very useful in various situations such as requesting and presenting diagnostic trouble codes, software identification for the affected ECUs, etc. The main difference between this, wired diagnostics and remote diagnostics is that no diagnostics tool is needed to complete the process. Everything is done within the vehicular system.

5.1.2 Road speed limit (RSL)

Road Speed Limit is a functionality provided by the manufacturer to not let the vehicle go beyond a speed that is described by legislation or a fleet owner or even to enforce the user’s cruise control limitations. A speed sensor transmits the current vehicle speed signal to a tachograph in an encrypted form, which goes through a RSL ECU where the speed signal is converted to actual vehicular speed in accordance with the error-correction data which the tachograph provides. The RSL ECU performs a comparison with its current speed limit parameters and choses the lowest speed. It is then sent to the Engine ECU where a redundant check is performed with the RSL parameters and fuel supply is cut or maintained accordingly to ensure the speed limit. Figure 5-1 shows overview of the RSL use case.

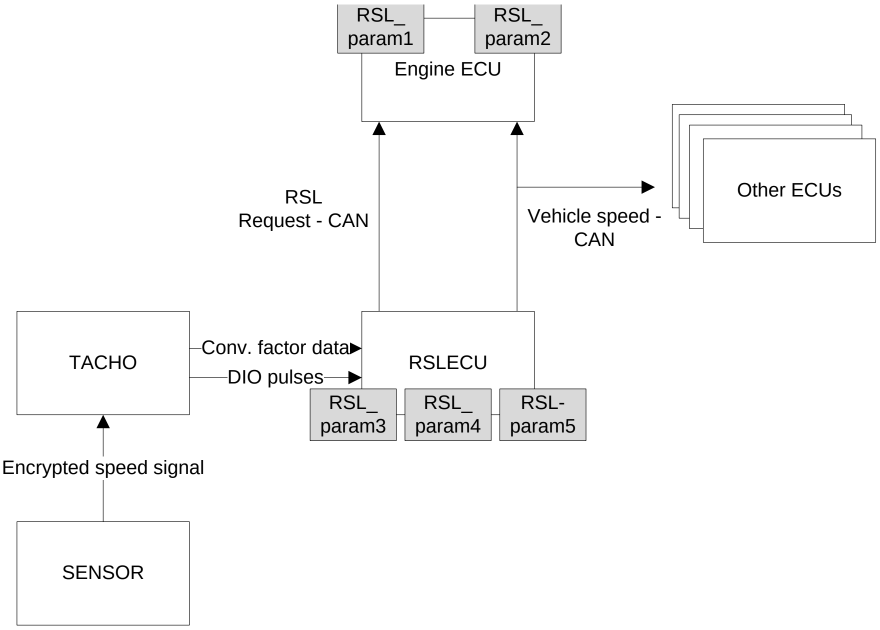


Figure 5-1: Overview of the RSL use case.

5.2 Threat analysis

We have adopted Microsoft STRIDE model [45] to perform threat analysis for the selected use cases. Based on the specific use cases and scenarios as described in Section 5.1 (Use cases) of this deliverable, we have first constructed the Data Flow Diagram (DFD) for each use case by using Microsoft Threat Modeling tool [38]. Then, for each use case, we have identified the assets and the threats associated with the assets. We present the results for on-board diagnostics (OBD) in Section 5.2.1 and for road speed limit (RSL) in Section 5.2.2.

5.2.1 Threat analysis for on-board diagnostics (OBD)

We have started threat analysis activities based on the high-level operational description of the on-board diagnostics (OBD) use case. We have created a DFD as shown in Figure 5-2.

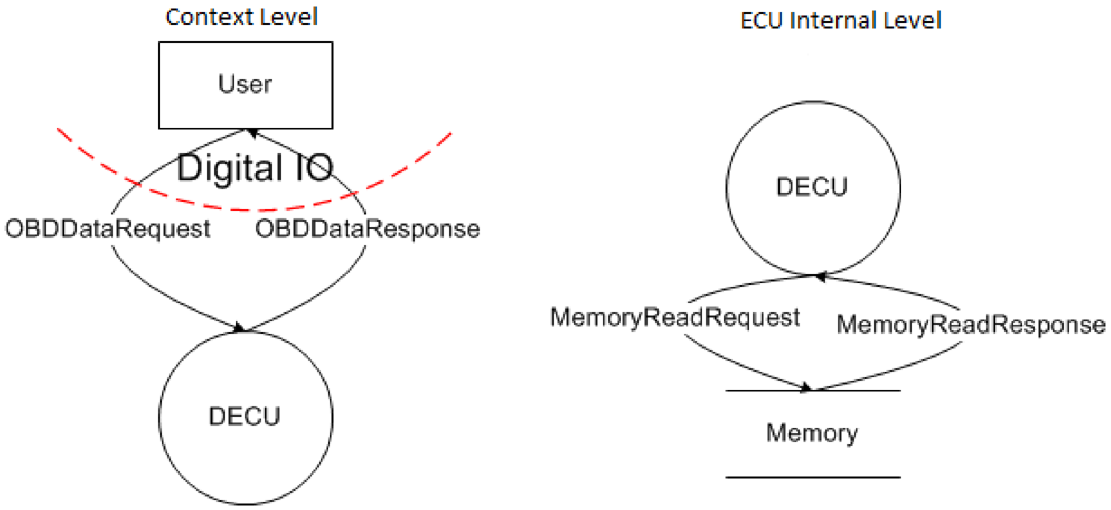
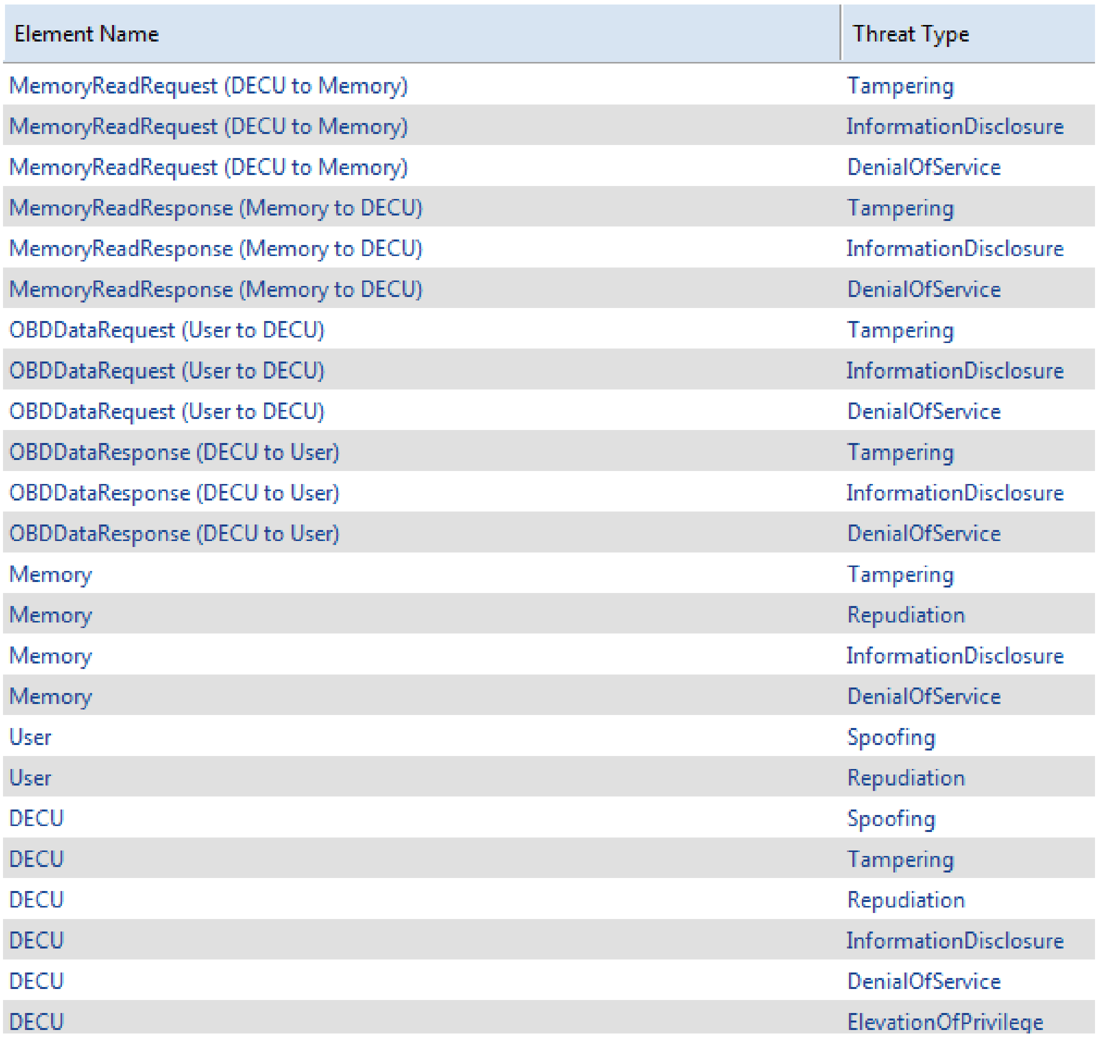


Figure 5-2: Data flow diagram of on-board diagnostics (OBD) use case.

The DFD consists of two different abstraction levels that are both shown in the same figure. Once the DFD is completed and no validation error is found, we have used the tool to analyze the DFD and to automatically generate the threats associated with the assets of the OBD use case. An extract from the identified threats are shown in Figure 5-3.



Element Name	Threat Type
MemoryReadRequest (DECU to Memory)	Tampering
MemoryReadRequest (DECU to Memory)	InformationDisclosure
MemoryReadRequest (DECU to Memory)	DenialOfService
MemoryReadResponse (Memory to DECU)	Tampering
MemoryReadResponse (Memory to DECU)	InformationDisclosure
MemoryReadResponse (Memory to DECU)	DenialOfService
OBDDataRequest (User to DECU)	Tampering
OBDDataRequest (User to DECU)	InformationDisclosure
OBDDataRequest (User to DECU)	DenialOfService
OBDDataResponse (DECU to User)	Tampering
OBDDataResponse (DECU to User)	InformationDisclosure
OBDDataResponse (DECU to User)	DenialOfService
Memory	Tampering
Memory	Repudiation
Memory	InformationDisclosure
Memory	DenialOfService
User	Spoofing
User	Repudiation
DECU	Spoofing
DECU	Tampering
DECU	Repudiation
DECU	InformationDisclosure
DECU	DenialOfService
DECU	ElevationOfPrivilege

Figure 5-3: Threats associated with the OBD use case.

5.2.2 Threat analysis for road speed limit (RSL)

We have then looked at the road speed limit (RSL) use case. The DFD, which is split up into two abstraction levels, is shown in Figure 5-4.

From the DFD of the RSL, one can easily see the increased complexity of this with respect to the OBD use case. However, as long as we are able to derive the correct DFD in relation to a particular use-case, we can use the tool to analyze the DFD to generate the threat report. Figure 5-5 shows an extract of the threats relevant to the RSL use case.

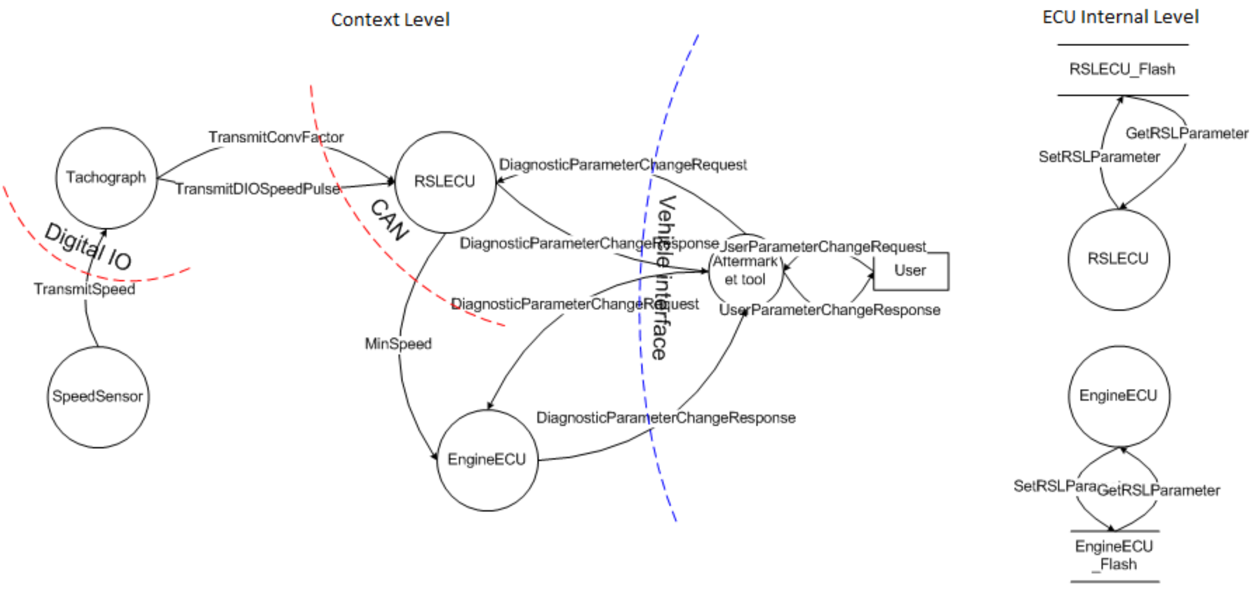
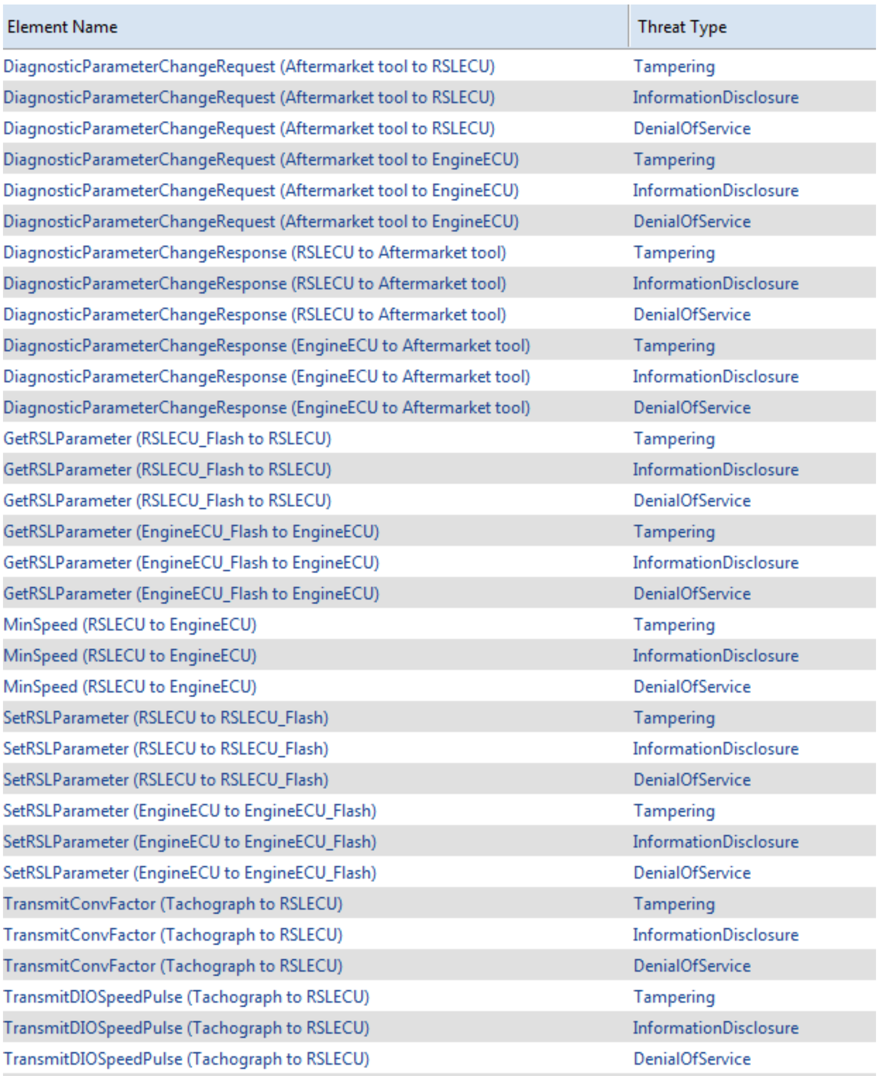


Figure 5-4: Data flow diagram of road speed limit (RSL) use case.



Element Name	Threat Type
DiagnosticParameterChangeRequest (Aftermarket tool to RSLECU)	Tampering
DiagnosticParameterChangeRequest (Aftermarket tool to RSLECU)	InformationDisclosure
DiagnosticParameterChangeRequest (Aftermarket tool to RSLECU)	DenialOfService
DiagnosticParameterChangeRequest (Aftermarket tool to EngineECU)	Tampering
DiagnosticParameterChangeRequest (Aftermarket tool to EngineECU)	InformationDisclosure
DiagnosticParameterChangeRequest (Aftermarket tool to EngineECU)	DenialOfService
DiagnosticParameterChangeResponse (RSLECU to Aftermarket tool)	Tampering
DiagnosticParameterChangeResponse (RSLECU to Aftermarket tool)	InformationDisclosure
DiagnosticParameterChangeResponse (RSLECU to Aftermarket tool)	DenialOfService
DiagnosticParameterChangeResponse (EngineECU to Aftermarket tool)	Tampering
DiagnosticParameterChangeResponse (EngineECU to Aftermarket tool)	InformationDisclosure
DiagnosticParameterChangeResponse (EngineECU to Aftermarket tool)	DenialOfService
GetRSLParameter (RSLECU_Flash to RSLECU)	Tampering
GetRSLParameter (RSLECU_Flash to RSLECU)	InformationDisclosure
GetRSLParameter (RSLECU_Flash to RSLECU)	DenialOfService
GetRSLParameter (EngineECU_Flash to EngineECU)	Tampering
GetRSLParameter (EngineECU_Flash to EngineECU)	InformationDisclosure
GetRSLParameter (EngineECU_Flash to EngineECU)	DenialOfService
MinSpeed (RSLECU to EngineECU)	Tampering
MinSpeed (RSLECU to EngineECU)	InformationDisclosure
MinSpeed (RSLECU to EngineECU)	DenialOfService
SetRSLParameter (RSLECU to RSLECU_Flash)	Tampering
SetRSLParameter (RSLECU to RSLECU_Flash)	InformationDisclosure
SetRSLParameter (RSLECU to RSLECU_Flash)	DenialOfService
SetRSLParameter (EngineECU to EngineECU_Flash)	Tampering
SetRSLParameter (EngineECU to EngineECU_Flash)	InformationDisclosure
SetRSLParameter (EngineECU to EngineECU_Flash)	DenialOfService
TransmitConvFactor (Tachograph to RSLECU)	Tampering
TransmitConvFactor (Tachograph to RSLECU)	InformationDisclosure
TransmitConvFactor (Tachograph to RSLECU)	DenialOfService
TransmitDIOSpeedPulse (Tachograph to RSLECU)	Tampering
TransmitDIOSpeedPulse (Tachograph to RSLECU)	InformationDisclosure
TransmitDIOSpeedPulse (Tachograph to RSLECU)	DenialOfService

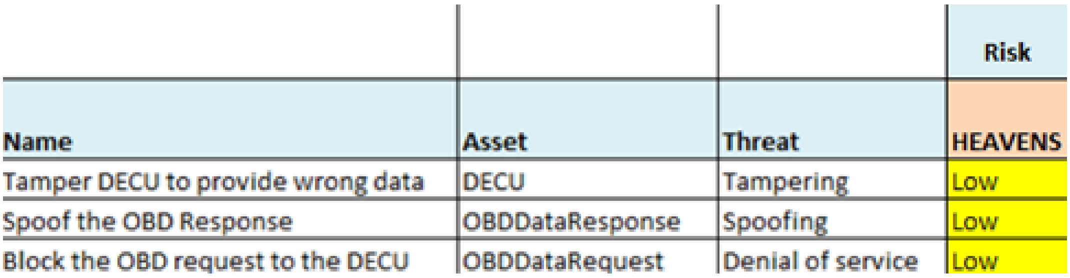
Figure 5-5: Threats associated with the RSL use case.

5.3 Risk assessment

In this sub-section, we present preliminary results of the HEAVENS risk assessment methodology. At the end of the risk assessment step, we get a security level for each asset-threat pair.

5.3.1 OBD use case

Figure 5-6 shows an extract from the results of HEAVENS Risk Assessment Methodology for the OBD use case. During our analysis, we end up at “Low” for threat level and “Medium” for impact level for each asset-threat pair (see Figure 5-6). This leads to a security level “Low” as per our analysis.

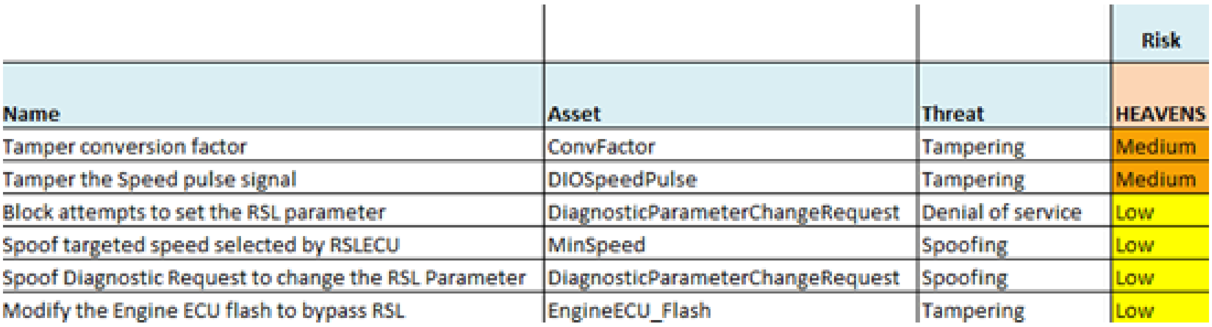


			Risk
Name	Asset	Threat	HEAVENS
Tamper DECU to provide wrong data	DECU	Tampering	Low
Spoof the OBD Response	OBDDataResponse	Spoofing	Low
Block the OBD request to the DECU	OBDDataRequest	Denial of service	Low

Figure 5-6: Risk rating of the OBD use case based on the HEAVENS methodology.

5.3.2 RSL use case

Figure 5-7 shows an extract of the results of HEAVENS Risk Assessment Methodology for the road speed limit (RSL) use case. During our analysis, we end up at “Medium” for the first two asset-threat pairs (ConvFactor-Tampering, DIOSpeedPulse-Tampering) for threat level and at “Low” for the other four asset-threat pairs. For impact level, we reach at “High” for all the asset-threat pairs because all can potentially lead to high operational damages. Eventually, we reach a security level of “Medium” for the first two rows (ConvFactor-Tampering, DIOSpeedPulse-Tampering) and security level of “Low” for the remaining four cases as shown in Figure 5-7.



			Risk
Name	Asset	Threat	HEAVENS
Tamper conversion factor	ConvFactor	Tampering	Medium
Tamper the Speed pulse signal	DIOSpeedPulse	Tampering	Medium
Block attempts to set the RSL parameter	DiagnosticParameterChangeRequest	Denial of service	Low
Spoof targeted speed selected by RSLECU	MinSpeed	Spoofing	Low
Spoof Diagnostic Request to change the RSL Parameter	DiagnosticParameterChangeRequest	Spoofing	Low
Modify the Engine ECU flash to bypass RSL	EngineECU_Flash	Tampering	Low

Figure 5-7: Risk rating of the RSL use case based on the HEAVENS methodology.

5.4 Security requirements

In this sub-section, we present examples of how to derive security requirements using the HEAVENS security model. Note that the derived security requirements are conceptually similar to the functional safety requirements that are derived during the concept phase as an end result of hazard analysis and risk assessment described in the ISO 26262.

5.4.1 OBD use case

We establish a mapping across asset, threat, security attribute, and security level for each of the asset-threat pair of the OBD use case as shown in Table 5-1. We then derive a security requirement for each row of the table.

Table 5-1: Asset, threat, security attribute and security level for the OBD use case.

No.	Asset	Threat	Security Attribute	Security Level
1	DECU	Tampering	Integrity	Low
2	OBDDataResponse	Spoofing	Authenticity	Low
3	OBDDataRequest	Denial of service	Availability	Low

Security Requirement 1

The DECU shall ensure integrity of the stored data.

Security Requirement 2

The authenticity of the OBDDataResponse signal shall be ensured.

Security Requirement 3

The authorized users shall be able to use the OBDDataRequest signal to extract information from the DECU whenever needed.

5.4.2 RSL use case

We establish a mapping across asset, threat, security attribute, and security level for each of the asset-threat pair of the RSL use case as shown in Table 5-2. We then derive a security requirement for each row of the table.

Table 5-2: Asset, threat, security attribute and security level for the RSL use case.

No.	Asset	Threat	Security Attribute	Security Level
1	ConvFactor	Tampering	Integrity	Medium
2	DIOSpeedPulse	Tampering	Integrity	Medium
3	DiagnosticParameterChangeRequest	Denial of service	Availability	Low
4	MinSpeed	Spoofing	Authenticity	Low
5	DiagnosticParameterChangeRequest	Spoofing	Authenticity	Low
6	EngineECU_Flash	Tampering	Integrity	Low

Security Requirement 1

The integrity of the ConvFactor signal shall be ensured.

Security Requirement 2

The integrity of the DIOSpeedPulse signal shall be ensured.

Security Requirement 3

The authorized users shall be able to use the DiagnosticParameterChangeRequest signal to set the RSL parameter whenever required.

Security Requirement 4

The authenticity of the MinSpeed signal shall be ensured.

Security Requirement 5

The authenticity of the DiagnosticParameterChangeRequest signal shall be ensured.

Security Requirement 6

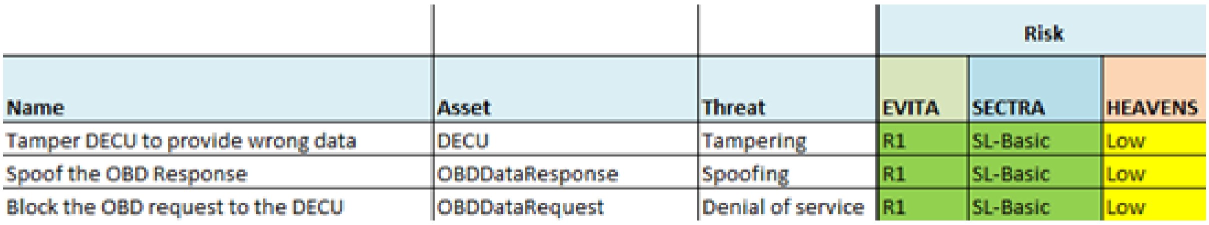
The integrity of the data stored in the flash of the engine ECU shall be ensured.

5.5 Comparison across various TARA methods

We compare EVITA, SECTRA and HEAVENS methodologies for the selected uses cases and present the results in this sub-section.

5.5.1 OBD use case

Figure 5-8 shows an extract from the comparative risk assessment results based on EVITA, SECTRA and HEAVENS methodologies for the OBD use case. In relation to the risk assessment, the results of SECTRA and EVITA are similar to HEAVENS; hence there is not a significant difference in risk levels among these three methodologies.

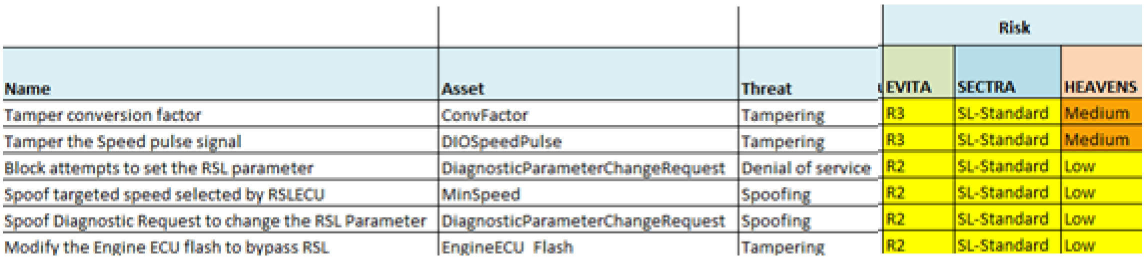


			Risk		
Name	**Asset**	**Threat**	**EVITA**	**SECTRA**	**HEAVENS**
Tamper DECU to provide wrong data	DECU	Tampering	R1	SL-Basic	Low
Spoof the OBD Response	OBDDataResponse	Spoofing	R1	SL-Basic	Low
Block the OBD request to the DECU	OBDDataRequest	Denial of service	R1	SL-Basic	Low

Figure 5-8: Comparative results of the risk assessment for the OBD use case.

5.5.2 RSL use case

Figure 5-9 shows an extract from the comparative risk assessment results based on EVITA, SECTRA and HEAVENS methodologies for the RSL use case. The RSL use case also shows great similarities across the different methodologies, with HEAVENS and EVITA being the most similar. The main difference between SECTRA and the rest is mainly caused by the threat level, where the SECTRA methodology kept a consistent Low rating over all the threats.



			Risk		
Name	Asset	Threat	EVITA	SECTRA	HEAVENS
Tamper conversion factor	ConvFactor	Tampering	R3	SL-Standard	Medium
Tamper the Speed pulse signal	DIOSpeedPulse	Tampering	R3	SL-Standard	Medium
Block attempts to set the RSL parameter	DiagnosticParameterChangeRequest	Denial of service	R2	SL-Standard	Low
Spoof targeted speed selected by RSLECU	MinSpeed	Spoofing	R2	SL-Standard	Low
Spoof Diagnostic Request to change the RSL Parameter	DiagnosticParameterChangeRequest	Spoofing	R2	SL-Standard	Low
Modify the Engine ECU flash to bypass RSL	EngineECU Flash	Tampering	R2	SL-Standard	Low

Figure 5-9: Comparative results for the risk assessment for the RSL use case.

We see from the figures (Figure 5-8 and Figure 5-9) that the EVITA risk ratings of “R1” and “R2” appear as “Low” in the HEAVENS model whereas the EVITA risk rating of “R3” appears as “Medium” in the HEAVENS model.

5.6 STRIDE-per-element and STRIDE-per-interaction comparison

This chapter largely depends on findings presented in a master’s thesis that has been conducted within the HEAVENS project during 2015, and are partially reproduced here with permission from the authors of the paper [68].

The thesis compared two variants of threat analysis against each other to see which the best fit for the automotive environment was in general, and AUTOSAR in particular. The two variants were STRIDE-per-elements and STRIDE-per-interaction, both from Microsoft.

5.6.1 STRIDE-per-Elements

The first version of STRIDE that Microsoft released was the STRIDE-per-Element variant. In this variant, every element in the dataflow diagram, DFD, is evaluated for threats.

Figure 5-10 shows an example of a Data Flow Diagram that will be used as a base for the examples in this section. It shows a process that communicates with a data store and a user.

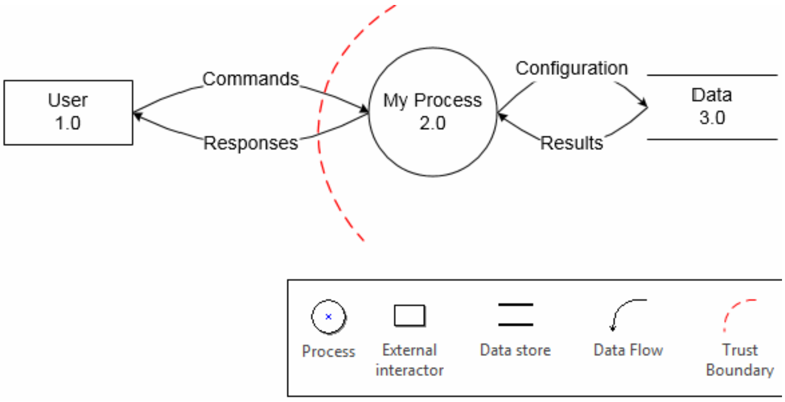
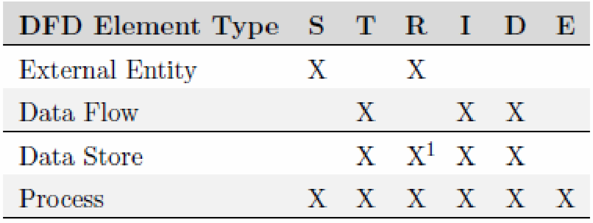


Figure 5-10 Example of an easy DFD model

Table 5-3 shows a mapping between the different elements of the DFD and the different categories in STRIDE. As shown not all types of elements are susceptible to all types of threats.

Table 5-3 Mapping of STRIDE to DFD Element types

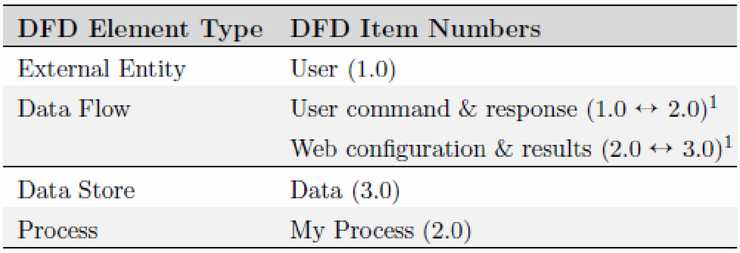


DFD Element Type	S	T	R	I	D	E
External Entity	X		X			
Data Flow		X		X	X	
Data Store		X	X[1]	X	X	
Process	X	X	X	X	X	X

[1] If the data store contains logging or audit data, repudiation is a potential threat, because if the data is manipulated, the attacker could cover his or her tracks [69]

After the DFD model of the system has been created, a list of all the elements in the diagram has to be created. Table 5-4 show the list of elements created from the DFD in Figure 5-10.

Table 5-4 Elements from DFD in Figure 5-10

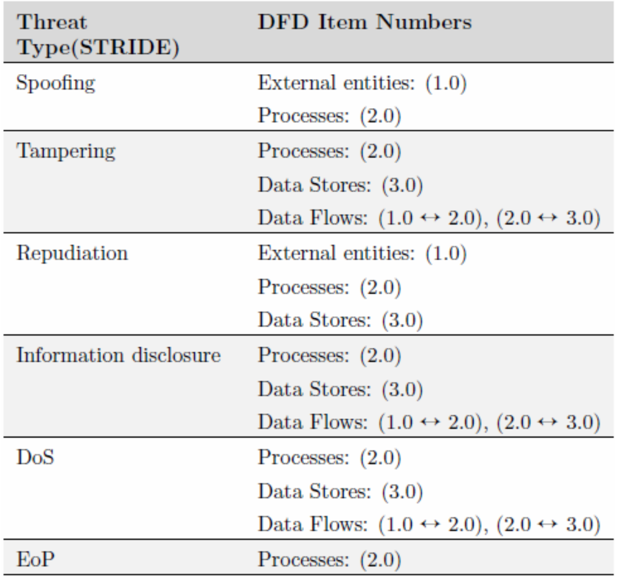


DFD Element Type	DFD Item Numbers
External Entity	User (1.0)
Data Flow	User command & response (1.0 ↔ 2.0)[1] Web configuration & results (2.0 ↔ 3.0)[1]
Data Store	Data (3.0)
Process	My Process (2.0)

[1] To reduce the number of entities in the list, the request and response have been combined. This can be done since the Data Flows is between the same elements and cross the same trust boundaries.

Once the list of DFD elements is done, STRIDE will be applied to each element in the list. However, not all types of threats have to be applied to all types of elements. To help with this, Table 5-3 can be used. In Table 5-5 the result of the STRIDE-per-element analysis can be seen. The threats have been grouped after the STRIDE categories. After STRIDE has been applied to the list of elements, it is time to calculate the risk attached to each threat.

Table 5-5 Threats to the model in Figure 5-10



Threat Type(STRIDE)	DFD Item Numbers
Spoofing	External entities: (1.0) Processes: (2.0)
Tampering	Processes: (2.0) Data Stores: (3.0) Data Flows: (1.0 ↔ 2.0), (2.0 ↔ 3.0)
Repudiation	External entities: (1.0) Processes: (2.0) Data Stores: (3.0)
Information disclosure	Processes: (2.0) Data Stores: (3.0) Data Flows: (1.0 ↔ 2.0), (2.0 ↔ 3.0)
DoS	Processes: (2.0) Data Stores: (3.0) Data Flows: (1.0 ↔ 2.0), (2.0 ↔ 3.0)
EoP	Processes: (2.0)

The advantage of STRIDE-per-element is that it is prescriptive; it helps to identify what to look for without having a checklist. When STRIDE-per-element is used by an experienced user, it can be useful for finding new types of weaknesses in components but can also find many common issues even though novices use it [70].

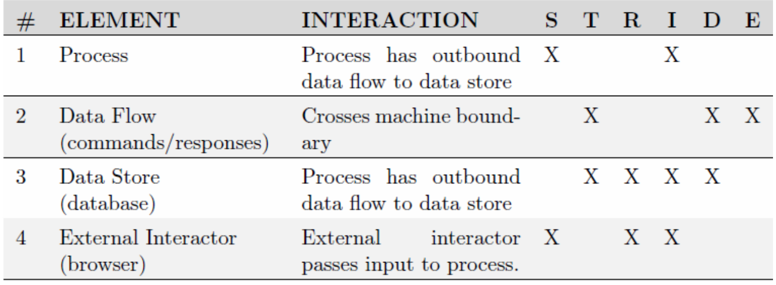
One weakness of STRIDE-per-element is that the same issue shows up in in several places in a model, for example if several elements are a part of the same attack. Another weakness is that Table 5-3 might be too general and not represent the issues in the specific project [70].

5.6.2 STRIDE-per-Interaction

The STRIDE-per-interaction approach was developed by Larry Osterman and Douglas MacIver. The meaning of this approach is that threat enumeration considers tuples such as origin, destination, interaction and the threats are enumerated against them. This approach had another goal during its development, which is to reduce the number of things that a modeler would have to consider [70]. However, STRIDE-per-element and STRIDE-per-interaction are expected to lead to the same number of threats but according to Shostack [70], the threats may be easier to understand with the STRIDE-per-interaction approach.

The STRIDE threats that are applicable to the interaction are also shown in Table 5-6. The difference between the two STRIDE variants are; the STRIDE-per-interaction approach is too complex without a reference chart handy, especially compared to STRIDE-per-element where the chart is easy enough to memorize and the approach, easy for beginners to understand [70].

Table 5-6 STRIDE-per-Interaction table: Threat Applicability [70]



#	ELEMENT	INTERACTION	S	T	R	I	D	E
1	Process	Process has outbound data flow to data store	X			X		
2	Data Flow (commands/responses)	Crosses machine boundary		X			X	X
3	Data Store (database)	Process has outbound data flow to data store		X	X	X	X	
4	External Interactor (browser)	External interactor passes input to process.	X		X	X		

5.6.3 Comparison

This section describes the factors that will be analyzed for the comparison of the STRIDE variants.

5.6.3.1 Quantitative comparison

The comparison was divided into several different categories. The number of relevant threats found was compared, as well as the number of irrelevant threats found, and the precision. The distribution of the threats into the STRIDE categories was also covered.

In addition the time spent performing the two variants of STRIDE were compared as well.

Precision is a measure of how good the STRIDE evaluation corresponds with the reality, i.e. what fractions of the threats found are relevant. The precision is calculated with the equation

$$P = TP / (TP + FP)$$

where TP are the true positives and FP is the false positives.

True and false positives refer to the number of identified threats that were correct/incorrect. Which threats that were categorized as what was decided after discussion with the domain expert.

5.6.3.2 Patterns

The threats found by the two variants of STRIDE are compared to find similarities and differences in the threats found. Patterns in the threats found are investigated. Both the true positives and the false positives will be compared.

Similarities & differences in True Positives and False Positives

This comparison is done by assembling a list of the advantages and disadvantages of the variants and comparing them with each other. The list contains the experiences of the two practitioners of STRIDE. Both variants were performed in the same system and used the same DFD as a base to give the best results of the study.

5.6.4 Experimental setup

The study is about two individuals each applying one of the two variants of STRIDE to the initial model of AUTOSAR, showed in Figure 5-11, and then comparing the variants to see what the differences are. The two individuals will not discuss their findings from the STRIDE analysis to not influence the results by giving each other ideas on what threats to look for or to ignore.

The same model will be used as a base for both STRIDE variants to keep the variations to a minimum. The model used is created from the AUTOSAR communication stack, with the new AUTOSAR security module called Secure Onboard Communication Module. It is also limited to only use the CAN network since it is the most widespread network in the automotive domain.

Secure Onboard Communication Module

In release 4.2, a module called Secure Onboard Communication (SecOC) was specified by AUTOSAR. This module was added to increase the security by adding authentication mechanisms for critical data. The module was designed to be resource efficient and to seamlessly integrate with the current communication systems in AUTOSAR.

The SecOC module works by using either Message Authentication Codes (MAC) or digital signatures of the messages to ensure that the received data is sent by the right ECU and contains the correct data.

In this study SecOC was configured to use Message Authentication Codes. And to create the MAC it was decided to use Hash-based Message authentication codes (HMAC). The SHA-256 hash was selected for the relative low resource use and high security.

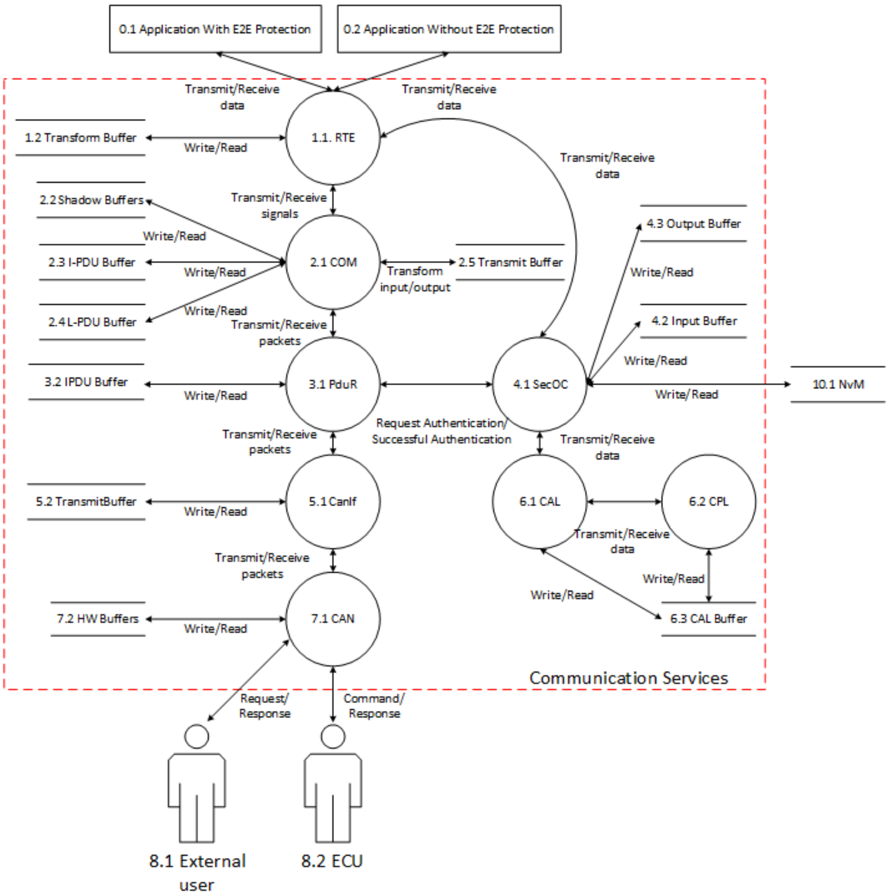


Figure 5-11 Model that the STRIDE analysis was based on

5.6.5 Quantitative comparison

As shown in Table 5-7 a total of 99 threats were found with STRIDE-per-element. 45 of the threats were false positive and 54 threats were true positives. Overall the precision was 54.55%. The main part of the threats was found in the Tampering and Denial of service categories, while no threats were found in the Elevation of privilege category.

The result of STRIDE-per-interaction is shown in Table 5-8, 114 threats were found in total. 83 threats were false positive while 31 threats were true positive. Total precision was 27.19% and threats were found for each category but no true positive threats were found in Elevation of privilege.

Based on the above results, STRIDE-per-element and STRIDE-per-interaction and about the same amount of threats, but STRIDE-per-element has significantly higher precision.

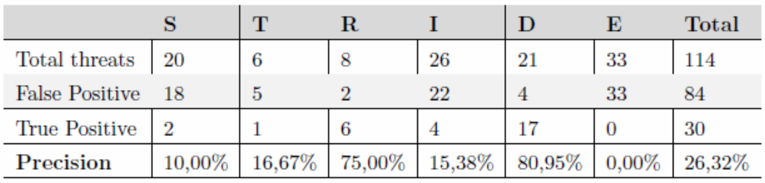
In Figure 11 the precision of STRIDE-per-element and STRIDE-per-interaction is compared. It can be seen that in STRIDE-per-element, most threats were found in the denial of service category, while in the STRIDE-per-interaction the threats were more evenly spread, but still with a large amounts of threats in the denial of service category.

Table 5-7 Descriptive statistics for STRIDE-per-element

	S	T	R	I	D	E	Total
Total threats	3	31	4	16	45	0	99
False Positive	0	30	2	13	0	0	45
True Positive	3	1	2	3	45	0	54
Precision	100,00%	3,23%	50,0%	18,75%	100,0%	100,00%	54,55%

As shown in Table 5-8 the overall precision is low for most categories. The most outstanding categories are spoofing (in the case of STRIDE-per-element), repudiation and Denial-of-service where the precision is high. The category with the biggest differences between the two STRIDE variants is spoofing and elevation of privilege, where STRIDE-per-element has 100% while STRIDE-per-element has 10% and 0%.

Table 5-8 Descriptive statistics for STRIDE-per-interaction



	S	T	R	I	D	E	Total
Total threats	20	6	8	26	21	33	114
False Positive	18	5	2	22	4	33	84
True Positive	2	1	6	4	17	0	30
Precision	10,00%	16,67%	75,00%	15,38%	80,95%	0,00%	26,32%

Figure 5-12 shows the true positives for both versions of STRIDE. The biggest differences are in the Repudiation and Denial of service categories. In the repudiation category STRIDE-per-interaction found more true threats, while in the denial of service category STRIDE-per-element found more true threats.

Figure 5-12 compares the distribution of false positives between the two STRIDE variants. This is the category where the biggest difference between the two variants can be found. STRIDE-per-interaction found much more false positives in all categories except tampering. The false positives found by STRIDE-per-element was almost exclusively found in the tampering and information disclosure categories, while STRIDE-per-interaction had its false positives mostly in the spoofing, information disclosure and elevation of privilege categories.

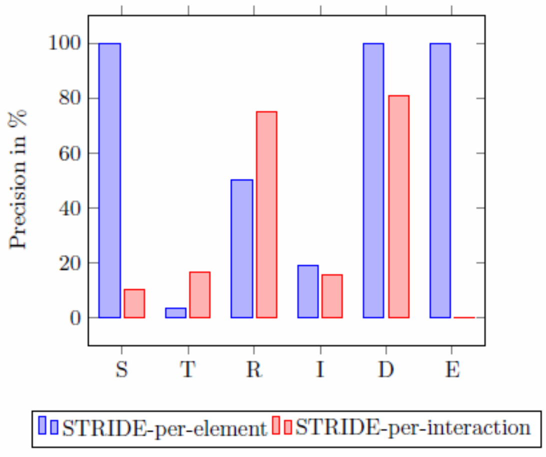


Figure 5-12 Comparison of the distribution of precision across the STRIDE categories.

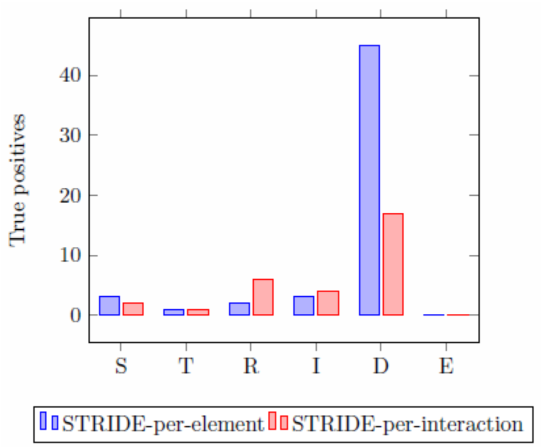


Figure 5-13 Comparison of True positives between STRIDE-per-element and STRIDE-per-interaction

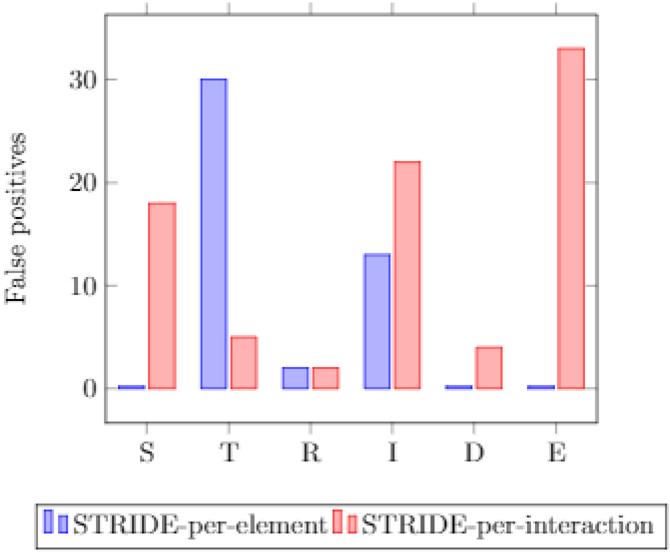
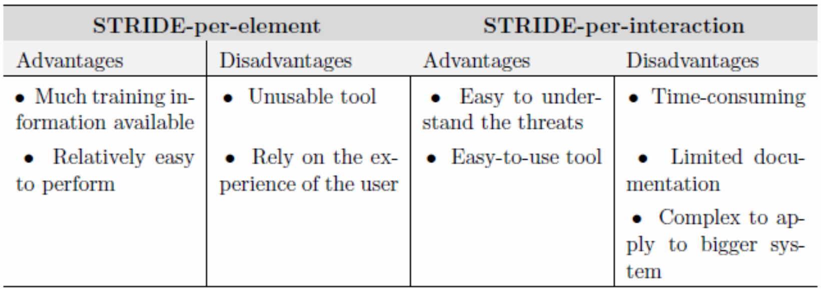


Figure 5-14 Comparison of false positives between STRIDE-per-element and STRIDE-per-interaction

Based on the results of applying the STRIDE variants shown in Table 5-9, STRIDE-per-interaction is better if the outcome needs to be understood by non-security experts but there is limited information about this variant and no other examples to follow than the book written by Shostack [70]. It is also time consuming and complex to apply STRIDE-per-interaction since each interaction to be filled into a table.

Table 5-9 Advantages and disadvantages of the STRIDE variants



STRIDE-per-element		STRIDE-per-interaction	
Advantages	Disadvantages	Advantages	Disadvantages
• Much training information available • Relatively easy to perform	• Unusable tool • Rely on the experience of the user	• Easy to understand the threats • Easy-to-use tool	• Time-consuming • Limited documentation • Complex to apply to bigger system

Similarities & differences in True Positives and False Positives

The threat descriptions are similar in both STRIDE variants even though STRIDE-per-element is based on brainstorming and STRIDE-per-interaction is based on the Microsoft Threat Modeling Tool 2014.

True positive

The true positives were similar between the two STRIDE variants. Most of the true positives threats were focused on the access to the memory, to read or change the keys used for the SecOC authentication mechanism, or to overburden the ECU to make it crashes.

False positive

The biggest difference between the two STRIDE variants is where the false positives were found. For STRIDE-per-element, the bulk of the false positives were found in the tampering and information disclosure categories, while STRIDE-per-interaction had most of the false positives in the spoofing, information disclosure and elevation of privilege categories.

The types of threats found differed as well. In the information disclosure category, the threats found by STRIDE-per-element were focused on disclosure of the key used for the SecOC authentication mechanism, while STRIDE-per-interaction had a more broad view and the threats were concerning all types of information handled by the ECU.

5.6.6 Conclusion

Based on the statistics and the advantages & disadvantages from earlier in this section, STRIDE-per-element was found to be better suited for use in the automotive domain and AUTOSAR. The precision for STRIDE-per-element is 54.55% compared to 26.32% for STRIDE-per-interaction.

5.7 Summary

In relation to the threat analysis, Microsoft's SDL threat modeling tool works quite well, but it is important to do evaluation to validate and ensure that the tool identifies all the threats. After creating the models, we have performed our own analysis and determined the threats that may present. During this analysis, we have decided not to cross reference any threats from the tool's analysis so that we could be thorough and complete. A number of threats that have come up were either directly or indirectly related to all threats within STRIDE. For example, threats such as spoofing, tampering, repudiation etc. have been found manually in conjunction with what has been found by the tool. Also, we have observed that the other threats can potentially be categorized into as per STRIDE model. So we believe that STRIDE approach and the threat modeling tool work well enough to cross reference with manual approach of analyzing threats.

Microsoft SDL Threat Modeling tool provides a straight-forward way of drawing dataflow diagrams, which not only provides good visualization, but allows the modeler to work at different abstraction levels and perform certain rating in the tool. For example, in the RSL and OBD examples in previous chapters, all threats related to Information Disclosure were tagged in the tool as not being important. It is also possible to add parts in the diagram that are informational only and not part of the threat report, in order to aid understanding of the system being modeled.

The risks rating results for the selected use cases show similar results across the three methodologies: EVITA, SECTRA and HEAVENS. Other use cases perhaps would have shown greater differences, but the similarities in output shows that the HEAVENS methodology is comparable with, for the automotive industry, well-known EVITA methodology, while adding benefits like clearer guidelines, better alignment with industry standards and being more versatile, to mention a few.

6. Concluding remarks

This section summarizes the findings and contributions of the WP2 Security models of the HEAVENS project as well as points to potential future works in relation to the HEAVENS security model.

The findings and contributions of deliverable D2 security models (Version 2.0) are as follows:

- It provides a general overview of different types of security models and their intrinsic differences. The difficulties of model evaluation and security metrics are also discussed to some extent. It presents a critical review of the state-of-the-art security models with focus on threat analysis and risk assessment, considering various domains, for example, IT security, software engineering, telecommunications, and defense.
- It proposes a new security model – HEAVENS security model – to perform threat analysis and risk assessment for the automotive E/E systems. The proposed model discusses threat analysis and risk assessment methodology along with tool support. The model is equally applicable to both passenger cars and commercial vehicles. The threat analysis and risk assessment method facilitates identifying security level for a particular asset-threat pair similar to the concept of ASIL in the ISO 26262 as well as deriving security requirements by combining asset, threat, security attribute and security level.
- It describes the HEAVENS security model in the context of the existing standards, for example, Common Criteria for IT security evaluation and ISO 26262 for functional safety for E/E systems of the road vehicles. This aims at supporting better understanding of the automotive security issues and opening up the opportunities of reusing the relatively established and well-known processes from the other field of studies (IT security, functional safety).
- It investigates the applicability of Microsoft’s STRIDE model for threat analysis and Microsoft’s Threat Modeling tool to realize the STRIDE model. While STRIDE is primarily developed to deal with threats in a software system, we observe that STRIDE in general is applicable to the automotive E/E systems to identify assets and threats. Furthermore, the tool support works well for the automotive use cases that we have considered in this deliverable.
- It presents a proof-of-concept implementation and evaluation of the proposed security model by using a couple of automotive use cases – on-board diagnostics (OBD) and road-speed limit (RSL). The results suggest that the proposed model is comparable with the existing methodologies with added benefits like clearer guidelines, better alignment with industry standards and being more versatile, to mention a few.

Potential future works include:

- To perform a comprehensive evaluation and validation of the model using a wide variety of use cases related to the automotive E/E systems and fine-tune the model, if necessary.
- To suggest countermeasures and/or security mechanisms according to the derived security level to fulfill the derived security requirements.
- To establish relationship between threats and vulnerabilities.

- To refine high-level security requirements to facilitate deriving software security requirements and hardware security requirements, and to allocate the security requirements to the elements of the TOE in a systematic way.

Acknowledgements

HEAVENS (Dnr 2012-04625) is a 3-year research project started in April 2013. The project is coordinated by Volvo AB and partly funded (50% public funding) by Vinnova (the Swedish Governmental Agency for Innovation Systems) within the scope of the Vehicle Development Program of Strategic Vehicle Research and Innovation (Sv: Fordonsstrategisk Forskning och Innovation) (FFI). FFI is a joint effort between the Government (Vinnova, the Swedish Road Administration, and the Swedish Energy Agency) and the Swedish Automotive industry (Volvo AB, Volvo Cars Corporation, Scania AB and FKG).

This page is intentionally left blank

References

[1] R. J. Anderson. *Security Engineering: A Guide To Building Dependable Distributed Systems*. Wiley, 2008 (cit. on p. 13).

[2] L. Apvrille and Y. Roudier. 'Towards the Model-Driven Engineering of Secure yet Safe Embedded Systems'. In: *arXiv preprint arXiv:1404.1985* (2014) (cit. on p. 19).

[3] A. Avivzienis, J.-C. Laprie, B. Randell and C. Landwehr. 'Basic Concepts and Taxonomy of Dependable and Secure Computing'. In: *IEEE Transactions on Dependable and Secure Computing* 1.1 (2004), pp. 11–33. DOI: 10.1109/TDSC.2004.2 (cit. on p. 19).

[4] CAR 2 CAR Communication Consortium. *CAR 2 CAR Communication Consortium*. URL: http://www.car-to-car.org/ (visited on 13th Nov. 2013) (cit. on p. 54).

[5] CCRA Members. *Common Criteria for Information Technology Security Evaluation*. CCMB-2012-09-00X, Version 3.1, Revision 4. Common Criteria (cit. on p. 38).

[6] CCRA Members. *Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model*. CCMB-2012-09-001, Version 3.1, Revision 4. Common Criteria (cit. on pp. 10, 20, 35, 36, 38–40, 54).

[7] CCRA Members. *Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components*. CCMB-2012-09-003, Version 3.1, Revision 4. Common Criteria (cit. on p. 20).

[8] CCRA Members. *Common Methodology for Information Technology Security Evaluation – Evaluation Methodology*. CCMB-2012-09-004, Version 3.1, Revision 4. Common Criteria (cit. on p. 40).

[9] ETSI. *Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA)*. Tech. Rep. TR 102 893, v1.1.1. 650 Route des Locioles, F-06921 Sophia Antipolis Cedex, France: ETSI, Mar. 2010 (cit. on p. 33).

[10] ETSI. *Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Methods and protocols; Part 1: Method and proforma for Threat, Risk, Vulnerability Analysis*. Tech. Spec. TS 102 165-1, v4.2.3. 650 Route des Locioles, F-06921 Sophia Antipolis Cedex, France: ETSI, Mar. 2011 (cit. on pp. 33, 34, 38, 40).

[11] EVITA Project. *E-safety Vehicle Intrusion Protected Applications (EVITA)*. URL: http://www.evitaproject.org/ (visited on 25th Nov. 2013) (cit. on pp. 7, 30, 38).

[12] B. Fabian, S. Gürses, M. Heisel, T. Santen and H. Schmidt. 'A comparison of security requirements engineering methods'. In: *Requirements engineering* 15.1 (2010), pp. 7–40 (cit. on p. 19).

[13] M. Felderer, B. Katt, P. Kalb, J. Jürjens, M. Ochoa, F. Paci, L. M. S. Tran, T. T. Tun, K. Yskout, R. Scandariato et al. 'Evolution of Security Engineering Artifacts: A State of the Art Survey'. In: *International Journal of Secure Software Engineering* 5.4 (2014) (cit. on p. 19).

[14] D. G. Firesmith. *Common Concepts Underlying Safety Security and Survivability Engineering*. en. Tech.rep. Dec. 2003 (cit. on pp. 6, 20).

[15] HEAVENS. *HEAVENS: HEAling Vulnerabilities to ENhance Software Security and Safety – Project Proposal*. Tech. rep. Version 3.0. Dec. 2012 (cit. on p. 1).

[16] M. Islam, C. Sandberg, M. Olsson, A. Bokesand, T. Olovsson, H. Broberg et al. *Deliverable D1.1 Needs and Requirements*. HEAVENS Project, Deliverable D1.1, Release 2. Mar. 2016 (cit. on pp. 5, 36, 37, 44,57, 58).

[17] ISO. *ISO/IEC 27005: Information technology — Security techniques — Information security risk management*. ISO/IEC 27005:2011. International Organization for Standardization, 2011 (cit. on pp. 14–16).

[18] ISO. *ISO/IEC 31000: Risk Management – Principles and guidelines*. ISO/IEC 31000:2009. International Organization for Standardization, 2009 (cit. on pp. 13, 14).

[19] ISO (International Organization for Standardization). *Road vehicles—Functional safety (ISO 26262:2011)*. ISO 26262:2011. 2011 (cit. on pp. 17, 30, 40, 42, 44).

[20] ISO (International Organization for Standardization). *Road vehicles — Functional safety — Part 3: Concept phase (ISO 26262-3:2011)*. ISO 26262-3:2011. 2011 (cit. on p. 18).

[21] E. Jonsson. 'Dependability and Security: Modelling, Metrics and Evaluation'. Lecture in Computer Security, Chalmers University of Technology. 2012 (cit. on p. 13).

[22] E. Jonsson. 'Towards an integrated conceptual model of security and dependability'. In: *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*. IEEE. 2006, pp. 646–653 (cit. on p. 13).

[23] A. A. Kaposi. 'Software Engineer's Reference Book'. In: ed. by J. A. McDermid. Butterworth-Heinemann Ltd, 1991. Chap. Measurement theory, pp. 12/3–12/19 (cit. on pp. 5, 19, 20).

[24] A. Kiening, D. Angermeier, H. Seudie, T. Stodart and M. Wolf. 'Trust assurance levels of cybercars in v2x communication'. In: *Proceedings of the 2013 ACM workshop on Security, privacy & dependability for cyber vehicles*. ACM. 2013, pp. 49–60 (cit. on p. 54).

[25] M. Line, O. Nordland, L. Røstad and I. Tøndel. 'Safety vs. security'. In: *Probabilistic Safety Assessment and Management (PSAM), Proceedings of the 8th international Conference on*. IAPSAM. 2006, pp. 685–699 (cit. on p. 6).

[26] B. B. Madan, K. Gogeva-Popstojanova, K. Vaidyanathan and K. S. Trivedi. 'Modeling and quantification of security attributes of software systems'. In: *Dependable Systems and Networks, 2002. DSN 2002. Proceedings. International Conference on*. IEEE. 2002, pp. 505–514 (cit. on p. 19).

[27] P. Mell, K. A. Kent and S. Romanosky. *The Common Vulnerability Scoring System (CVSS) and its applicability to federal agency systems*. Citeseer, 2007 (cit. on p. 27).

[28] P. Mell, K. Scarfone and S. Romanosky. 'A complete guide to the common vulnerability scoring system version 2.0'. In: *Published by FIRST-Forum of Incident Response and Security Teams*. 2007, pp. 1–23 (cit. on pp. 28, 29).

[29] Microsoft. *Simplified implementation of the Microsoft SDL*. URL: http://www.microsoft.com/enus/download/details.aspx?id=12379 (visited on 29th June 2014) (cit. on pp. 25, 26).

[30] S. Myagmar, A. J. Lee and W. Yurcik. 'Threat modeling as a basis for security requirements'. In: *Symposium on requirements engineering for information security (SREIS)*. Vol. 2005. 2005, pp. 1–8 (cit. on pp. 13, 23).

[31] D. K. Nilsson, P. H. Phung and U. E. Larson. 'Vehicle ECU classification based on safety-security characteristics'. In: *Road Transport Information and Control - RTIC 2008 and ITS United Kingdom Members' Conference, IET*. May 2008, pp. 1–7 (cit. on p. 19).

[32] OVERSEE Project. *Open Vehicular Secure Platform*. URL: https://www.oversee-project.com/ (visited on 6th Aug. 2011) (cit. on p. 7).

[33] OWASP. *OWASP Risk Rating Methodology*. URL: https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology (visited on 29th June 2014) (cit. on pp. 29, 30, 38).

[34] OWASP. *OWASP Threat Risk Modeling*. URL: https://www.owasp.org/index.php/Threat_Risk_Modeling (visited on 13th Sept. 2014) (cit. on pp. 10, 11).

[35] L. Piètre-Cambacédès and M. Bouissou. 'Cross-fertilization between safety and security engineering'. In: *Reliability Engineering & System Safety* 110 (2013), pp. 110–126 (cit. on pp. 15, 17).

[36] L. Piètre-Cambacédès and C. Chaudet. 'The SEMA referential framework: avoiding ambiguities in the terms "security" and "safety"'. In: *International Journal of Critical Infrastructure Protection* 3.2 (2010), pp. 55–66 (cit. on pp. 6, 7).

[37] L. Pirzadeh and E. Jonsson. 'A cause and effect approach towards risk analysis'. In: *Security Measurements and Metrics (Metrisec), 2011 Third International Workshop on*. IEEE. 2011, pp. 80–83 (cit. on p. 15).

[38] B. Potter. 'Microsoft SDL Threat Modelling Tool'. In: *Network Security* 2009.1 (2009), pp. 15–18. ISSN: 1353-4858. DOI: http://dx.doi.org/10.1016/S1353- 4858(09)70008- X. URL: http://www.sciencedirect.com/science/article/pii/S135348580970008X (cit. on p. 37).

[39] PRESERVE. *Preparing Secure V2X Communication Systems (PRESERVE)*. URL: http://www.preserveproject.eu/ (visited on 5th Dec. 2013) (cit. on p. 7).

[40] Y. Roudier, M. S. Idrees and L. Apvrille. 'Towards the model-driven engineering of security requirements for embedded systems'. In: *Model-Driven Requirements Engineering (MoDRE), 2013 International Workshop on*. IEEE. 2013, pp. 55–64 (cit. on p. 19).

[41] A. Ruddle, D. Ward, B. Weyl, S. Idrees, Y. Roudier, M. Friedewald et al. *Security requirements for automotive on-board networks based on dark-side scenarios*. EVITA Project, Deliverable D2.3, v1.1. Dec. 2009 (cit. on pp. 11, 30, 31).

[42] K. Scarfone and P. Mell. ‘An analysis of CVSS version 2 vulnerability scoring’. In: *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE Computer Society. 2009, pp. 516–525 (cit. on p. 27).

[43] SeVeCOM Project. *Secure Vehicle Communication (SeVeCOM)*. URL: http://www.sevecom.org (visited on 25th July 2012) (cit. on p. 7).

[44] G. Sindre and A. L. Opdahl. ‘Eliciting security requirements with misuse cases’. In: *Requirements engineering* 10.1 (2005), pp. 34–44 (cit. on p. 19).

[45] F. Swiderski and W. Snyder. *Threat modeling*. Microsoft Press, 2004 (cit. on pp. 24, 37, 60).

[46] M. A. Tariq, J. Brynielsson and H. Artman. ‘Framing the Attacker in Organized Cybercrime’. In: *Intelligence and Security Informatics Conference (EISIC), 2012 European*. IEEE. 2012, pp. 30–37 (cit. on p. 19).

[47] K. S. Trivedi, D. S. Kim, A. Roy and D. Medhi. ‘Dependability and security models’. In: *Design of Reliable Communication Networks, 2009. DRCN 2009. 7th International Workshop on*. IEEE. 2009, pp. 11–20 (cit. on p. 19).

[48] V. Verendel. ‘Quantified Security is a Weak Hypothesis: A Critical Survey of Results and Assumptions’. In: *Proceedings of the 2009 Workshop on New Security Paradigms Workshop*. NSPW ’09. Oxford, United Kingdom: ACM, 2009, pp. 37–50. ISBN: 978-1-60558-845-2. DOI: 10.1145/1719030.1719036. URL: http://doi.acm.org/10.1145/1719030.1719036 (cit. on pp. 19, 20).

[49] J. Viega and G. McGraw. *Building secure software: how to avoid security problems the right way*. Pearson Education, 2001 (cit. on p. 15).

[50] A. J. A. Wang. ‘Information Security Models and Metrics’. In: *Proceedings of the 43rd Annual Southeast Regional Conference - Volume 2*. ACM-SE 43. Kennesaw, Georgia: ACM, 2005, pp. 178–184. ISBN: 1-59593-059-0. DOI: 10.1145/1167253.1167295. URL: http://doi.acm.org/10.1145/ 1167253.1167295 (cit. on p. 20).

[51] J. Zalewski, S. Drager, W. Mckeever and A. Kornecki. ‘Can we measure security and how?’ In: *Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research*. CSIIRW ’11. New York, NY, USA: ACM, 2011, 46:1–46:1. ISBN: 978-1-4503-0945-5. DOI: 10.1145/2179298.2179348. URL: http://doi.acm.org/10.1145/2179298.2179348 (visited on 16th Aug. 2013) (cit. on p. 20).

[52] ISO/IEC 27000, Information technology — Security techniques — Information security management systems — Overview and vocabulary, Second Edition, December 2012.

[53] Deliverable D2.3, Security requirements for automotive on-board networks based on dark-side scenarios, EVITA (E-Safety Vehicle Intrusion Protected Applications), http://www.evita-project.org/, December 2009.

[54] National Information Assurance (IA) Glossary, CNSS Instruction No. 4009, 26 April 2010, Committee on National Security Systems, USA.

[55] Cherdantseva Y. and Hilton J. "A Reference Model of Information Assurance & Security," SecOnt 2013 workshop in conjunction with the 8th International Conference on Availability, Reliability and Security (ARES) 2013, University of Regensburg, Germany. September 2nd - 6th, 2013. IEEE Proceedings.

[56] Schmidt, K., Tröger, P., Kroll, H., Bünger, T. et al., "Adapted Development Process for Security in Networked Automotive Systems," SAE Int. J. Passeng. Cars – Electron. Electr. Syst. 7(2):2014, doi: 10.4271/2014-01-0334.

[57] Christey, Steve. "PLOVER: Preliminary list of vulnerability examples for researchers." NIST Workshop Defining the State of the Art of Software Security Tools. 2005.

[58] Johnston, R., “Being Vulnerable to the Threat of Confusing Threats with Vulnerabilities”, Viewpoint Paper. Journal of Physical Security, 4:30-34, 2010.

[59] BSI-Standard 100-4, Version 1.0, 2009, Federal Office for Information Security (BSI), Germany.

[60] Automotive Industry Action Group (AIAG), “Potential Failure Mode and Effects Analysis (FMEA)”, 2008.

[61] Saitta, Paul, Brenda Larcom, and Michael Eddington. "Trike v. 1 methodology document [draft]." URL: http://dymaxion. org/trike/Trike_v1_Methodology_Documentdraft.pdf, 2005, Last accessed: September 24, 2014.

[62] Alberts, Christopher, et al. "Introduction to the OCTAVE Approach." Pittsburgh, PA, Carnegie Mellon University, 2003.

[63] Tony UcedaVelez, “Real World Threat Modeling Using the PASTA Methodology”, OWASP AppSec EU 2012.

[64] Anders Hansson, “Security Model - Introduction”, Internal Report, SECTRA AB, Sweden, 2014.

[65] SHIELDS. Detecting known security vulnerabilities from within design and development tools, URL: http://www.shields-project.eu/, Last accessed on September 24, 2014.

[66] Lisa Boran, Committee Chair of the Security Guidelines and Risk Management Task Force (TEVEES18A) presentation title: “Automotive Cyber-Security”, ESCAR Conf., USA, July 2014.

[67] “Privacy Impact Assessment Guideline”, 2011, Federal Office for Information Security (BSI), Germany.

[68] A. Bretting, M. Ha, Chalmers University of Technology, “Vehicle Control Unit Security using Open Source AUTOSAR”, Master’s Thesis, 2015

[69] M. Howard and S. Lepner, The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software (Developer Best Practices). Microsoft Press, 2006.

[70] A. Shostack, Threat Modeling: Designing for Security. US: John Wiley & Sons Ltd, 2014.

[71] M. Wolf and M. Scheibel. A systematic approach to a qualified security risk analysis for vehicular IT systems. In E. Pl□odereder, P. Dencker, H. Klenk, H. B. Keller, and S. Spitzer, editors, Automotive - Safety & Security 2012, Lecture Notes in Informatics, pages 195 - 210.

[72] O. Henniger, L. Apvrille, A. Fuchs, Y. Roudier, A. Ruddle, and B. Weyl. Security requirements for automotive on-board networks. In Proceedings of the 9th International Conference on Intelligent Transport System Telecommunications (ITST 2009), Lille, France, 2009.

[73] M. Islam, et al, “A Risk Assessment Framework for Automotive Embedded Systems”, CPSS’16, May 30-June 03 2016, Xi’an, China