CHALMERS
UNIVERSITY OF TECHNOLOGY

Attack Analysis Methodologies

Attack Analysis Methodologies in the Automotive Industry

Master's thesis in Computer Systems and Networks

Seyed Reza Esmaeili & Afshin Soltani Esterabadi

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Master's thesis 2019

Attack Analysis Methodologies

Attack Analysis Methodologies in the Automotive Industry

Seyed Reza Esmaeili
Afshin Soltani Esterabadi

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Division of Computer Systems and Networks
Chalmers University of Technology
Gothenburg, Sweden 2019

Attack Analysis Methodologies
Attack Analysis Methodologies in the Automotive Industry
SEYED REZA ESMAEILI & AFSHIN SOLTANI ESTERABADI

Supervisor: Erland Jonsson, Department of Computer Science and Engineering

Advisors: Christian Sandberg & Manne Engelke, Volvo Group Trucks Technology

Examiner: Tomas Olovsson, Department of Computer Science and Engineering

Master's Thesis 2019
Department of Computer Science and Engineering
Division of Computer Systems and Networks
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2019

Attack Analysis Methodologies
Attack Analysis Methodologies in the Automotive Industry
SEYED REZA ESMAEILI & AFSHIN SOLTANI ESTERABADI
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

In recent years, we have witnessed that technology has advanced dramatically. While new, hi-tech, automated devices entered our lives, a tendency of moving from the disjointed nature of objects to a more interconnected world has emerged. Although such need of interconnection was originated in the IT industry and with the Internet of Things (IoT), automotive industry was also affected by such a trend. Connected, electric, highly-automated and autonomous vehicles are making their way into our lives. As a result of this paradigm shift, new security challenges are introduced in the automotive industry.

Vehicles are comprised of tens or sometimes a hundred of computers, also known as Electronic Control Units (ECUs) that need to communicate and be interconnected in order for the vehicle to function properly. Protecting vehicles from potential threats and attacks that may compromise the security and consequently the safety of both the vehicle and the passenger is of great importance. Hence, a comprehensible attack analysis methodology is needed to model the possible attacks in vehicles.

Attack analysis is part of the risk assessment process. To have an accurate risk analysis, two factors are needed: first, the impact of an attack vector, which is not the subject of this thesis, and second, the feasibility of an attack path which is what we address as a part of our thesis using the nominated attack analysis methodology. In this thesis, we investigate existing methodologies for modelling attacks and try to nominate one that is most suitable for the automotive industry. This judgement is based on a list of criteria that are collected either through surveying previous related works or through interviewing industrial and academic experts. Once the methodology is nominated, we introduce a method for calculating the feasibility of different possible attack paths using the proposed methodology. Finally, we use some use cases by means of which we demonstrate how our nominated method can be used to model attacks against some assets and how the feasibility of each attack vector can be calculated for the use cases.

Keywords: Automotive, Cybersecurity, Cyberattack, Attack surface, Attack analysis, Risk assessment, Threat analysis, Attack feasibility, Attack potentials.

Acknowledgements

We would like to offer our special thanks to our industrial supervisors Christian Sandberg and Manne Engelke as well as our academic supervisor, Erland Jonsson and examiner, Tomas Olovsson for their unsparing and relentless help and support throughout our thesis project.

We would also like to express our great appreciation to the following people for their assistance with the data collection process:

Urban Thorsson, Volvo Group Trucks Technology
Olayinka Oladele, Volvo Group Trucks Technology
Nasser Nowdehi, Chalmers University of Technology
Adi Karahasanovic, Combitech

Seyed Reza Esmaeili Gothenburg, June, 2019
Afshin Soltani Esterabadi Gothenburg, June, 2019

Contents

List of Figures

List of Tables

List of Acronyms

API	Application Programming Interface
ASIL	Automotive Safety Integrity Level
AUTOSAR	AUTomotive Open System ARchitecture
BSW	Basic Software
CAL	Crypto Abstraction Layer
CAN	Controller Area Network
CANFD	CAN Flexible Data rate
CEL	Common Exposure Library
CIA	Confidentiality, Integrity, Availability
CSM	Crypto Service Manager
DFD	DataFlow Diagram
DoS	Denial of Service
ECU	Electronic Control Unit
EVITA	E-safty Vehicle Intrusion proTected Applications
FTP	File Transfer Protocol
GPS	Global Positioning System
HEAVENS	HEAling Vulnerabilities to ENhance Software Security and Safety
HoliSec	Holistic approach to improve data Security
IL	Impact Level
IoT	Internet of Things
IPA	Information Promotion Agency
ITS	Intelligent Transport Systems
LIN	Local Interconnect Network
MOL	Methods and Objectives Library
MOST	Media-Oriented System Transport

OBD	On-Board Diagnostics
OCTAVE	Operationally Critical Threat, Asset, and Vulnerability Evaluation
OEM	Original Equipment Manufacturer
OWASP	Open Web Application Security Project
PASTA	Process for Attack Simulation and Threat Analysis
PATS	Passive Anti-Theft System
RKES	Remote Keyless Entry/Start
RSL	Road Speed Limit
RTE	Run-Time Environment
SAE	Society of Automotive Engineers
SecOC	Secure On-board Communication
SL	Security Level
SWC	Software Component
TAL	Threat Agent Library
TARA	Threat Analysis and Risk Assessment
TCP	Transmission Control Protocol
TL	Threat Level
TOE	Target Of Evaluation
TPMS	Tire Pressure Monitoring System
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to everything
VLAN	Virtual Local Area Network

1 Introduction

> I think one of the biggest concerns for autonomous vehicles is somebody achieving a fleet-wide hack.
>
> *Elon Musk - Tesla CEO*

Today, our evolution came toward a leap point where connectivity and automation play crucial roles in our daily life. Not only the advent of Internet of Things (IoT), but also the emergence of autonomous products including autonomous vehicles are giving rise to the ever-increasing need of security. The legacy definition of security in automotive was focusing on safety in physical sense, which means the ability to ensure that it is not possible to break into a vehicle or steal it in any way. However, this definition has been changed since in the modern automotive industry, depending on the brand and the type of vehicle, there can exist more than 100 computers (Electronic Control Units, ECUs) and about 100 million lines of code [1], [2]. Therefore, automotive security now also encompasses both computer and network security, which is also known as cybersecurity [3].

1.1 Background

The current automotive industry is leading toward producing vehicles with autonomous driving or an advanced driver-assistance system [4]. In order to fulfill such smart capabilities and also equip the vehicle with more functional features such as Intelligent Transport Systems (ITS), different levels of connectivity needs to be considered in the architecture level. Different communication channels are being developed such as vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) in order to fulfill ITS different goals, including increase of roads safety and traffic

flow [5]. Moreover, vehicles also need to become connected to the Cloud services for remote diagnostics or remote software updates [6], [7]. Figure 1.1 illustrates this connection and communication. Although the evolution trend seems to bring more functionality and intelligence into the automotive sector, however, this can also make modern vehicles subject to various types of intrusion and malicious activities which could be considered as major threats to both the humans factor and the vehicle itself. In order to provide inter-connectivity, connected vehicles rely on wireless and cellular communication interfaces. This exposes them to a wide range of security risks. In 2015, Miller and Valasek [8] performed a research on possible remote attacks on vehicles and they succeeded in breaking into a Jeep Cherokee and consequently taking control over the steering and the braking systems. In the big rig truck's scenario which happened in 2016, attackers succeeded to gain control over the accelerator and the braking system [9]. Lack of safety in a vehicle could lead to major disasters such as loss of life, therefore security breaches are highly intolerable in automotive industry. As a result, there is an ever-increasing public concern toward the cybersecurity of autonomous and connected vehicles [10].

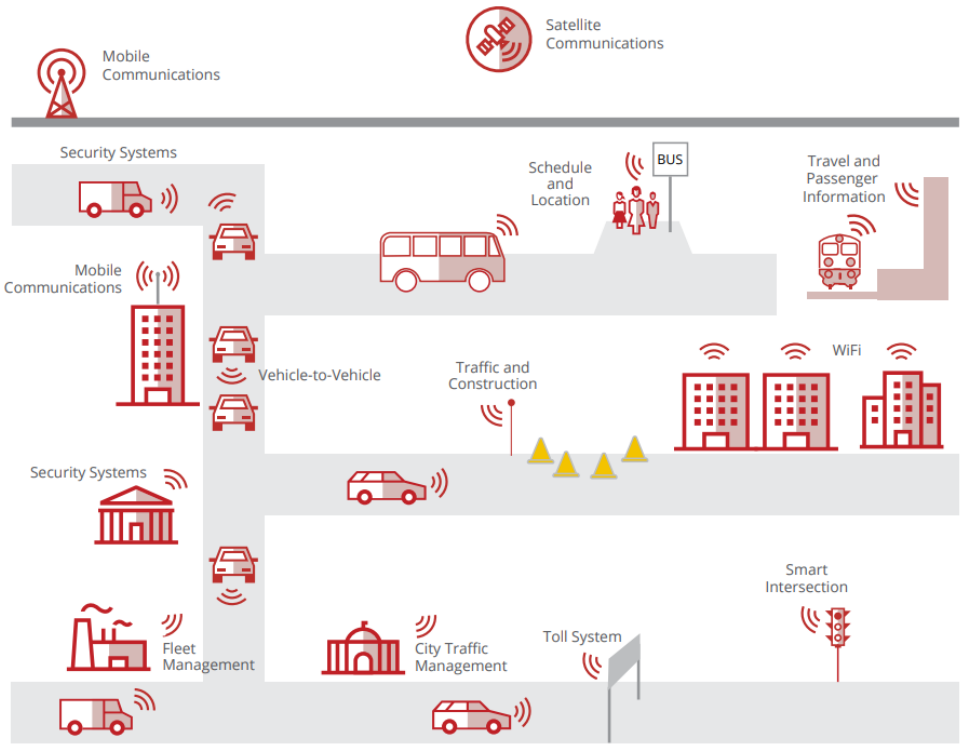

Figure 1.1: Next generation of vehicular communication [10]

In order to tackle cybersecurity concerns in the automotive industry, threat modeling and risk assessment frameworks have been developed [11]. Risk assessment methods usually start with the asset identification step where critical assets are identified. Then by considering standard damage scenarios, different threats to those assets are classified via threat analysis procedures. Thereafter, impact assessment is done by determining the impact levels associated with a compromised asset. As the security design process proceeds, different vulnerabilities may emerge through the vulner-

ability analysis stage. Eventually, by means of a comprehensive attack analysis method, potential attack paths and their associated feasibility are analyzed. The results derived from the risk assessment process are the basis upon which the risk treatment is applied as the closing stage [12].

1.2 Thesis scope and domain background

Nowadays, risk assessment plays a crucial role in the development process. Several threat modeling and risk assessment frameworks have been developed for the automotive domain [13], however an alleged state-of-the-art study [11] performed by the HEAVENS project in 2016 indicated that "security design and architecture has only been addressed to some degree in vehicular systems" and "internal security is more or less absent". The main goal of a risk assessment framework is to categorize possible threat scenarios according to their impact on a stakeholder considering the associated attack paths [13]. The scope of this thesis is defined over the risk assessment process where attack analysis plays a vital role in identifying potential attack paths and their associated feasibility. Various attack analysis approaches are being used in different security frameworks. However, there is no standardized attack analysis methodology for the automotive industry. Therefore, after a comprehensive study of the state-of-the-art attack analysis methodologies and by considering automotive-specific criteria and security expectations in accordance to the latest automotive-specific risk assessment frameworks, which are closely aligned with the safety processes of ISO 26262 [14], we demonstrate the most suitable methodology for the automotive sector. Figure 1.2 demonstrates the risk assessment process and where attack analysis, upon which this thesis is based, stands.

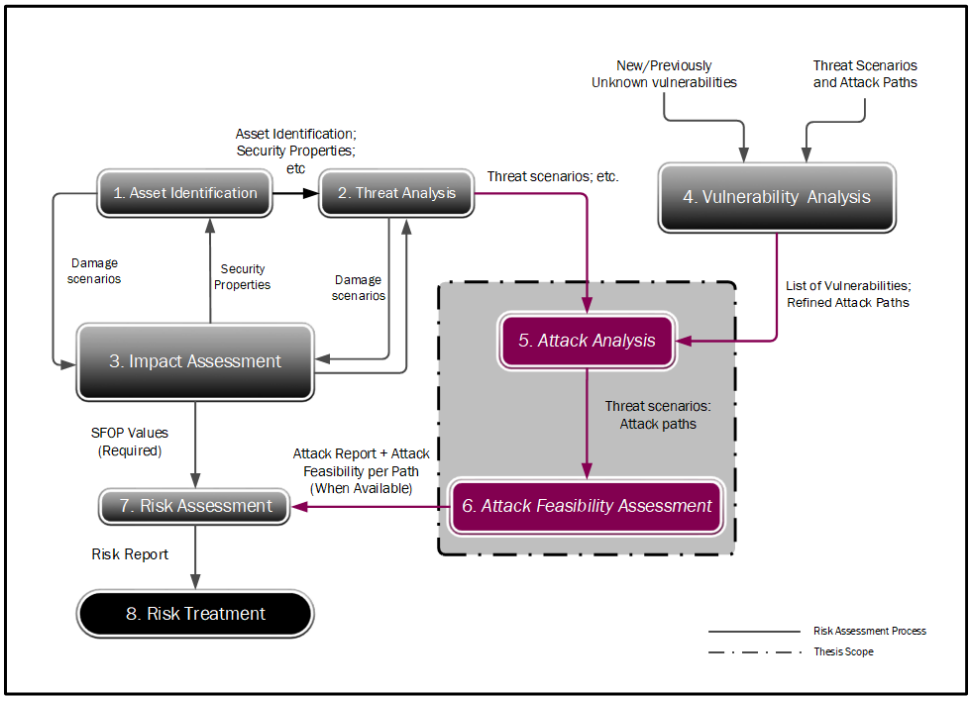

Figure 1.2: Risk assessment process and the thesis scope

1.3 Approach

The objective of attack analysis is to develop and/or update a set of attack paths by which vulnerabilities could be exploited to realize a threat scenario. In order to analyze the security of an information system, plenty of different approaches have been considered by different research articles so far. The most well-known attack modelling techniques used to analyse cyberattacks are Attack Graph [15], Attack Tree [16], Attack Vector [17], Attack Surface [18], Diamond model [19], OWASP's threat model [20] and Cyber-Kill-Chain [21].

To the best of our knowledge, attack trees and graphs have been used so far to create a model to analyze attacks that occur in computer networks and the IT domain. Therefore, in order to discover the best practice among attack analysis methods for the automotive industry, we do further investigations to find the possible advantages of these two approaches compared to other methodologies, if any. In addition, in order to have a list of the most prominent criteria for evaluating our attack analysis methodology, we are aiming both to analyze existing related articles, and to conduct a series of interviews with cybersecurity experts at Volvo Group company. Afterwards, using some use cases, the feasibility of different types of attacks is assessed using mathematical calculations on attack paths.

1.4 Contribution

The thesis focuses on finding the best answer for the following questions:

1- Among existing attack analysis methodologies, which one is best suited for automotive industry?

2- What are the criteria based on which the proposed solution is selected?

Not only do we try to answer the aforementioned questions, but from the industrial perspective, we propose the most suitable method and modelling tool to perform attack analysis in a structured way that is applicable to the automotive industry. Afterwards, we suggest a framework by means of which the feasibility of attack paths can be calculated. The outcome of our work will later be used to find the corresponding risks generated by each attack. If the proposed solution is proved to be the most suitable among other available methodologies (this would be assessed based on the found criteria) and aligned with the needs in the automotive industry, it would be used as the basis for attack analysis methodology for the upcoming ISO standard in the area.

2

In-vehicle Common Architecture

In the most recent architecture of modern vehicles, there exists a considerable number of different ECUs, each responsible for a series of functionalities. To enumerate some of these functionalities, we can mention: engine control, anti-lock braking system, telecommunication, adaptive cruise control and the like. In order to provide each of the aforementioned functionalities, these ECUs need to communicate not only with each other but also with various objects throughout the whole vehicle. Hence, due to different types of messages and different requirements of the ECUs, diverse types of networks must be implemented and then integrated through multiple gateways. Therefore, not only does the design phase of the inner architecture consider the transmission of signals among different controllers but also the accomplishment of the following goals: 1) reducing cable cost, 2) saving package space, and 3) enhancing communication safety [22].

2.1 ECU Classification

As illustrated in Figure 2.1, ECUs can be classified into two categories of safety critical and non-safety critical based on their functionality [23]:

- *Powertrain* ECUs are responsible for controlling safety-critical parts of the vehicle including braking system and engine control. Any failure on these ECUs could lead to varieties of malfunctioning which might cause inability to control the vehicle.

- *Vehicle Safety* ECUs are responsible for providing safety-assisting functionality to the driver, namely collision avoidance system, airbags, anti-lock braking system, adaptive cruise control and tire pressure monitoring. These ECUs could be considered safety-critical since a failure could lead to life threats of the driver, the passengers and other vulnerable road users.

- *Comfort* ECUs provide different driver-assisting functionalities such as parking assistance, thermal management, and electric suspension. Failure of these components might not be considered as a direct threat to the safety of the driver, however the combination of failures still could affect the driver's safety.

- *Infotainment* ECUs are responsible for audio and video support system inside a vehicle. This category is comprised of non-safety functionalities including audio streams, digital broadcasting TV, navigation systems, etc.

- *Telematics* ECUs also provide non-safety functionalities which could be briefly described as telecommunication and informatics integration. These components include those that receive information such as weather and traffic condition from external sources [24].

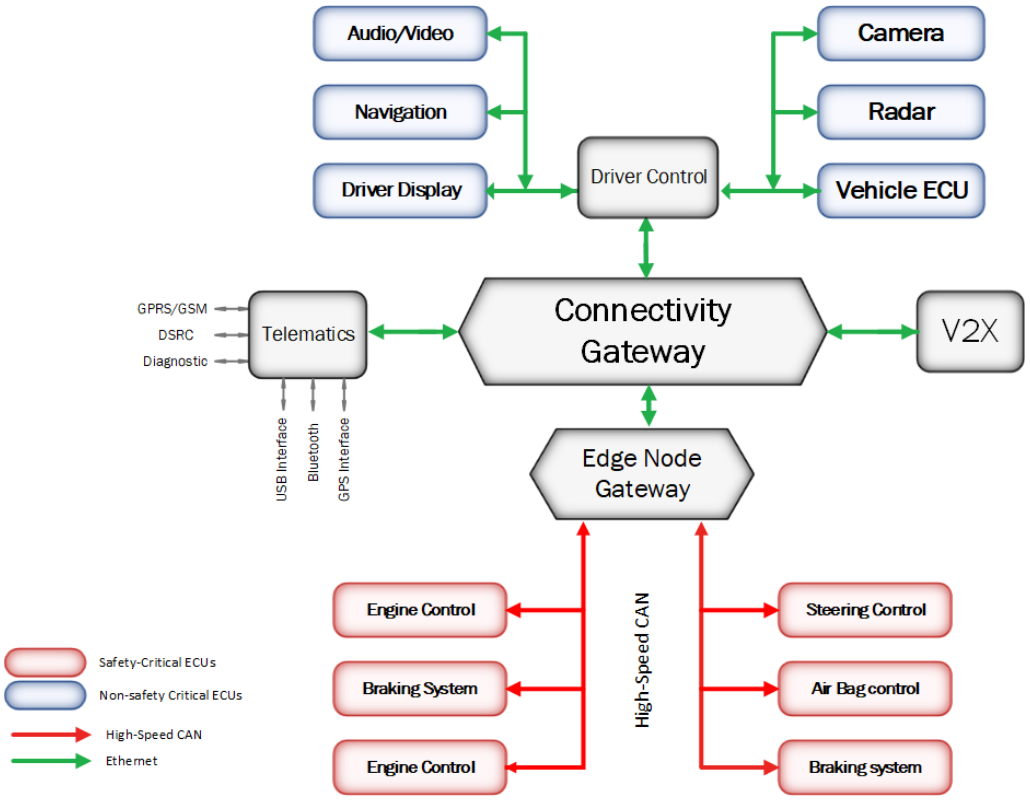

Figure 2.1: Different types of ECUs in in-vehicle common architecture

Many changes happened, through the evolutionary trend, to the modern ECUs compared to legacy ones which were first introduced as a controller for adjustment of fuel/air mixture for combustion process. As mentioned earlier, current vehicles are utilizing multiple ECUs for different functionalities of the vehicle. Many of these ECUs have often been standardized and been utilized by many manufacturers. These standard ECUs are not built by manufacturers anymore but a third-party supplier such as Bosch is producing them. As a part of the standardization process, there is the AUTOSAR project on which we elaborate more in the following section.

2.2 AUTOSAR

AUTomotive Open System ARchitecture (AUTOSAR) project was lunched in 2003 as a collaboration among different companies within the vehicular industry. The main goal of this project was proposing an open and standardized software architecture between manufacturers and suppliers for the sake of reducing the growing complexity rate of softwares [25]. There are nine core companies that are participating in the evolutionary trend of the AUTOSAR including Ford, BMW, General Motors and many others like Volvo as premium partners [26]. Figure 2.2 shows this partnership program.

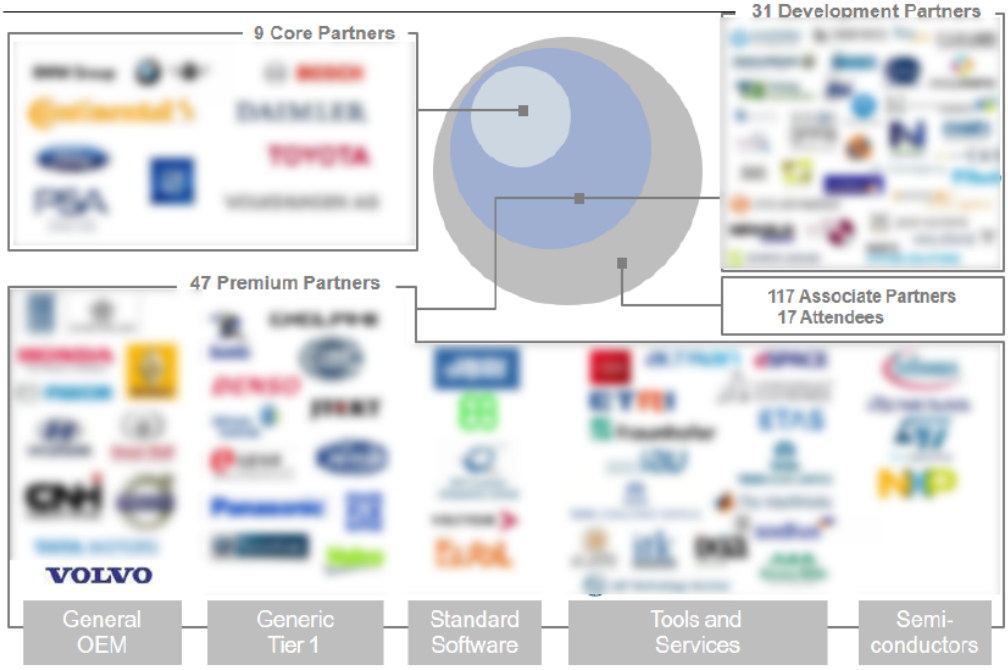

Figure 2.2: Overview of the AUTOSAR partnership program [26]

AUTOSAR defines the software architecture and interfaces together with the design flow and the way the software needs to be mapped to the ECUs during the product development phase. Hence, every company needs to follow the same process in order to be able to implement it on their products. However, the high configurability of this architecture allows it to be customized according to the specific needs of the manufacturer. This allows the involving company to "Cooperate on standards, compete on implementation" which is AUTOSAR official motto [27].

The basic software architecture consists of:

- **Service Layer** - this is the highest level which provides operating system functions including ECU state management, logical and temporal program flow monitoring, communication services, memory services, and diagnostic services. This layer also offers the principal security mechanism of the AUTOSAR architecture, namely SecOC security module and CSM cryptographic module.

- **ECU Abstraction layer** - this layer acts as a medium between different functions of the application and the drivers of the micro-controller abstraction layer. It works as an Application Programming Interface (API) to different micro-controllers including both internal and external. In this way, higher layers do not need to be concerned about the ECU design.

- **Micro-controller Abstraction layer** - this layer provides accessibility to the micro-controller and internal parameter in a way that makes higher layers independent.

- **Complex Drivers layer** - This layer helps to integrate drivers of special purpose devices which are not defined in the AUTOSAR standard. This layer also provides direct accessibility to the micro-controller layer.

A more detailed map of the AUTOSAR software architecture can be seen in Figure 2.3.

2.2.1 Security Features

For the sake of security, AUTOSAR provides significant security mechanisms in software level, which can be used by different modules and Software Components (SWCs). This standard also provides solution for securing on-board communication while the rest of the security consideration is outsourced to the original equipment manufacturers (OEMs). OEMs are usually responsible for utilizing different cryptographic protocols in order to be implemented on their vehicles [28].

Three principal security mechanisms provided by AUTOSAR are:

- **CSM** - Crypto Service Manager, which according to layered structure of AUTOSAR shown in Figure 2.3, resides in the service layer of Basic Software (BSW) and offers services to higher layers. It actually provides different applications with a unified service for accessing different cryptographic primitives. For instance, one application may utilize SHA256 while the other one requests for MD5. It is worth noting that CSM service is only accessible locally inside an ECU [28].

- **CAL** - Crypto Abstraction Layer also provides very similar functionality as CSM does. This static library, provides direct cryptographic functionality through bypassing the Run-Time Environment (RTE). Different software modules such as BSW/SWC can directly call different functions provided by CAL. It is worth noting that this library is not related to any layers of AUTOSAR architecture.

- **SecOC** - Secure On-board Communication module offers an authentication mechanism for secure data communication for all ECUs that are handling critical data. The security mechanism provided by this module is not only

relatively light-weight but also highly compatible with all existing communication protocols including both legacy and modern ones [26].

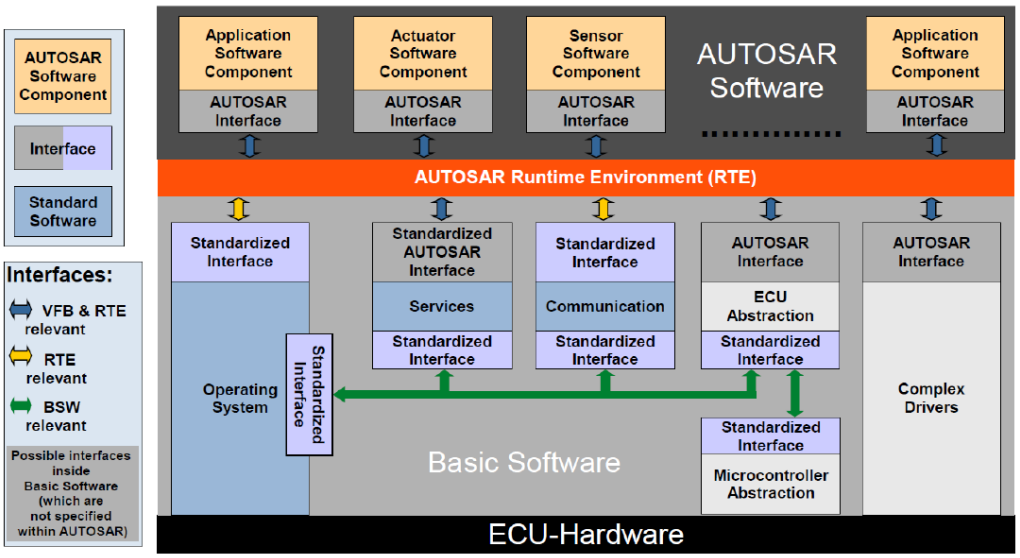

Figure 2.3: AUTOSAR software architecture - components and interfaces [26]

Nowdays, AUTOSAR standard is widely used by most of the companies within the vehicular industry. Accordingly, the security of softwares that are being implemented on vehicles are highly dependent on the security scheme and route map provided by this standard. Therefore, in order to provide the best practice scheme for securing modern vehicles in the software level, there is a constant need for contribution of cybersecurity experts from these companies. This collaboration could help to improve the AUTOSAR framework not only according to the most recent needs of automotive industry but also, to some extent conforming to the most recent proprietary security goals of each company.

2.3 Communication Bus System

There are five different network types for communication in vehicular systems: 1) Local Interconnect Network (LIN) which is dedicated to functions with the lowest data-rate such as mirror control, door locks, and climate control; 2) Media-Oriented System Transport (MOST) that is designed for high-speed bandwidth with 24.8 Mbps information rates which was used for Global Positioning System (GPS) and in general the navigation unit, media showcases and entertainment system [29]; 3) Controller Area Network (CAN), which is our main focus in this thesis, covers the highest proportion of the communication channels inside a vehicle and is used for communication between controllers, actuators and sensors. CAN supports 1Mb/s data rates. However, CAN Flexible Data rate (CANFD), which is a high speed CAN, supports a higher data rate and bandwidth [30]; 4) FlexRay is dedicated to deterministic communication and safety-critical applications including stability

control, brake-by-wire, and steer-by-wire [31]. FlexRay is designed to transfer event-triggered and time-triggered information in the same cycle. The high production cost of FlexRay makes it less popular among vehicular corporations [30]; and 5) Ethernet is considered as the future of communication inside vehicles. Due to its high-speed transmission limits of up to 10 Gbit/s, it is considered as the best alternative to other communication channels such as CAN which is prone to bottlenecks. Moreover, Ethernet provides high reliability and adaptability making it a good alternative for the backbone channel in future in-vehicle network architecture [32].

2.4 Communication Specifications

Today's vehicles are enriched with wide varieties of functionality. Not only do this give a rise to the high rate of internal communication between different ECUs, sensors, and actuators but also increases the need to communicate with external points including back office, nomadic devices, infrastructures, other vehicles, and Cloud servers. Therefore, several communication channels need to be considered throughout the architecture design phase in order to fulfill different objectives.

2.4.1 Internal Communication

As it is shown in Figure 2.2, modern vehicles use different communication channels to exchange data between various components inside the car such as different sensors, ECUs, and actuators. For this sake, different communication protocols such as LIN, CAN, FlexRay, MOST, and Ethernet are used in the physical layer. A common process among all corporations is to move from low bandwidth protocols like CAN toward technologies that provide a relatively higher bandwidth; Protocols such as FlexRay, CANFD and Ethernet which is able to provide up to 1Gbit/s of bandwidth. Generally, communications over internal buses take place through broadcast messages. Hence, every node situated on the same subnet as the sender, would be able to read broadcast messages. Most of the events inside a vehicle are considered as real-time events, thus there are many time-critical messages transferred through communication buses where each of them guarantee criticality by means of different policies. For instance, CAN as a network with high portion of time-critical communications provides this guarantee by giving different priority IDs to the messages in a way that high priority messages can override low priority ones. Although CAN is providing so many good features which makes it the most suitable communication network for in-vehicle communications, it lacks providing a security mechanism. This results in different applications having to implement their own security mechanism for the sake of message authentication, otherwise any intruder can falsify a message and reach a malicious purpose by just broadcasting the message on the CAN bus [33].

There are many services running in modern vehicles requiring higher bandwidth as well as certain levels of isolation and security. This gives rise to more demands toward a high capacity channel with more networking abilities. As the Ethernet provides higher bandwidth for communication, it is possible to partition an Ethernet

LAN into multiple Virtual Local Area Networks (VLAN) to add different properties to the channel such as prioritization and security. This makes Ethernet the best option for the implementation of a high speed channel inside a vehicle. Table 2.1 shows the VLANs specifications provided by HoliSec architecture [33].

Table 2.1: VLANs in the HoliSec reference architecture

VLAN ID	Description
VLAN 1	High priority communication between applications in connectivity ECU, V2X and Driver Display
VLAN 2	Low priority communication between applications in connectivity ECU and Driver Display
VLAN 3	Medium priority communication

2.4.2 External Communication

By equipping modern vehicles with a broad range of functionalities, there is an ever increasing trend toward external communication. Initially, the first external channel has been established between the vehicle and the back office in order to provide services such as collecting diagnostic trouble reports, accident reports, and multimedia connectivity [33]. Gradually, more and more services were provided by the OEMs and the data exchange rate between the vehicle and the back office increased accordingly. To enumerate some of such services, we can mention navigation system, fleet management system, diagnostics, remote vehicular functions, media streaming via nomadic devices, etc. Moreover, by the advent of V2V and V2I networks, soon there will be even more need for reliable and high-speed external communication channels. The potential communication channels for the aforementioned networks will most likely use 5G and wireless 802.11p. Nevertheless, there are still limitations such as cell coverage and signal strength associated with these technologies which need to be addressed for the sake of higher reliability and availability [33].

3

Taxonomy of Cybersecurity Concepts

Prior to our scientific approach towards the given problem, a complete understanding of preliminary cybersecurity concepts and definitions needs to be acquired. Therefore, in the following section we demonstrate some of the most common cybersecurity terms, concepts, and standards related to our research scope so that the readers could reach a better understanding and adaptation of the content when reading the following chapters.

3.1 Preliminary Security Definitions

In this subsection we take a deeper look into some of the initial security definitions related to cybersecurity of vehicular section.

- **CIA:** As shown in Figure 3.1, Confidentiality, Integrity and Availability form one of the most comprehensive triangular security models which is being used widely within the domain of information systems. These three are considered as the key factors needed to be guaranteed in order to assure the security of a system.
  - **Confidentiality** means that within a certain information network, only an authorized user should be able to access certain types of data. As this face is one of the most fundamental aspects of a secure system, it is commonly targeted by attackers. Cryptographic solutions are being developed as a main countermeasure toward assuring this aspect of information systems.

  - **Integrity** means that in an information network, the data that is being sent by the sender should be received accurately and unchanged. In other words, the data needs to be protected from any unauthorized access and modification while being transferred across the network.

– **Availability** means to ensure that information-critical resources are accessible to authorized users on demand. This face is taken into account due to the presence of different types of attacks that target an information resource and try to make it unavailable.

Figure 3.1: The CIA triad model

- **Asset:** An asset (system resource) within an information system could be any type of data, service, and equipment including both software and hardware [34]. This factor could be considered as a point of focus around which different security definitions are defined and developed.

- **Vulnerability:** Any flaw or weakness in the system's design and operation that can be considered as a potential leakage towards the violation of the CIA triad [34].

- **Threat:** A possible danger to the system that can exploit different vulnerabilities of a system and consequently violate the CIA triad.

- **Attack:** Any malicious activity caused by an unauthorized person threatening the CIA triad through exploiting the potential vulnerabilities of a system. Attacks could be classified into two main classes: 1) Active: any attempt to make alteration to system resources or intervene their operation. 2) Passive: any type of exploitation of system's information resources without affecting its functionality.

- **Attacker:** Any entity that performs the act of attack or could be considered as a threat to a system in anyway.

- **Risk:** A risk is a probability and impact of a potential threat or attack within a system [12]. We will have a deeper look into this concept once demonstrating different security and risk assessment frameworks in section 4.

- **Countermeasure:** Any activity that strengthens a system against different vulnerabilities, threats, or attacks. This is usually done by a complete analysis of the system and reporting accordingly so that the consequent actions can be taken in order to mitigate the risk.

- **TOE:** Target Of Evaluation is the product or the system that is the subject of evaluation.

3.2 Automotive Cybersecurity Guidelines

As we pointed out in previous sections, because of a broad range of functionalities and services provided by modern vehicles, there is an ever increasing trend towards using more ECUs and connectivity media in order to fulfill those expected functionalities. The automotive industry has been implementing ISO 26262 [14] since 2011 to address functional safety requirements. However, because modern vehicles are also becoming prone to computer and network security risks, complementary guidelines have been developed to address cybersecurity. It is the matter of concern to go through these guidelines before demonstrating our approach since the provided approach is defined based on the cybersecurity risk assessment process. Time-wise, Esafety Vehicle Intrusion Protected Applications (EVITA) [35] was the first framework co-funded by the European Commission trying to complement the safety process with cybersecurity concerns. The primary design goal of EVITA was to provide a secure architecture for vehicular networks to prevent them from being tampered or compromised by an external source. Two years after the publication of EVITA, Information Promotion Agency (IPA), which is located in Japan, proposed another document with similar cybersecurity objectives but this time for Asian manufacturers. This trend continued until 2016, where two significant frameworks namely HEAVENS [12] and SAE-J3061 [3] were published by Volvo and Society of American Engineers (SAE) respectively. These two guidelines were published to formulate a set of recommendations on how cyber-threats need to be addressed by having the functional safety and the secure development life-cycle in mind. However, SAE is considered to be more comprehensive since it considers a more general case for the vehicle industry compared to HEAVENS that can be considered as Volvo's proprietary cybersecurity guideline. In this section after a brief demonstration of the secure development life-cycle in vehicular industry, we will shortly discuss the most significant security recommendations and guidelines in vehicular systems.

3.2.1 Automotive Secure Development Life Cycle

Automotive systems are considered to be safety-critical systems where different physical or cyber-physical threats to the system could lead to life threatening hazards. Hence, most of the companies in automotive domain are following the common functional safety standard ISO-26262 [14] during the product development life-cycle. This standard has been extracted from the more general standard IEC 61508. According to ISO-26262, every product needs to be developed according to a traditional

V-model that is shown in Figure 3.2. This model is comprised of three distinct phases where each consists of different sub-phases:

- **Concept phase:** During this phase by developing initial system design, the safety life-cycle process is lunched. The most significant step in this phase is hazard identification and risk assessment where after careful assessment of potential safety risks, corresponding safety goals and Automotive Safety Integrity Levels (ASILs) are defined for every object in the system. Once the safety goals are defined, the concept phase ends [36].

- **Product development phase:** This phase is comprised of three nested v-models including i) product development at system level, ii) product development at hardware level, and iii) product development at software level. This phase ends once the product is released [37].

- **Production and operation phase:** The final stage handles validation and verification of the operation and the safety of the product.

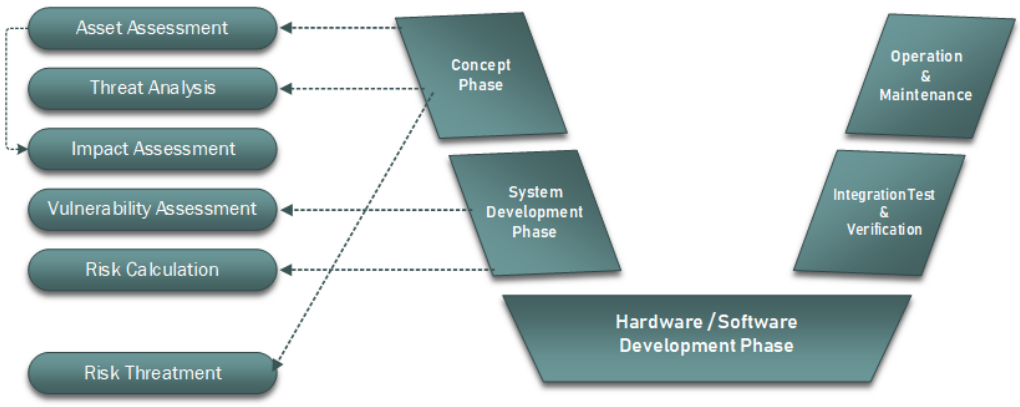

Figure 3.2: Mapping risk management process onto the V-model of product development

In order to ensure the security and safety of a vehicle, risk assessment is a crucial stage during the development process. There exist many risk assessment frameworks where few of them are developed according to the needs of the automotive industry. HEAVENS [12], which was launched in collaboration with Volvo Trucks in 2013, provides a risk assessment framework closely aligned with ISO-26262 safety process. This project was a concrete step towards cybersecurity standardization for automotive systems and was referred to as "Cybersecurity Guidebook for Cyber-Physical Vehicle Systems" in the first international security guideline in automotive, J3061 [3].

3.2.2 EVITA

E-safety Vehicle Intrusion proTected Applications (EVITA), was developed in collaboration with European Commission from 2008 to 2011. The aim of this project was to formulate the fundamental security requirements of applications based on V2X and V2V communication. The significance of this project was not only to provide an automotive-specific cybersecurity risk analysis framework, but also to secure the in-vehicle architecture as well as the communication protocols. This project provided security in the sense that, safety-critical components of a vehicle need to be identified in the first step. Furthermore, security strategies need to be applied during the whole product development phase in order to protect different components against tampering and to protect sensitive data from any external interference. Moreover, the CIA aspects of a system need to be mostly fulfilled via cryptographic solutions. The risk analysis stage in EVITA suggests a model to assess both the risk associated with an attack and the severity of its impact on the stakeholders together with the likelihood of the attack to be successful [35].

In order to provide security during the development phase, EVITA suggests four different security requirements which need to be satisfied from the highest point of view, including:

- **Operational:** maintaining the intended level of operational performance for all vehicles and ITS systems.
- **Safety:** ensuring the functional safety of all persons who are affected by the vehicle operation, namely road users and the people inside the vehicle.
- **Privacy:** preventing any sensitive data of the driver, manufacturer, and the supplier from being disclosed to an untrusted third party.
- **Financial:** handles prevention of fraudulent commercial transactions and theft of vehicles.

Different threats to the system then could be identified and classified according to the aforementioned security goals. EVITA provides a grading system according to the classification and the severity of the impacts as can be seen in Table 3.1. For the sake of risk analysis, EVITA needs to calculate a quantitative value corresponding to the probability of a successful attack which is a function of the amount of time that the attack needs to be performed together with the level of experience and knowledge that the attacker needs for a successful attack. Finally, the associated risk is derived from the combination of this probability and the severity level [3].

Table 3.1: Severity classification scheme for security threats in EVITA

Severity class	Aspects of security threats			
	Safety	**Privacy**	**Financial**	**Operational**
0	No injuries.	No unauthorized access to data.	No financial loss.	No impact on operational performance.
1	Light or moderate injuries.	Anonymous data only	Low-level loss.	Impact not discernible to driver.
2	Severe injuries (survival probable). Light/moderate injuries for multiple vehicles.	Identification of vehicle or driver. Anonymous data for multiple vehicles.	Moderate loss. Low losses for multiple vehicles.	Driver aware of performance degradation. Indiscernible impacts for multiple vehicles.
3	Life threatening or fatal injuries. (survival uncertain) Severe injuries for multiple vehicles.	Driver or vehicle tracking. Identification of driver or vehicle for multiple vehicles.	Heavy loss. Moderate losses for multiple vehicles.	Significant impact on performance. Noticeable impact for multiple vehicles.
4	Life threatening or fatal injuries for multiple vehicles.	Driver or vehicle tracking for multiple vehicles.	Heavy losses for multiple vehicles.	Significant impact for multiple vehicles.

3.2.3 HEAVENS

HEAling Vulnerabilities to ENhance Software Security and Safety (HEAVENS) project was launched in 2013 and was delivered in 2016. This project, which was developed by a team of security experts from both academia and industry, pursue some important goals such as [12]:

- Identify needs and requirements of security in automotive industry.
- Study and identify state-of-the-art work of security in automotive industry.
- Identify potential threats, threat agents and vulnerabilities to construct security models.
- Map security issues from related domains (e.g. software engineering, computer networks, etc) into automotive domain.
- Define methodologies and identify tool support for evaluating software security.
- Investigate the interplay of safety and security according to the common architecture, considering ISO 26262, AUTOSAR [26] and other relevant standards.
- Demonstrate proof of concept.

HEAVENS security framework policy is comprised of two important factors:

- **Security Attributes:** These attributes are inherited from STRIDE model that classifies security concerns into six different categories. We will elaborate on them in the upcoming sections of this chapter.

- **Security Objectives:** Security objectives are adapted and categorised in four groups of operational, safety, privacy and financial, as it was done in EVITA.

As illustrated in Figure 3.3, HEAVENS security model consists of three distinct phases [12]:

- **Threat analysis:** Functional use cases are provided as input into the threat analysis stage which consequently produces two different types of output: i) mapping between threats and assets that is done per asset in the context of use case, ii) classification of threats based on the provided security attributes, in order to find out about the security attributes that are being violated by a particular threat.

- **Risk Assessment:** After threat analysis, the next step is to rank the threats. The results derived from threat analysis stage together with Threat Level (TL) and Impact Level (IL) are inputs to the risk assessment phase. At the end of this stage, the Security Level (SL) of each threat and its associated asset are identified.

- **Security Requirements:** Ultimately, the mapping between threat and asset together with the associated security level are used to formulate the security requirements of the asset and the TOE. Hence, the security requirements are a function of asset, threat, security attribute, and security level.

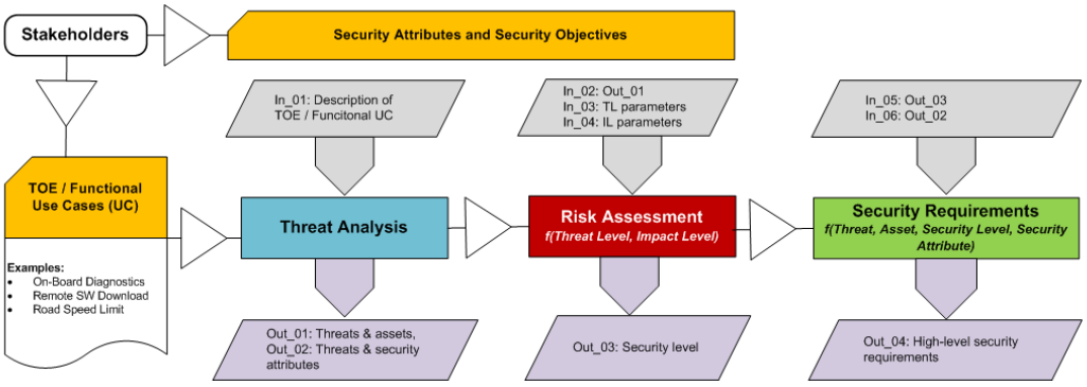

Figure 3.3: Workflow of HEAVENS security model [12]

As shown in Figure 3.4, risk assessment process, as the main part of HEAVENS, takes two numerical grades, namely TL and IL into account in order to generate the security level factor.

Security Level (SL)	Impact Level (IL)					
Threat Level (TL)		0	1	2	3	4
	0	None	None	None	None	Low
	1	None	Low	Low	Low	Medium
	2	None	Low	Medium	Medium	High
	3	None	Low	Medium	High	High
	4	Low	Medium	High	High	Critical

Figure 3.4: Security level based on threat level and impact level [12]

The threat level factor represents the likelihood of the threat derived from attacker's capabilities, required equipment for performing the attack, and the windows of opportunity. The latter is a factor which derived from the time and the type of access (e.g. physical and/or remote) that attacker needs for a successful attack. The results, which affect the impact level, are evaluated according to EVITA's security objectives, as depicted in Figure 3.5 (it is worth mentioning that threat analysis process in HEAVENS framework uses Microsoft STRIDE model in combination with EVITA's threat classification). The impact level on the other hand is defined over the consequences that an attack posses to the users and stakeholders. For instance, an attack that tampers the normal operation of the break ECU, can cause a safety-critical situation which may consequently have a sever impact on the system. The security level factor represents the severity level of the vulnerability endangering the TOE.

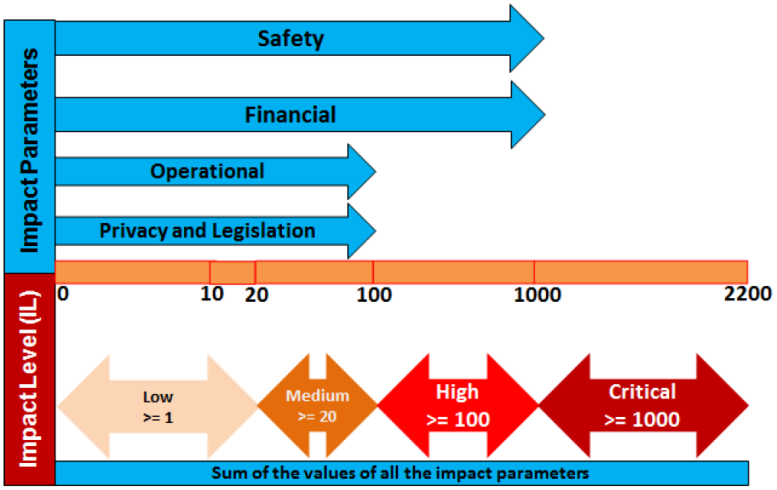

Figure 3.5: Impact parameters and impact level in HEAVENS security model [12]

3.2.4 SAE - J3061

In 2016, the SAE-J3061 standard [3] was published as the most significant cybersecurity guideline for cyber-physical vehicle systems. This standard was developed on top of the well-known functional safety standard ISO-26262 in order to integrate the cybersecurity consideration with functional safety objectives. This standard utilizes Threat Analysis and Risk Assessment (TARA) as a threat analysis framework for secure development process. It is worth mentioning that SAE-J3061 offers "Attack Tree" model for attack analysis stage in the risk assessment process. This standard coordinates a very good interaction between security and safety process. As Figure 3.6 shows, it can be inferred that SAE-J3061 is an information security standard tailored for the automotive safety process [38].

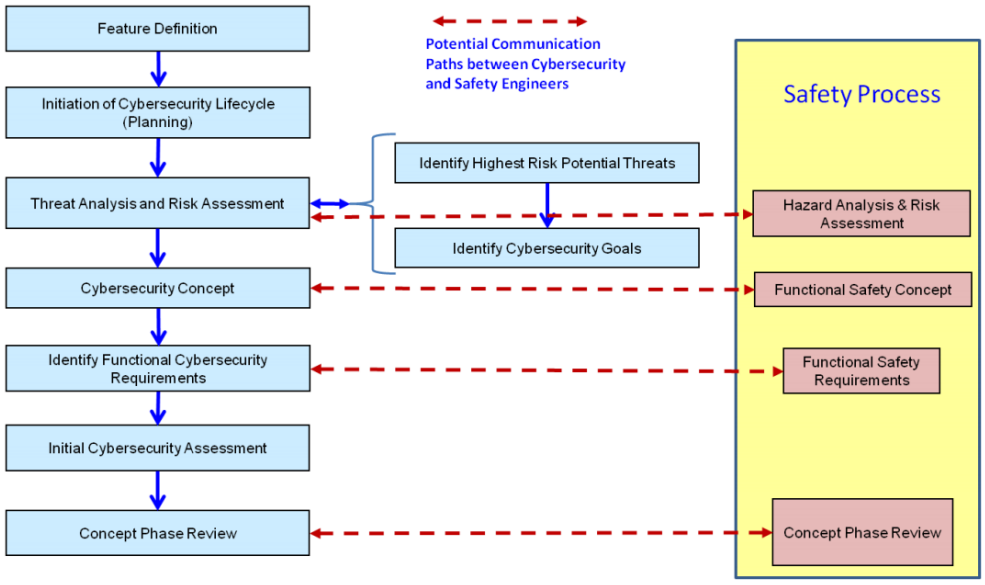

Figure 3.6: Communication paths of the concept phase activities [38]

3.3 Attack Surfaces in Modern Vehicles

The attack surface of a vehicle is the sum of different vulnerable points (e.g. attack vectors) that can be leveraged by an attacker in order to perform malicious activities. Therefore, one of the basic security considerations is to keep the attack surface as small as possible [39]. After explaining the common architecture of modern vehicles, in this section we are going to elaborate on possible attack surfaces according to the modern architecture. In 2011, Chekoway et al. [2] provided the first taxonomy on attack surfaces. Thereafter, in 2014, Miller and Valasek [40] published an extensive list of possible attack surfaces by performing a study on 21 different car models . At the same time, in another paper published by Zhang et al. [41] different attack surfaces were studied while focusing on malwares. In this section, we elaborate more on the discovered attack surfaces according to the aforementioned articles.

3.3.1 Physical Access

All manufacturers equip their vehicles with several physical ports in order to provide either direct or indirect access to the vehicle's internal network. Among these existing interfaces, the followings are considered as potential attack surfaces:

- **OBD-II Port:** On-Board Diagnostics port, which is installed in almost all modern vehicles, was designed for diagnostic purposes. It provides either direct access to the CAN bus, or indirect access via central gateway. In the past, special purpose devices were used for diagnosis through the OBD-II port, but nowadays inexpensive dongles can be connected to this port and most of garage servicemen can use regular computers or smart phones to connect to these dongles which makes this port prone to attacks.

- **Entertainment/removable media ports:** Today, there is a rising trend in equipping vehicles with different entertainment systems including CD-players, USB ports, IPod connectors, etc. Since these interfaces are connected to the vehicle's internal network to support, for instance, hands-free features, this makes them a potential attack surface.

3.3.2 Short Range Wireless Access

Modern vehicles no longer provide entertainment services via physical ports, but instead they are equipped with different short range (e.g. 3 to 300 meters) media such as Bluetooth or Wireless. These are used not only for connecting smart phones but also for the remote key entry and tire pressure monitoring.

- **Passive Anti-Theft System (PATS):** This feature is placed in most of new vehicles. It provides an anti-theft system by integrating a sensor in the steering column to the ignition key. The key is triggered by an RF signal generated by one of the ECUs in the vehicle so that the vehicle checks whether the key is close enough to the vehicle and that is has not been started by an attacker. The potential attack surface related to this system is prone to Denial of Service (DoS) attack in which the attacker tries to bombard the vehicle with intervening signals to prevent it from establishing a connection to the key.

- **Tire Pressure Monitoring System (TPMS):** This system constantly checks the tires pressure and reports the status to the corresponding ECU. The attack surface in this case could be a false report to the ECU. However, Ishtiaq Roufa et al. [42] have shown that it is also possible to cause a DoS attack on the associated ECU.

- **Remote Keyless Entry/Start (RKES):** Nowadays, most of the modern vehicles support remote key-less entry or even remote start system. This is done by establishing a sort of authenticated and encrypted channel between the key and the corresponding ECU. Here the attack surface is relatively small since it is only prone to DoS in the sense that the attacker may prevent the

vehicle from recognizing the key or being started remotely.

- **Bluetooth:** Almost all of the modern vehicles are equipped with Bluetooth so that external smart devices can become synced with the vehicle. Although manual pairing protects Bluetooth protocol from being tampered, since it still provides a link to the in-vehicle network, this can be considered as a large attack surface.

- **WiFi:** WiFi connectivity has recently been available in cars to serve as a hotspot for Internet usage of smart devices. Since this is done by bridging the vehicle's Internet connection to the smart devices, therefore it can also be considered as an attack surface.

- **Emerging short range channels:** By the advent of ITS and V2V/V2X, sooner or later vehicles will communicate with each other and the infrastructure. This is possible via special wireless channels known as 802.11p. The short range channels are another attack surface which need to be considered as soon as this technology is widely used.

3.3.3 Long Range Wireless Access

Today's vehicles are also equipped with long range wireless systems in order to be able to communicate over distances further than 1 km. Within this range, we can categorize communication channels into broadcast (e.g. GPS, Radio) and addressable (e.g. 4G Internet connection) channels.

- **Radio Data System:** Today, radio channels are capable of receiving both audio and data signals. Since there is no sign of a data parser in between, these receiver systems can be prone to code execution. However, because of low probability of such attacks, this attack surface is considered relatively small.

- **Global Positioning System:** Nowadays, GPS is widely used in many smart devices and vehicles are not an exception. It is utilized for navigation and internal automation purposes. Moreover, GPS information is also useful for reporting traffic jam and road condition in the context of connected vehicles. However, it has been reported that such positioning devices are prone to spoofing attacks [43]. This attack could lead to diverting the vehicle or making it out of operation.

- **Telematics/Cellular:** Modern vehicles are also equipped with telematic units and cellular channels in order to be able to receive information such as weather condition or traffic status. Since the telematic unit acts as a gateway that connects the vehicle to the Internet, it could be considered as a relatively large attack surface. Moreover, because of telematic's higher bandwidth, they use media-Bus for internal communication while still connected to CAN-Bus

via bridging the ECU and the gateway. In a study done by Checkoway et al. [2] it has been shown that it is possible to cause malicious activities such as killing the engine or activating the windscreen wipers through the telematic gateway.

- **Internet/Apps:** IoS and Android are offering different applications such as navigation for use inside the vehicles. This trend gives rise to a relatively wide attack surface ranging from Malwares to browsers' vulnerabilities.

3.3.4 Sensors

For a vehicle to be able to communicate with its environment and more importantly to function correctly, it must be equipped with various types of sensors. According to [44], these sensors can be tricked to generate false data. However, since these devices are outside the scope of the digital world, we consider them outside the scope of this thesis.

4

Attack Analysis Methodologies

While a framework is more general in its nature, a model is a simplified developed and tested framework. A model tries to focus on the interesting parts and ignore all other unimportant and unnecessary details. Therefore, a threat model highlights the security details with respect to a particular type of system and the assets under consideration by addressing both treat's capabilities and its intent. Fundamentally, threat modelling could be considered as a structured attempt of identifying cyber threats based on their objectives and related vulnerabilities and consequently providing mitigation techniques addressing identified threats. There are wide variety of threat and attack modelling techniques that could be classified into three general categories [45], namely Attacker-centric, System-centric, and Asset-centric based on their behavior and identification strategy. Figure 4.1 shows this classification.

- **Attacker-centric:** This approach focuses on attacker's capabilities, motivations, and goals and the way they can be achieved. It has been considered by some of the well-known models such as Intel's TARA (Threat Analysis and Risk Assessment) and Cyber Kill Chain.

- **Asset-centric:** Asset-centric approach focuses on the target information or resources of a system that an attacker tries to compromise. This approach is more common than the attacker-centric method. However, models that use this approach are considered as both time and resource consuming since they need more time and more resources to model different threats against the target system. The most well-known asset-centric models are PASTA and OCTAVE.

- **System-centric:** This approach, also known as 'software-centric'or 'design-centric', focuses on a software being developed or a system being built. This model starts from the design phase of a software or system and investigates different possible threats against each component of the system through the whole development process. System-centric approach is commonly used in different information systems and has become a legitimate standard in the scope of information systems. Two of the most well-known system-centric models are STRIDE, which has been developed by Microsoft, and DREAD.

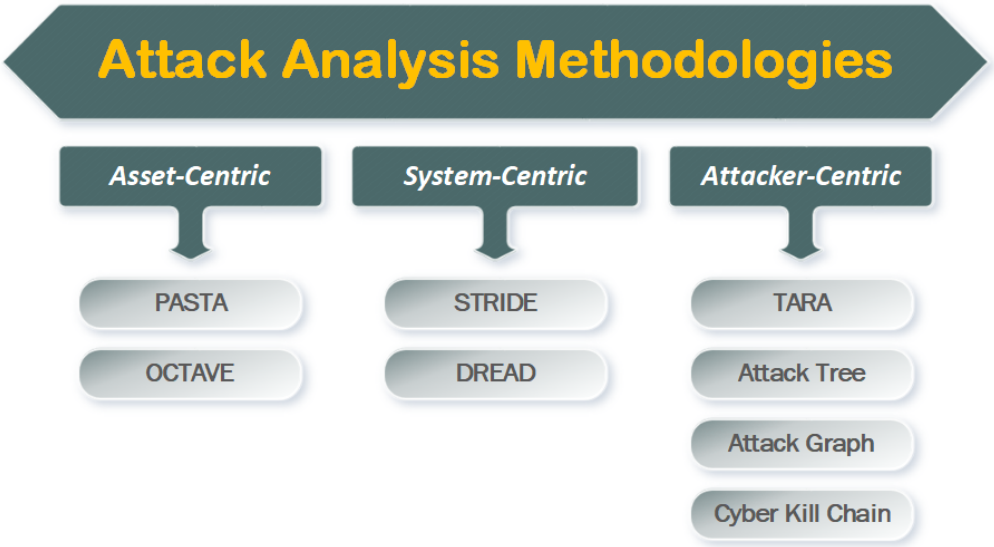

Figure 4.1: Attack analysis methodologies classification

Not all of these techniques are applicable to the needs of automotive industry. Hence in this section, we elaborate more on those recently recommended and utilized by the cybersecurity guidelines described in Section 3.2.

4.1 TARA

Threat Analysis and Risk Assessment (TARA) is a predictive attacker-centric threat modelling developed in order to assess, prioritize, and mitigate cybersecurity risks. This model has been designed in a way to provide easy understanding of the whole risk assessment process to different levels of decision makers without any prior knowledge of cybersecurity while being comprehensive enough to be effective [13]. Because of its comprehensibility and ease of use, this model is widely being used in many industries including the automotive section [12].

One of the main goals of TARA methodology is to reduce the cost of risk assessment by limiting the area of concern to the ones that are most critical and vulnerable within a system. This way the result becomes maximized with the minimum cost.

In order to determine the most exposed area within a system while having known vulnerabilities and corresponding mitigation techniques in mind. TARA needs to identify the most hazardous attackers and their intended goals as well as different techniques they use to reach their desired results.

The Defense-in-Depth strategy [46], developed by Intel IT, is an attempt to optimize the security process by interlocking four different phases:

- **Prediction:** An attempt to anticipate all possible and existing threats.
- **Prevention:** An attempt to prevent threats from happening by utilizing different limitations.
- **Detection:** An attempt to identify actual threats by analyzing the system.
- **Response:** An attempt to design mitigation techniques against actual and possible threats.

TARA methodology can be mapped into the prediction phase. The main objective of this method is to implement an optimal mitigation strategy by identifying the most probable attack paths. Unlike other methodologies providing more general vulnerability treatment, TARA as mentioned before, offers a security strategy by focusing on areas having the highest overall risk [47]. Figure 4.2 illustrates the process of narrowing down the field of attacks. In this model, Intel provides a specific definition of attackers or the so-called threat agents as follows:

"Threat agents are attackers who represent a security risk of loss, and they are classified by characteristics including skills, capabilities, resources, intent, and access."

In order to make a predictive conclusion, TARA relies on three different components:

- **Threat Agent Library (TAL):** This factor defines eight different threat agent attributes whose combination generates 22 unique threat agent archetypes such as disgruntled employee or organized crime [47].
- **Common Exposure Library (CEL):** This factor maintains known security vulnerabilities and exposure of the system under consideration.
- **Methods and Objectives Library (MOL):** This factor covers the list of known threat agent objectives including their goals and the most likely strategies they exploit in order to reach their goals [47].

In order to identify the likelihood of possible attacks, by considering many factors such as typical methods, preferred vulnerabilities, objectives, and resources, MOL factor is multiplied by TAL. In this way, an estimation of consequences is derived accordingly.

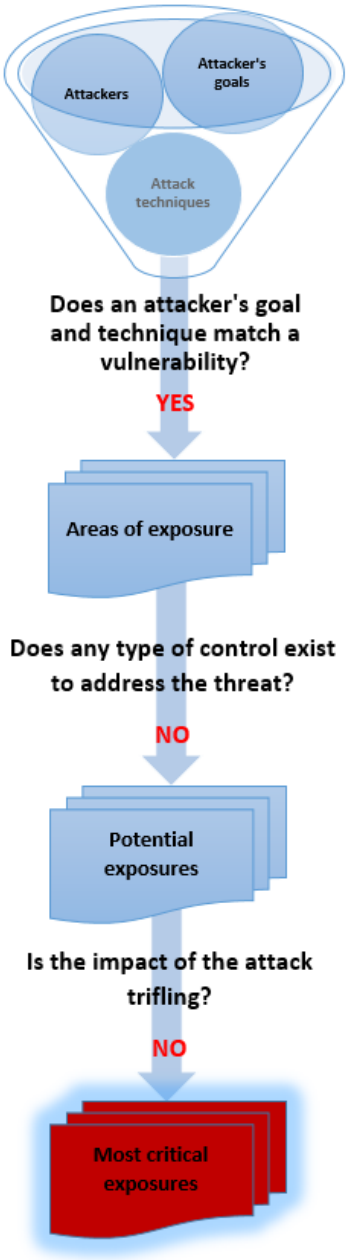

Figure 4.2: Constriction of the field of attacks

By bringing CEL factor into the game, vulnerabilities with sufficient controls and low risks are removed from the process, and the remaining attack vectors are considered as the area with the highest risk. Figure 4.3 shows TARA threat modelling process where a complete filtering of threats happens through six different stages. Through these stages, areas with lower risks are removed from the security experts spotlight while areas with the highest exposure remain.

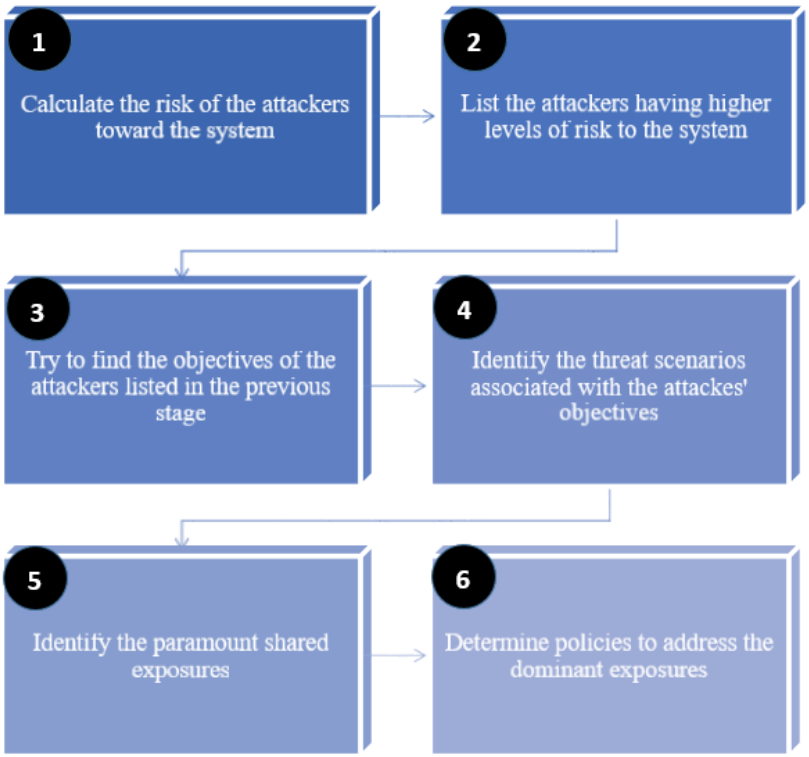

Figure 4.3: Multiple stages of TARA process in detail

4.2 STRIDE

STRIDE is a system-centric threat modelling approach, proposed by Microsoft in 1999, commonly used in their own product development process as well as many other industries including the automotive section [12]. This method is supported by some of the most prominent secure software schemes such as OWASP's Comprehensive, Lightweight, Application, Security Program (CLASP) [48] and Microsoft's SDL [49]. STRIDE is an acronyms induced from different classification of threats that might endanger the system under consideration, these classifications are as follow:

- **Spoofing:** Attackers obtains illegitimate access to sensitive information by manipulating their identity. This threatens system's confidentiality according to CIA triad.

- **Tampering:** Manipulating the data traversing communication channels or stored in a database. This is considered as a violation of the Integrity according to CIA triad.

- **Repudiation:** The inability to trace back an attack to identify the potential attacker.

- **Information disclosure:** Unauthorized access of the attacker to the data in transit or in a database.

- **Denial of Service:** Any attempt of the attacker for disrupting the normal operation of the system and making it out of service.

- **Elevation of privilege:** Attacker gains unauthorized access to a system enabling him to perform critical operations by attaining root privilege in the system.

The whole process of threat analysis by STRIDE model can be described in four consecutive steps [50]:

Building the dataflow diagrams

STRIDE model uses DataFlow Diagrams (DFDs) as an input to the threat analysis process. These DFDs are graphical representations of the system from the attacker's perspective. This is used by security analysts in order to scan the levels of trust throughout a system [51]. Levels of trust are modelled in the DFD as a trust boundary, where the communication of data happens between trusted and untrusted components [52], [53]. Modelling different components in DFD is considered as the initial step that provides the scope for threat analysis [53].

Mapping the dataflow diagrams onto existing threat categories

There are two different ways of performing threat modelling via STRIDE: (i) per element and (ii) per interaction. Comparatively, STRIDE-per-element is more complex to perform since every component of the DFD needs to be analyzed. On the other hand, STRIDE-per-interaction is easier to perform and its treatment strategies are considered as effective enough since most of the cybersecurity threats comprises malicious interactions among system components [54]. These transactions are shown in Figure 4.4.

After building up the dataflow diagram, based on STRIDE-per-element, different components of the diagram need to be analyzed according to six different threat categories mentioned above. However, one might choose STRIDE-per-interaction as it is more convenient to deploy where different malicious interactions between different components need to be classified according to aforementioned threat categories.

Threat Analysis

After mapping system's DFD onto provided threat categories in STRIDE, threat analysis comes into the process. In this stage, checklists provided by STRIDE for each of the six categories need to be examined. These checklists are provided in the form of an attack tree, as shown in Figure 4.5, presenting the hierarchy of

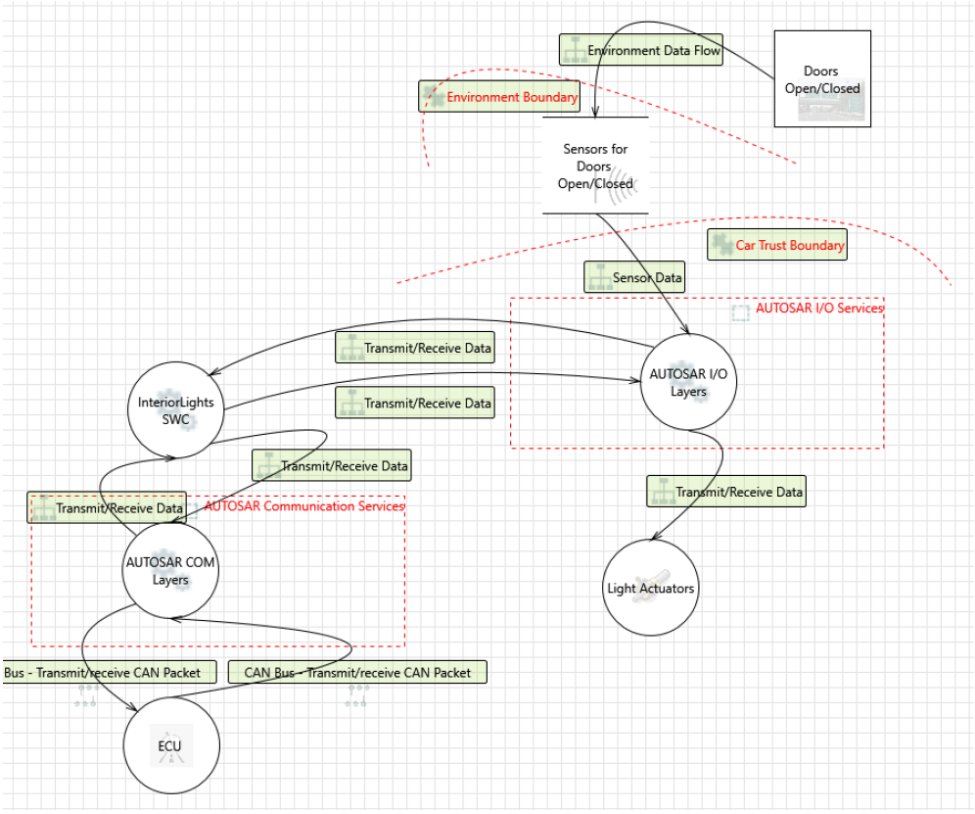

Figure 4.4: DFD created with MS threat modelling tool [55]

threat patterns that can be mapped according to the system under investigation. STRIDE reference book [56] offers twelve different attack trees that each could be used as a checklist for a corresponding category [49]. The reason behind providing a tree-based checklist is to simplify the navigation and rationalization of the relation between different threats. The result derived from this stage is used as an input for the risk assessment process.

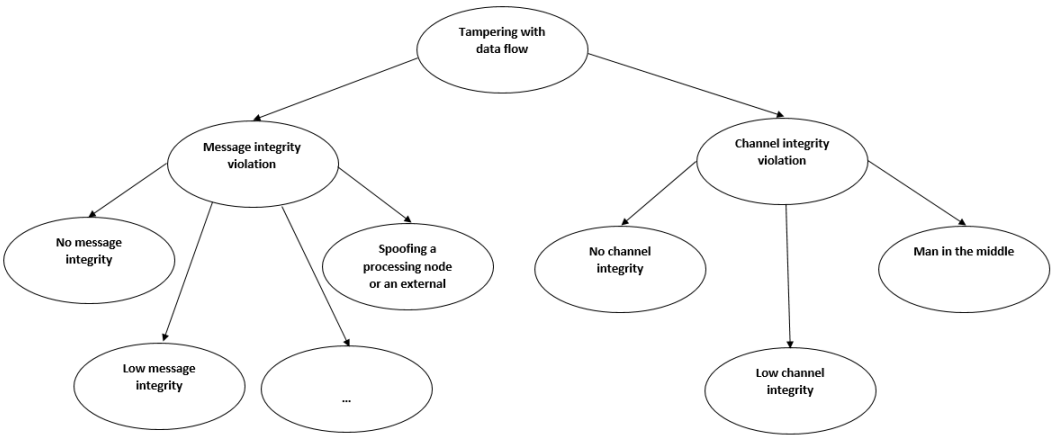

Figure 4.5: Attack tree example showing the data flow tampering

4.3 DREAD

After the threat analysis stage, DREAD method, which is also a Microsoft model, can be utilized during the risk assessment process. This method actually influences the process of quantifying, prioritizing and consequently categorizing the associated risks. It is comprised of five different categories for risk analysis. DREAD is an acronym derived from the initial letters of each of these five categories including [51]:

- **Damage potential:** Quantifying the scope of a damage derived from exploitation of a known vulnerability.
- **Reproducibility:** Ranking the likelihood of a successful exploitation of a known vulnerability.
- **Exploitability:** Quantifying the efforts that an attacker needs for a successful exploitation of a known vulnerability. This could also be considered as a precondition that an attacker might need in order to perform a successful attack.
- **Affected users:** A value representing the number of installed instances of the system that would be affected if an exploit becomes widely available.
- **Discoverability:** This factor specifies the probability that an unpatched vulnerability can be found by external security researchers, hackers, etc.

Each of the five aforementioned categories is valued by DREAD on a rating scale of 0-10. As the rate grows from 1 to 10, it represents higher probability of occurrence with a higher damage potential. Accordingly, the overall risk to the system could also be calculated based on the provided formula shown in Figure 4.6. This formula uses the average of the values of DREAD's five categories. Trivially, the calculated risk always resides between 0-10 where a higher value represents higher risk to the system.

$$Risk\_DREAD = \frac{\textbf{D}\text{amage} + \textbf{R}\text{eproducibility} + \textbf{E}\text{xploitability} + \textbf{A}\text{ffected Users} + \textbf{D}\text{iscoverability}}{5}$$

Figure 4.6: DREAD algorithm's equation for risk calculation

4.4 Attack Graph

Attack graphs are a powerful modelling technique used for representing the paths through which attacks can be performed to reach a malicious goal [57]. In general, nodes in an attack graph depict the states of the system while an attack is happening. These nodes are categorised into starting state, intermediate states and the

goal state. Edges, on the other hand, represent the actions taken by the attacker to transit from one state to another until reaching the final goal.

For instance, as shown in Figure 4.7, consider a scenario in which the attacker (Eve) attempts to gain the root access on a target machine (Alice) by using vulnerabilities on the other hosts (Bob and Charlie) in the system. There is an open port on Charlie's machine with a remotely-exploitable vulnerability. Furthermore, the firewall only allows the traffic destined to Bob's machine to pass and all other traffic is dropped. Bob has the network address of Alice and can freely communicate with her. In addition, Alice's machine does not have a tight security, namely the remote login feature and the FTP service are vulnerable and root access can also be acquired by exploiting buffer overflow.

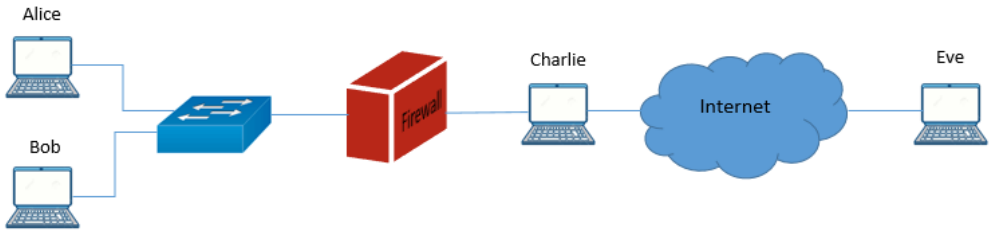

Figure 4.7: Simple scenario for an attacker escalating its privileges

Figure 4.8 represents the attack graph corresponding to the scenario above. The attacker exploiting the vulnerability associated with the open port, gains remote access to Charlie's machine. Once there, the attacker is allowed to communicate with Bob, as the firewall only permits incoming traffic destined to Bob's machine. Subsequent to compromising Bob's machine, the attacker can either use the remote login feature or deploy a new hosts file using FTP to access Alice's machine. Eventually, the attacker exploits buffer overflow to obtain the root access.

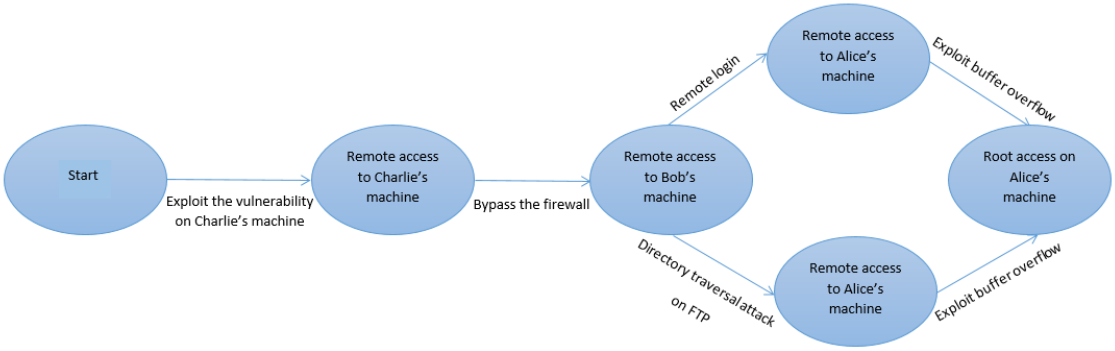

Figure 4.8: Attack graph modelling of the privilege escalation scenario

4.5 Attack Tree

Attack trees are another popular attack modelling technique that are used to analyse different attacks or threats leading to those attacks in order to identify different possible paths to achieve a malicious goal [57]. In other words, they are used to structure the process of identifying attacks endangering a particular system or target.

There are three different types of nodes in an attack tree [57]. The first type are the leaf nodes that are located in the deepest level of the tree and play the role of the initiators of the attack paths. They are in fact attacks or actions that can be performed without any further requirement. The second type of nodes are the OR nodes. When an attack is represented by an OR node in an attack tree, it means that each of the immediate children of this node is a possible path for this attack to take place, in other words, OR nodes indicate possible options for an attack to occur. The third type of nodes are the AND nodes. When an attack is depicted by this node, it means that the immediate children of this node must all happen in order for the main attack to happen itself. Otherwise stated, AND nodes indicate constraints related to the attack on the layer above.

Figure 4.9 illustrates a simple attack tree modelling possible attack paths to acquire a user's password. The red node in the root of the tree represents the goal of the attacker, which in this case is obtaining the user's password. To do so, four different options are available: online guessing, convincing the user to reveal the password, learning when user is typing the password and obtaining the password from the *passwd* file. Since online guessing can be considered as an atomic action, it is not further broken down into smaller steps. On the other hand, persuading the user and learning while typing can each be done in two distinct ways (this means that "persuading the user" and "learn while typing" are OR nodes). In contrast, to retrieve the password from the passwd file, two conditions must hold: accessing the password hash and performing a dictionary attack (this means that the "retrieve from passwd" is an AND node).

Although attack trees look simple and straightforward in the first sight, there are some challenges associated with them that should be taken into consideration. For instance, the attack tree can be drawn in various versions. The root of the tree can be either an asset that is interesting for an attacker or the attacker's goal, which is the attacker's intention for performing the attack. It should be noted that each version will result in a different type and number of trees.

The other issue is the depth of the tree. The atomic attacks and actions need to be identified in order to play the role of the leaf nodes, otherwise each node can be broken down infinitely many times and this can make the tree excessively deep.

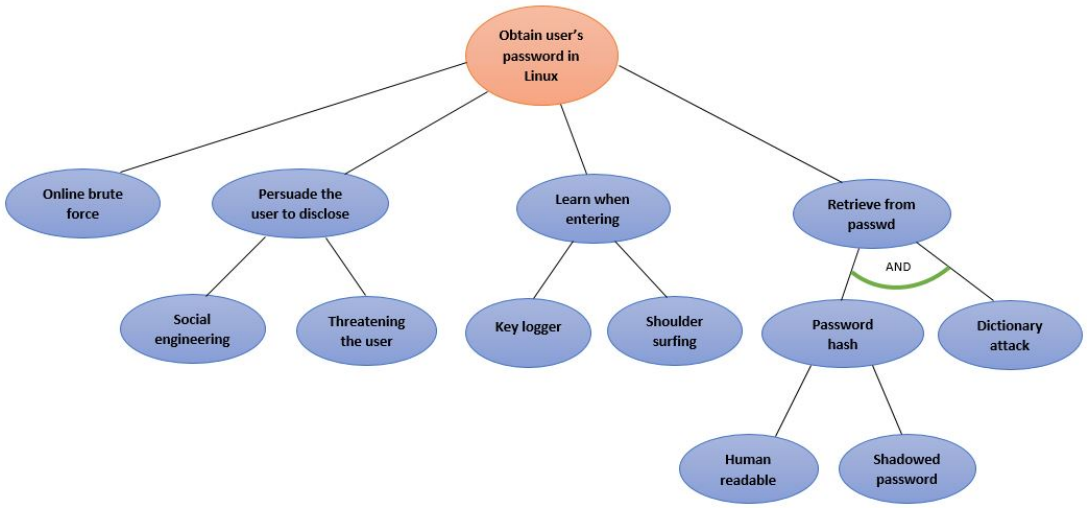

Figure 4.9: Modelling attacks trying to obtain a user's password using attack tree

Attack tree design

There are no predefined rules for drawing an attack tree and its design depends on the taste and the need of the designer. However, some steps can be considered when designing an attack tree:

A) In the first step, the attacker's goal and the threat scenarios endangering a system must be defined. Once the threat analysis and the asset identification processes are finished, the output is a list of threat scenarios.

B) Each threat scenario can be translated into an attack path. In other words, in this step how an attack can be mounted is identified. To have an attack path that is as accurate as possible some knowledge regarding how the attack is actually performed is required.

C) Next, the type of the nodes must be determined. To show the attack paths on an attack tree, nodes, except for the root, can either be the actions taken by the attacker to realize the attack or be the vulnerabilities exploited by the attacker to reach his goal.

D) The process of drawing each attack path initiates with a top-down approach starting from the root. In each step, an attack is broken down into smaller steps until a point is reached where the node can not be further broken down.

4.6 Miscellaneous Models

- **OCTAVE** (Operationally Critical Threat, Asset, and Vulnerability Evaluation) is a risk-based strategic assessment and planning technique for security [58]. It is considered as a self-directed approach. Unlike most of the assessments of a system which focuses on technological risks and tactical issues, OCTAVE highlights organizational risk and focuses on strategic and practical issues. In order to address the security requirements, OCTAVE considers the whole organization and people from both information technology and other operational departments. According to Figure 4.10, OCTAVE claims to assist organizations in balancing three key aspects of any network infrastructure, namely operational risks, security practices, and technology.

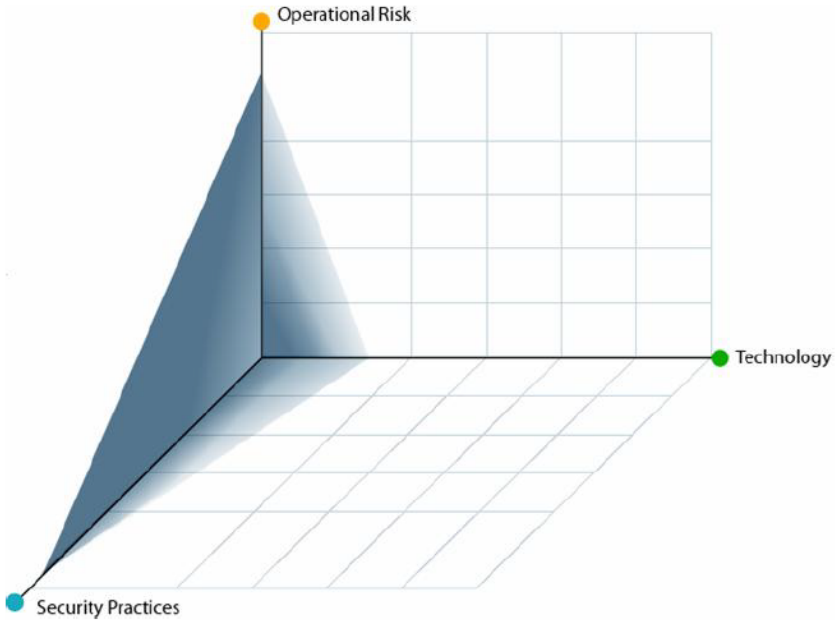

Figure 4.10: Three key aspects balanced by OCTAVE [58]

OCTAVE is an asset-driven evaluation approach. Teams that perform an analysis on a specific system or infrastructure [12]:

1. Identify critical information-related assets (e.g. information and systems).

2. Focus the risk analysis tasks on the assets that are being evaluated as the most critical to the organization.

3. Consider the vulnerabilities (both organizational and technological) of the critical assets, the threats against those vulnerabilities and the relationship among associated assets.

4. Evaluate risks from the operational point of view - how the assets are used in organization's business and how they are at risk because of the security threats.

5. Create a practical defense mechanism for organizational improvement together with a mitigation strategy in order to reduce the risk concerning the critical assets.

According to the aforementioned description, this strategy sheds more light on the organizational aspects rather than technological ones. However, vehicular industry needs a framework that focuses on the technological aspect during the whole product development process. Therefore, we find this strategy not applicable to the needs of automotive sector.

- **PASTA** (Process for Attack Simulation and Threat Analysis) is a strategy that seeks to provide an attack simulation process together with a cyber threats analysis scheme and ultimately to reduce the cyber-crime risks derived from these threats by utilizing mitigation strategies. As shown in Figure 4.11, in order to reach the above goals, PASTA conducts seven consecutive steps for the sake of threat and risk analysis [59]. By following these seven stages, any business can characterize required mitigation factors to address the risks associated with the cyber-threats and consequent attacks to an application.
  PASTA combines system level or application level threat analysis with business objectives, business analysis, and compliance [59]. Despite the fact that it is possible to adapt this method to different environments, since this model focuses on shareholders and business impact of security threats, we consider it as mostly compatible with business models rather than product development process. Therefore, we classify this model as miscellaneous when it comes to the needs of the automotive industry.

- **Cyber Kill Chain** has been developed by an American global aerospace, defense, security and advanced technologies company called Lockheed Martin [60]. This method focuses on different steps that an attacker needs to take in order to accomplish his malicious goals. In each stage, attacker needs to fulfill a specific objective in order to move to the next one. In this way, this method offers a step-wise concentration where the company is recommended to prepare security concerns by focusing on relevant assets in their system. The name of this method is derived from the seven consecutive steps or chain of actions needed to be passed by an attacker in order to penetrate and exploit a system [60].

  These seven steps of Cyber Kill Chain are as follow:

  – **Reconnaissance:** Research, identification and selection of the target.

  – **Weaponization:** Creating a malicious package to be sent to the target.

Figure 4.11: PASTA model of threat and risk analysis

– **Delivery:** The malicious package is delivered to the target by e-mail or other means, and this represents just one of many intrusion techniques the attacker can use.

– **Exploitation:** Refers to the actual execution of the malicious package on the target system.

– **Installation:** Refers to installing a backdoor trojan or similar which would grant remote access to the target machine over a longer period of time.

– **Command and Control:** Establishing an outside connection or a channel by which the attacker can gain "command and control" over the target machine from a remote location.

– **Actions on objectives:** This is the final step of the attack which can take months to successfully be performed. The attacker performs actions that would accomplish his initial goal [60].

One of the advantages that this method poses is that it tries to analyze the attacker's behavior pattern and mindset while focusing on the target asset. In order to provide security countermeasures to the existing threats and defend the system from future possible attacks, the corporation needs to analyze the chains of attacks that were performed earlier. Hence, it is expected that if this method is performed correctly, the company can always be one step ahead of the attacker from the cybersecurity point of view. Again, like the two other methodologies described earlier in this section, Cyber Kill Chain is more useful for the business level rather than product development. Thus, we list this method as a miscellaneous one.

5 Feasibility

Once the attack paths are determined, an estimation of how easy or hard it is to exploit an attack path should also be taken into account. This estimation is known as attack feasibility. Since feasibility calculation should be able to be done in both the concept phase, in which all details are not known, and the design phase, in which more concrete features of the system are known, some parameters must be defined to ease the estimation process. The definition of these parameters, also known as attack potentials, are explained in the following section.

5.1 Analysis parameters

Several attack potentials can be considered when it comes to estimating how likely is for an attack to happen. The five prominent ones that will directly be used to calculate the feasibility of an attack are described in this section. Furthermore, in Chapter 9, the reason for excluding the rest of the parameters from the calculation process is explained.

In order to ensure the compliance with the recent trend in the automotive industry, the names of these parameters are kept the same as in ISO/IEC 18045.

Equipment (Eq) or the availability of resources refers to hardware or software equipment required for exploiting an attack path:

- Standard. The equipment is readily available and easily accessible to the attacker. This equipment can also be a part of the TOE itself, e.g. a built-in debugger in the operation system. Examples for this type can be a notebook or a simple diagnostic device that can be purchased for a moderate amount of money.

- Specialised. The equipment is not readily available to the attacker, but can be obtained without excessive effort. This equipment is more automotive specific and can be a developed script or program or even a standard tool but for a higher price. Examples for this type can be an automotive-specific debugger or an RF monitor or a costly diagnostic device.

- Bespoke. The equipment is not readily available to the public and its distribution is restricted. This equipment is normally specially made and is considerably more expensive that the previous categories. Examples of this type are tools that are custom-made.

- Multiple Bespoke. This type allows a situation in which various types of Bespoke equipment is needed to exploit an attack path.

Expertise (Ex) or the skill level refers to the amount of knowledge required to exploit an attack path:

- Layman. The attacker has very limited knowledge about the vehicles. He has no expertise in the automotive domain and normally has only IT skills for domestic usage. In addition, he can only use ready-to-use tools with clear instructions.

- Proficient. The attacker has general knowledge about the vehicles and is familiar with security behavior of the product, as he is probably involved in the automotive industry. He can not develop new attacks, but can use available tools to exploit an attack path even if the instructions are not clear.

- Expert. The attacker has knowledge about the existing algorithms, protocols, hardware, software, cryptographic approaches, security behaviors and concepts in the security domain and the automotive industry. He is able to develop new tools and methods to exploit attacks paths.

- Multiple Experts. The attacker has multiple fields of expertise to exploit attack paths. Here, the fields of expertise must be distinct from one another for the attacker to fall under this category .

Knowledge of the TOE (Kw) or awareness refers to specific expertise in terms of information about the system under scrutiny required by the attacker to exploit attack paths:

- Public. The information related to the TOE is accessible by public. Information available on the Internet or in a book that can easily be purchased by everyone fall under this class. For instance, the information regarding communication protocols such as Transmission Control Protocol (TCP) or CAN.

- Restricted. The information related to the TOE is shared and controlled by partners under a non-disclosure agreement. An example of this class can be information shared among involved organizations such as the supplier and the manufacturer.

- Sensitive. The information related to the TOE is shared among teams within one organization and the access is only restricted to the members of those teams. Information such as source code or the vehicle configuration database fall under this category.

- Critical. The information related to the TOE is shared among a few individuals and access to this information is strictly restricted on a need to know basis to those individuals. Information such as security keys fall under this class.

Window of opportunity (Wo) or opportunity refers to the amount of access required by an attack to be carried out:

- Unlimited. The attack does not need any opportunity to be realized as there is no risk of being detected during the access. In other words, the TOE is available without any time limitation. Unlimited physical access to the target or always-on Internet connection fall under this category.

- Easy. The attack only requires less than a day of access to the TOE to be realized.

- Moderate. The attack only requires less than a month of access to the TOE to be realized.

- Difficult. The attack requires at least a month of access to the TOE to be realized.

  In case the window of opportunity is not sufficient enough, in other words, the available time to exploit an attack is less than the time required by the attack to be realized, the attack would not be possible to be performed.

Elapsed time (Et) refers to the time required to exploit an attack path. Identification of a vulnerability and exploitation of the corresponding attack path may need considerable time. However, since the time needed for identifying a vulnerability may significantly differ from the time needed for exploiting an attack path due to some probable intervals in the vulnerability identification phase, for the sake of accuracy, only the amount of time needed for exploiting an attack path is considered as the elapsed time.

- Less than a day. The attack takes less than a day to be performed.

- Less than a week. The attack takes less than a week to be performed.

- Less than a month. The attack takes less than a month to be performed.
- More than a month. The attack takes more than a month to be performed.

5.2 Parameter rating

Likelihood estimation of an attack path is possible through assigning a value to each of the parameters explained in the previous section. Since each attack potential is classified into four levels, values 0-3 is assigned to each of them, as shown in Table 5.1. Value 0 is assigned to the worst case or the most relaxed condition while 3 is assigned to the least relaxed one.

Table 5.1: Attack potential values

Parameter	Value
Equipment	
Standard	0
Specialised	1
Bespoke	2
Multiple Bespoke	3
Expertise	
Layman	0
Proficient	1
Expert	2
Multiple Experts	3
Knowledge of the TOE	
Public	0
Restricted	1
Sensitive	2
Critical	3
Window of opportunity	
Unlimited	0
Easy	1
Moderate	2
Difficult	3
Elapsed time	
Less than a day	0
Less than a week	1
Less that a month	2
More than a month	3

Once all the parameters are evaluated, the overall feasibility of an attack path must be calculated. A straightforward and linear formula, as shown in Figure 5.1, is used to ease the reasoning and the calculation of the feasibility. For each parameter, the final value corresponds to the multiplication of its value in its weight. However, due to complicated interrelation of the parameters, we do not assign weight (priority) to the parameters in this stage and assume that all of them are equally important (w=1).

$$Sum = v_{Eq}w_{Eq} + v_{Ex}w_{Ex} + v_{Kw}w_{Kw} + v_{Wo}w_{Wo} + v_{Et}w_{Et}$$

Figure 5.1: Linear formula for feasibility calculation

The calculated *Sum* will be later mapped to the corresponding feasibility value in Table 5.2. It should be noted that a basic attack path, for instance an attack path having standard equipment, a non-expert attacker and public knowledge of the target with no need for an opportunity to be realized, is more likely to be exploited, and consequently its feasibility would be higher. In contrast, higher required level of the attack potentials makes the exploitation of the attack paths more difficult, and consequently the corresponding feasibility would be lower.

Table 5.2: Feasibility calculation framework

Sum	**Feasibility**
0 - 2	Very high
3 - 6	High
7 - 9	Moderate
10 - 12	Low
>12	Very low

In the following chapter, some scenarios are evaluated by means of the attack trees. Once the attack paths are discovered in each scenario, in order to present examples of feasibility calculation, the feasibility of some of the paths will be calculated.

6
Use cases

In this section, we model two scenarios by means of attack trees to demonstrate how our nominated modelling technique works in the automotive world. It is worth mentioning that the reason for selecting attack trees over other methodologies is thoroughly explained in Chapters 7, 8 and 9.

The type of the attack tree we use to model such scenarios is the one in which the root of the tree is the attacker's goal and not the asset itself. There are two reasons behind this decision: first, if a scenario contains multitude of assets, using each asset as the root of the tree results in one attack tree for each asset. Given the real-world use cases where tens of assets are involved in a scenario, this would result in unnecessary many trees, which consequently makes the analysis difficult. Second, based on our experiments, selecting the attacker's goal as the root of the tree results in a more comprehensive and realistic attack tree. Furthermore, in order to have a better alignment with the CIA triad classification, we decided to select the CIA triangle faces as the root of the tree in all scenarios. As a result, in both of the use cases, the attacker's goal is violating one or more of the confidentiality, integrity or availability of the system, where applicable.

Prior to explaining each use case, the reference architecture on which our scenarios are defined should be presented. Figure 6.1 depicts the HoliSec reference architecture [33] that inherits all of its attributes from HEAVENS, except for some features that have been improved [61].

As it can be seen in the figure, engine and brake ECUs that are categorized as power train ECUs are connected to the main gateway via CAN network. The connectivity gateway which is responsible for infotainment and telematic is connected to the back office and is equipped with both physical and remote interfaces. The two other entities in the architecture are V2X ECU and the driver display that have remote

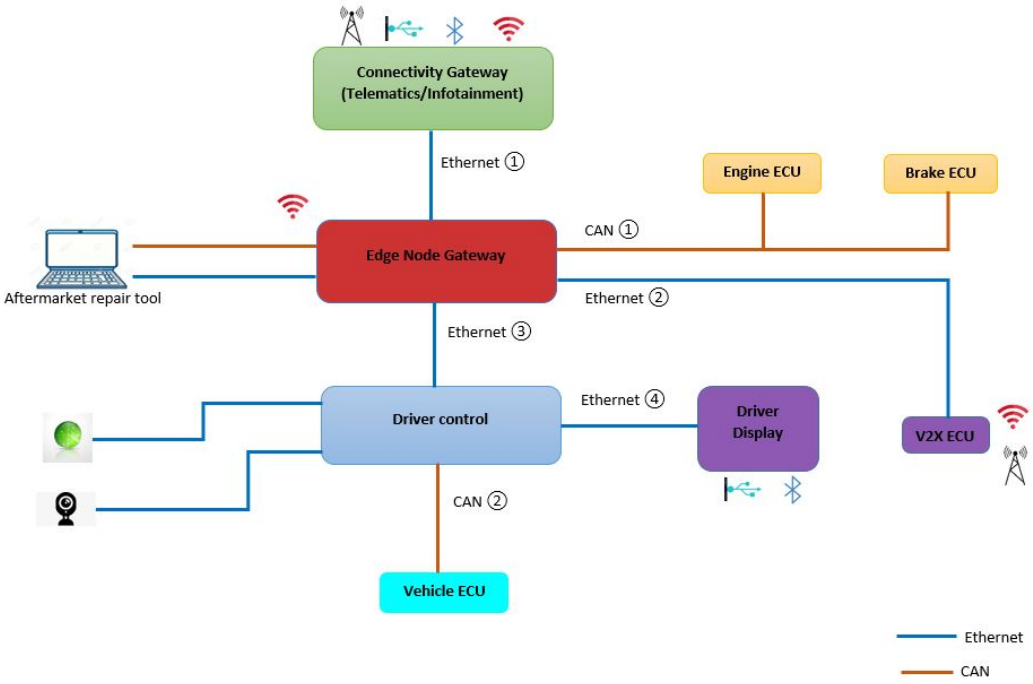

Figure 6.1: HoliSec reference architecture

communication with other vehicles and the vehicle passengers, respectively. The driver control on the other hand is connected to the sensors, namely the camera and the radar to both serve the user with some information via the driver display and to send information to other components in the architecture. Finally, the vehicle ECU represents a general ECU in a vehicle that handles several responsibilities.

There are two physical networks connecting ECUs and gateways in this generic architecture. The CAN network whose benefits were discussed in Section 2.3 and the Ethernet network which is automotive specific and is normally used where high bandwidth is needed [32]. Aside from the physical connections, as illustrated in Figure 6.1, some of the ECUs and gateways are also equipped with wireless interfaces. All these interfaces can be categorized into two groups of internal and external.

Table 6.1 lists the ECUs in addition to their software architecture, interfaces and a summary of their role in the system.

Table 6.1: HoliSec reference architecture entities

Units	**Software architecture**	**Interfaces**		**Role**
		Internal	**External**	
Vehicle ECU	AUTOSAR	CAN	—	Sends speed signals and responsible for the warning lights
Engine ECU	AUTOSAR	CAN	—	Responsible for the engine system
Brake ECU	AUTOSAR	CAN	—	Responsible for the breaking system
V2X ECU	Linux	Ethernet	WiFi and 4G	Communication with other vehicles, infrastructure, etc.
Connectivity gateway	AUTOSAR (internal), Linux (External)	Ethernet	WiFi, Bluetooth, 4G and USB	GPS position, collects fleet information to send to the back office, has the immobilizer status
Edge node gateway	AUTOSAR	CAN and Ethernet	CAN, Ethernet and WiFi	Diagnostic response to external resources, translates and forwards signals between networks
Driver control	AUTOSAR	CAN and Ethernet	Bluetooth and USB	Translates and forwards signals between networks
Driver display	Linux	Ethernet	—	Visualizing information for the driver

Before explaining the use cases and how our proposed methodology is used to model attack paths, some general assumptions must be made that hold for both of the use cases: (1) Only the ECUs having a role in the scenarios are evaluated in each use case. (2) In the reference architecture we received, some ECUs were not explained in details. Hence, we had to make some assumptions for each of them in terms of interfaces, software architecture and how they play their role in the scenario. Nevertheless, the assumptions are close to the real-world scenarios based on the interviews we had with the industrial experts. (3) Attack trees can be deepened almost indefinitely, however, as mentioned before, it is important to decide where to cut the tree. In our use cases, we deepened the trees based on the information we managed to acquire about each attack path and as long as each level was still in the automotive domain. (4) An attacker may have two different types of goal. First, the moral goals such as threatening passenger's life, annoying the driver and the like and second, the technical goals such as disabling a service or manipulating a parameter. In the following use cases, the technical goals are addressed only. (5) Attacks are initiated from potential attack surfaces, hence, whenever possible, we tried to have the attack surfaces as the leaves of the tree, and (6) Wherever possible, the attack path is divided into a path starting from a physical interface and a path starting from a remote interface. This separation is done because the feasibility of an attack path may differ when started from a physical or a logical interface.

6.1 Use case 1: GPS positioning and warning lights

In this scenario, the vehicle gathers the GPS position from the connectivity ECU as well as the warning lights from the vehicle ECU. This information is broadcast from the V2X ECU to other nearby road vehicles. Figure 6.2 shows this scenario as well as the signal paths.

Signals: warning light signals are broadcast from the vehicle ECU. They traverse the CAN network and are forwarded by the driver control to the Ethernet network. Later, the edge node gateway forwards the signals to another Ethernet network. Finally, they are received by the V2X ECU to be broadcast to other vehicles. On the other hand, GPS position signal is sent from the connectivity gateway over the Ethernet network. Edge node gateway forwards this signal to another Ethernet channel. Eventually, the signal is received by the V2X ECU to be broadcast to other vehicles.

Assumptions

1) In this scenario, we do not see violating confidentiality as an issue, as both the warning light signals and the GPS signal are going to be broadcast to other entities on the road. Hence, only attack paths violating integrity and availability are addressed.

2) The OBD port is considered a physical interface, however nowadays there are some dongles in the market that can also provide a remote access to this port.

3) As the detailed implementation of the architecture and the exact process of mounting the attack paths are not available, the attack paths will be drawn in a high-level of abstraction.

4) GPS signals are received by the external interfaces of the connectivity gateway governed by a Linux operating system. The signals are later encapsulated into TCP packets to be sent to the internal network of the vehicle.

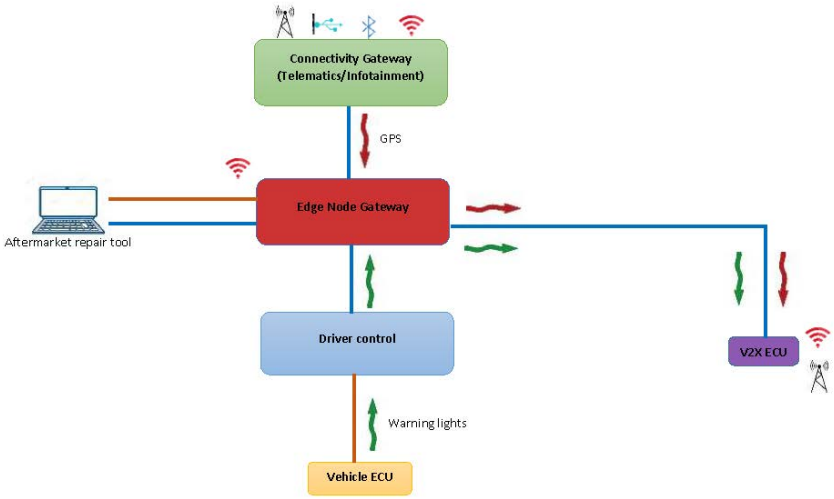

Figure 6.2: GPS positioning scenario

Attack path modelling

Here, for each of the violations related to this scenario, one attack tree is designed. As shown in Figure 6.3, GPS spoofing can cause a malicious change in the direction of a vehicle. This is feasible either via GPS signal manipulation or false GPS signal injection. Injecting false GPS signals can be done through the connectivity gateway, where the GPS signals are transmitted, through the edge node gateway, which plays the role of a terminal in passing the signals, or by compromising the V2X ECU that is located in the edge of the system. GPS signal manipulation on the other hand, can be performed by altering the GPS packets transmitted from the Linux operation system on the connectivity gateway into the internal AUTOSAR architecture and then into the internal network.

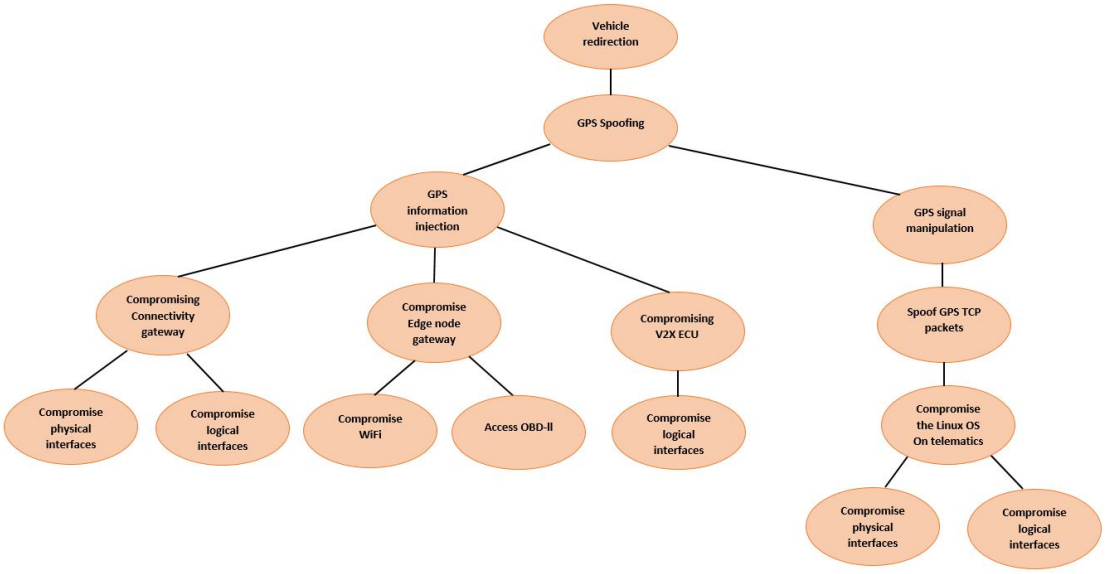

Figure 6.3: Vehicle redirection attack

Figure 6.4 illustrated the second attack in this scenario. To broadcast false information, either the GPS signals or the warning light signals can be falsified. GPS signal spoofing sub-attacks are the same as the ones in the previous attack mentioned above. Falsification of the warning light signals can be done via signal injection or manipulation. To do so, the attacker has to compromise the driver control, the edge node or the V2X ECU. Since the sub-trees of compromising the edge node and the V2X ECU in the right sub-tree are similar to the ones in the left sub-tree, they are not expanded and are shown by red dashed circles.

In contrast to the two previous attacks that violate the integrity of the system, the attack depicted in Figure 6.5 violates the availability of the system. In order to hinder the intra-vehicle communication, the signals must be possible to be transmitted via the V2X ECU. Hence, attacks such as GPS jamming, WiFi jamming, cellular jamming or a DoS attack on the vehicle are all applicable.

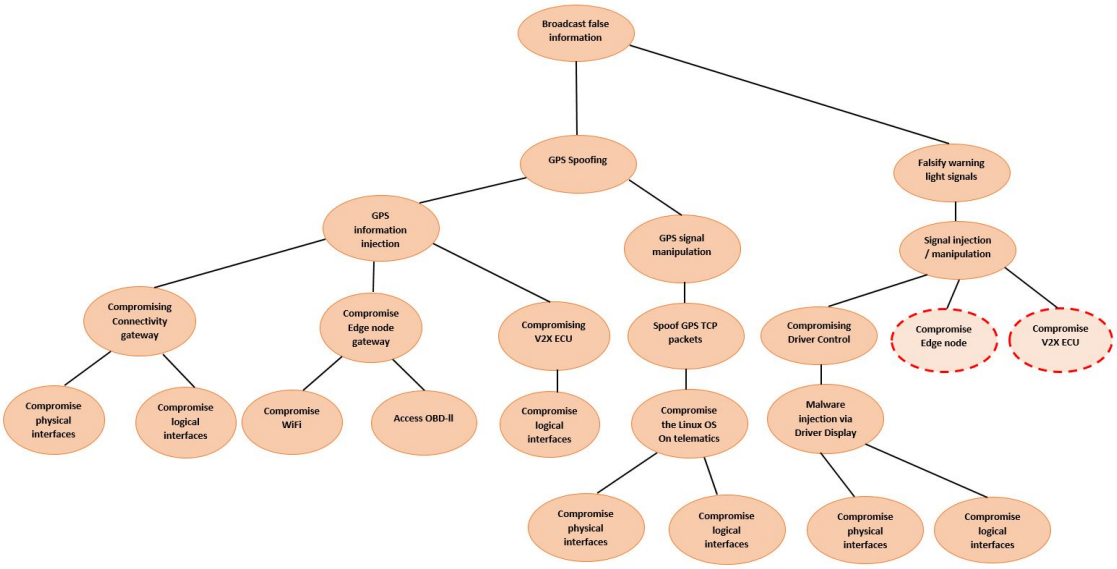

Figure 6.4: Broadcasting false information

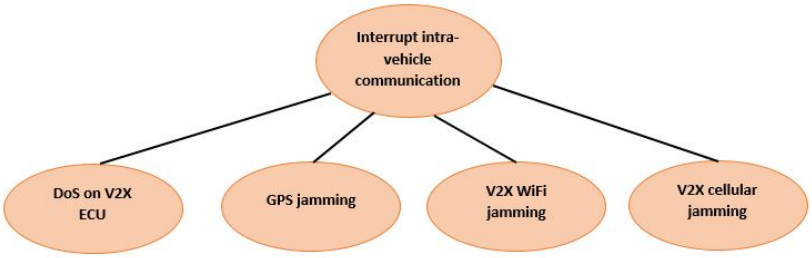

Figure 6.5: Intra-vehicle communication hindrance

In order to have all the attack trees concerning this scenario in one picture, a node named after the faces of the CIA triad is used as the root of the final tree. In this case that is shown in Figure 6.6, the root expresses the attacker's goal and this goal is divided into all potential attacks that can make this goal feasible. Repeated paths in the tree are not expanded and are shown by red dashed circles.

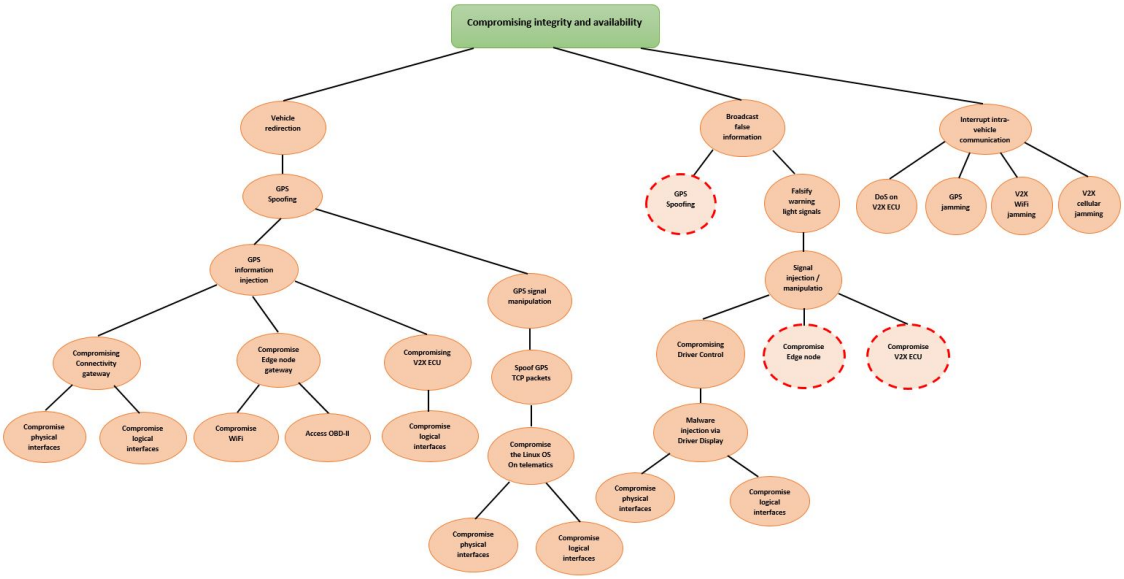

Figure 6.6: Attacks toward the GPS and the warning lights in one picture

Now that the final attack tree is designed and the attack paths are determined, we calculate the feasibility of two of the attack paths as an example of how this calculation can be done. Figure 6.7 shows the selected paths.

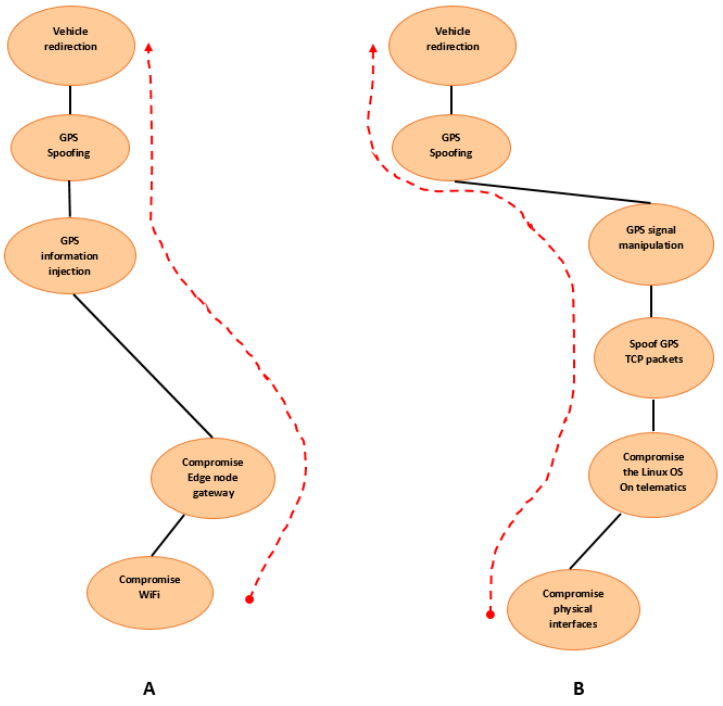

Figure 6.7: Example of feasibility calculation in GPS positioning use case

Part A and B in the figure represent two different attack paths toward GPS spoofing and consequently the vehicle redirection. Since exact information about how these attack paths are actually developed is not available, the values assigned to the attack potentials associated with each path are done based on several discussions among the authors of this report and they may change once more information is available. For both cases, we assume that the attacks should be performed by an expert who knows the underlying behaviour of the system. In addition, the equipment required for these attacks to be done is at least specialised, if not bespoke. Next, we assume that the required knowledge about the target falls under the restricted category. In contrast to path A that starts from a remote interface, for path B to happen, physical access is required and this means that path B needs a larger window of opportunity to be developed. Eventually, we consider that for both of the attacks to be mounted, less that a week is enough.

Table 6.2 represents the value assignments and the final feasibility of each path. As it can be seen, feasibility of path B is moderate while for path A it is high. The reason is the required physical access in order to perform path B which results is a more difficult or in other words a less feasible attack.

Table 6.2: Feasibility calculation for two paths of the GPS spoofing attack

Path	Equipment	Expertise	Knowledge of TOE	Window of opportunity	Elapsed time	Sum	Feasibility
A	1	2	1	0	1	5	High
B	1	2	1	2	1	7	Moderate

6.2 Use case 2: Target cruise control speed

In this use case, cruise control function which is located in the vehicle ECU is activated and then the driver must set the speed. As shown in Figure 6.8, two signals are needed by the engine ECU in order to adjust the target speed: the current speed signal and the target speed signal.

Signals: vehicle current speed and vehicle target speed are broadcast from the vehicle ECU into the CAN network. The driver control forwards these signals into the Ethernet network. Later, the edge node gateway forwards the signals into the CAN network and then, into the engine ECU.

Assumptions

1) In this scenario, we do not see violating confidentiality as an issue. Hence, only attack paths violating integrity and availability are addressed.

2) Vehicle and Engine ECU are equipped with some Road Speed Limit (RSL) parameters that are used for adjusting the speed.

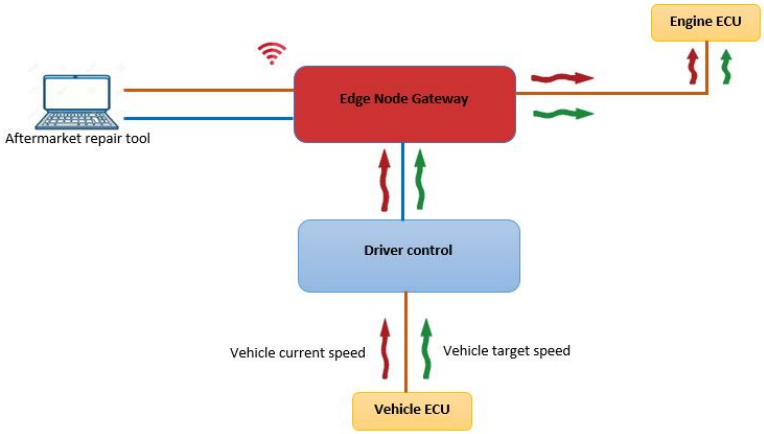

Figure 6.8: Cruise control scenario

3) Both target speed and current speed signals are required for the engine ECU to function properly.

4) Since Engine ECU is considered as a safety-critical power train ECU, the internal RSL parameters can not be altered and can only be bypassed.

5) In CAN networks, if several messages are sent at the same time, the message with the highest priority (based on its ID) is delivered and the rest of them are dropped and have to be re-transmitted.

Attack path modelling

In this scenario, same as the previous use case, for the violation against each of the CIA faces, one attack tree is designed. As shown in Figure 6.9, manipulating the cruise speed is possible through bypassing the RSL parameters either in the vehicle or in the engine ECU or by injecting or changing the speed signals. Speed signal alteration can be performed through compromising the edge-node gateway or the driver control, using their physical or remote interfaces. These attacks violate the integrity of the system.

Figure 6.10 depicts the second attack toward the cruise control system. In order to prevent the cruise system from operating, high-priority massages should be injected into the CAN network to prevent real speed signals from being delivered. In this case, the availability of the cruise control is violated.

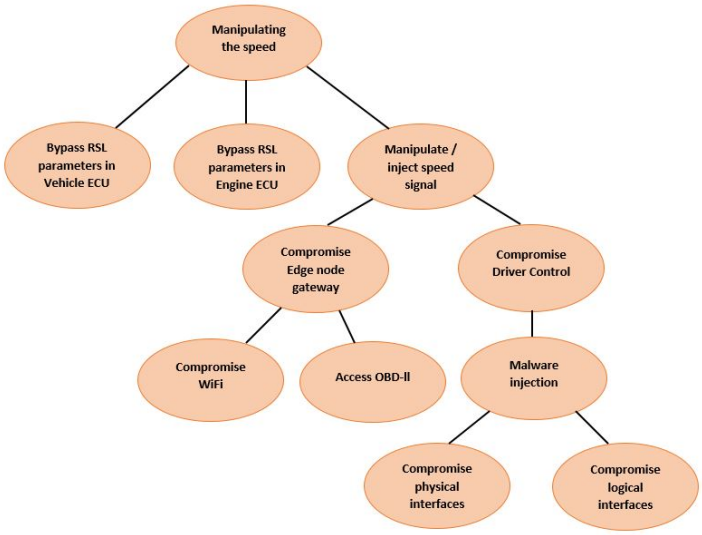

Figure 6.9: Cruise speed manipulation

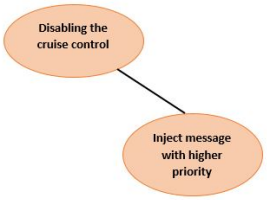

Figure 6.10: Prevent the cruise control from functioning

Once all probable attack paths are discovered, as shown in Figure 6.11, a node named after the CIA faces is used as the new root of the tree to connect all trees together in order to represent the entire picture of the use case. Another reason for the selection of such node as the root of the tree in both use cases is to address one of the challenges regarding the attack tree which is choosing an adequate goal for the attacker in order to have a more comprehensive tree.

Now that the final attack tree is designed and the attack paths are determined, again we calculate the feasibility of an attack path as an example of how this calculation can be done. Figure 6.12 shows the selected paths. For this attack to be mounted, we consider that the attacker must be an expert and must have information about the architecture and the underlying protocols. Furthermore, specialized equipment is needed for this attack to be realised while the attacker has unlimited access to the target and only needs the publicly accessible information in order to successfully perform this attack. Finally, the required time to exploit the vulnerabilities that have made such attack path possible is less than a week. The values assigned to the attack potentials and the resulting feasibility are presented in Table 6.3.

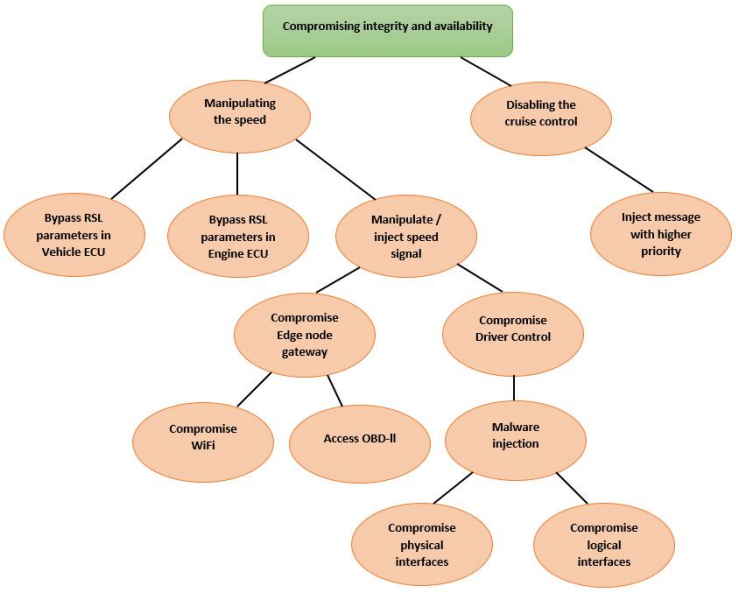

Figure 6.11: Big picture of the attack paths in the cruise control scenario

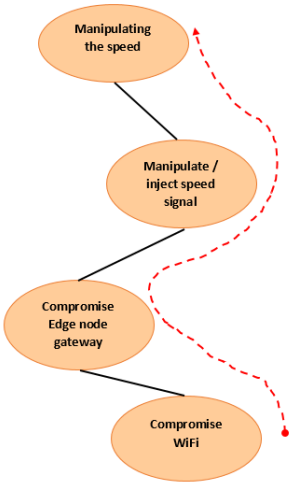

Figure 6.12: Example of feasibility calculation in cruise control use case

Table 6.3: Feasibility calculation for speed manipulation

Equipment	Expertise	Knowledge of TOE	Window of opportunity	Elapsed time	Sum	Feasibility
1	2	0	0	1	4	High

7 Method

Once the scope of the thesis was defined, as the automotive security topic was a new domain to the authors of this thesis and in order to answer the first research question, related security literature had to be surveyed in the first place. This survey part of the thesis was divided into three separated steps: literature related to the attack modelling techniques, literature related to the existing attacks in the automotive world and literature related to the calculation of feasibility. However, since the attack analysis was a new phase in the automotive risk assessment framework (in the previous version of risk analysis, the output of the threat analysis step, namely the identified assets and the extracted threat scenarios, was directly used as the input to the risk assessment procedure), the security literature found was almost all related to threat analysis frameworks. Nevertheless, due to the similarities between the threat analysis and the attack analysis procedures, as described in Chapter 8, all the material concerning the threat analysis was also applicable to the attack analysis phase.

Once all the modelling techniques used as attack analysis methodologies were found, it was almost decided that which of the existing methodologies was more suitable to the automotive domain. To do this, since the methodologies were different in nature, they had to be assessed from dissimilar perspectives. Three of the existing methodologies were only used in the decision-making processes and not in the product development process. Hence, as we needed a methodology to analyze attacks in the product development cycle, these three methodologies had to be excluded.

Among remaining methodologies, STRIDE and DREAD have been widely used for threat analysis purposes and some of the parameters in the model were not applicable when trying to model attacker's goals, parameters such as non-repudiation, authenticity and the like. On the other hand, TARA has been a comprehensive risk assessment framework whose goal is focusing on optimising the risk assessment process. This framework would not have fit into the attack analysis phase. Put

differently, it was more practical for the proposed attack analysis methodology to be included in TARA framework and not vice versa. The remaining methodologies after this filtering were applicable to the product development process and could be utilised for the attack analysis phase.

Afterwards, in order to find an answer for the second research question, interviews were conducted with industrial security experts to find the criteria based on which these methodologies could be assessed. The reason for these interviews apart from studying related papers, was to collect information that was as accurate as possible. However, both the literature study and the interviews resulted in some informal criteria for assessment of the methodologies. Thus, a new set of interviews were conducted with academic experts. Similarly, the final results were again informal and similar to the ones derived from the previous set of interviews. These criteria are listed in Chapter 8 as reasons behind selecting the final methodology.

The second phase of the thesis concerned the feasibility calculation of the attack paths modelled using the nominated methodology. For this, a comprehensive study of the previous related work as well as the ISO/IEC 18045 was carried out to determine and evaluate the common criteria based on which an attack feasibility could have been estimated. After a few meetings and discussion sessions, the proposed formula and feasibility table, illustrated in Chapter 5, were picked as the suitable framework for feasibility calculation.

Next, to indicate how the nominated methodology works and how the feasibility can be evaluated using the proposed formula and the feasibility table, some use cases, which were provided by Volvo Trucks, were used as examples of how the attack analysis procedure can be performed. Eventually, there were two issues that needed to be discussed: the reasoning behind some of the decisions made through the thesis and the potential future works as a continuation of this thesis. In order to address both of these issues, a few meetings were held between the involved individuals in the thesis. The derived results of these meetings are listed in Chapter 9.

8 Results

Attack tree as the well-suited solution
Subsequent to our extensive survey of attack analysis methodologies, we selected the attack tree approach as the best-suited methodology to be used to model automotive attack scenarios. The first reason behind this decision is that attack tree is an uncomplicated and straightforward visual method to model various attack paths in a target system [62]. Since everyone working in the automotive domain is not an expert in all related fields, such as cybersecurity, safety, embedded systems and the like, a comprehensible-enough methodology is needed to aid everyone with reaching a shared understanding. The second reason is that an attack tree can be used for modelling attack paths in high levels of abstraction [63]. In other words, unlike DFDs that are used in some of the methodologies for threat analysis, attack trees can eliminate unnecessary technical details such as messages exchanged among different parties to increase the readability of the analysis methodology.

Furthermore, in contrast to attack trees, some of the surveyed methodologies are more applicable to the domains other than the automotive industry. For instance, the OWASP model is commonly used for web application and its adaptation to the automotive world requires serious effort, if not impossible. On the other hand, models such as OCTAVE, PASTA and Cyber Kill Chain, as mentioned in Section 4.6, are mostly used in business level for decision-making purposes and not in the product development process.

The final driving reason behind proposing the attack tree is the current trend in the security domain [57]. It is easier for the security experts, either academic or industrial, who have been using attack trees to model security threats, to use the same trend for modelling the corresponding attacks. In addition, several other trees can be derived from attack trees, such as protection trees, defense trees, attack-defense trees that can be later used by the experts in the risk assessment process

[64]. According to our investigations, attack trees have been used in the IT domain without any explanation regarding the reason behind the selection of this modelling tool. We believe that this selection that seems to be unintentional is due to the flexibility of attack trees. Put differently, they can be drawn in different phases of the product development life cycle, namely the concept phase, where details about the system are not clearly known, and in the design phase and beyond, where more details concerning the system are available [65].

Eventually, it should be noted that although attack trees and attack graphs seem to be from the same family, attack trees have some significant advantages over the attack graphs. Attack trees can be drawn in both the concept phase and the design phase [65]. In contrast, attack graphs are only beneficial when the details about the system are known, for instance, in the test and verification phase. To make the most out of the attack graphs, a comprehensive vulnerability scanning is needed. The output of this scanning phase would be later used to draw the attack graph [66]. It should be noted that due to the high complexity of drawing attack graphs in real-world scenarios, it is not feasible for the attack graph to be drawn by hand [67], hence some advanced tools are required both for the vulnerability scanning and the drawing steps [68].

Attack analysis VS Threat analysis

In the automotive world, attack analysis and threat analysis terms are used interchangeably. Throughout this thesis, we found that these two terms represent the same concept but from different perspectives. Put differently, attack analysis is system analysis from the point of view of the system owner, while threat analysis is from the point of view of the attacker. Figure 8.1 illustrates the differences and the similarities of these two concepts.

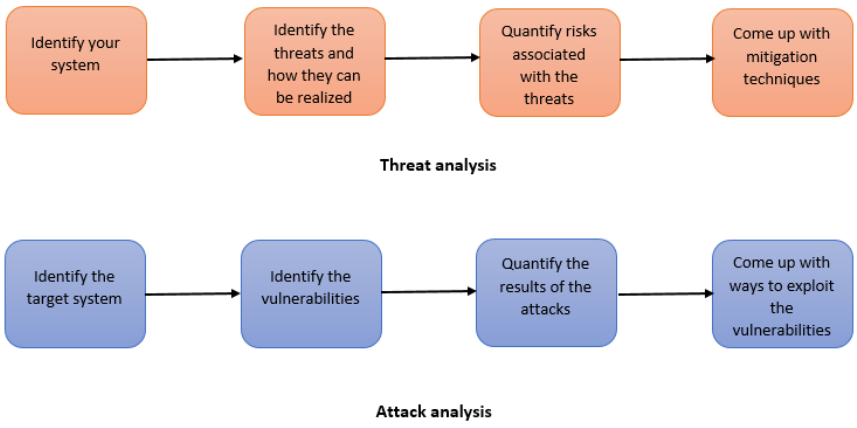

Figure 8.1: Threat analysis VS Attack analysis

Both techniques start with defining and identifying the system under evaluation. In the next step, in threat analysis, the owner of the system tries to list existing threats endangering the system, while in attack analysis, the attacker tries to list all vulnerabilities that can be exploit toward compromising the assets. Once done, the threat analysis calculates the risks associated with each threat, while attack analysis lists the consequences associated with exploiting the vulnerabilities of the system. Finally, in threat analysis, the analyst tries to find countermeasures for the potential threats, while the attack analyst tries to find a way for exploiting potential attack paths.

Feasibility and Impact
In the risk assessment process, impact of an attack or a threat leading to an attack is also taken into account as well as the feasibility factor. The reason behind this consideration is lying behind the word "reasonable". In the automotive industry, similar to many other industries, time and cost are paramount factors. There may be a threat or an attack with high feasibility but with an insignificant impact. On the other hand, there might be attacks with trifling feasibility, but devastating impacts. Thus, when conducting a comprehensive risk assessment, both the feasibility and the impact must be included in order to determine how much of the existing resources must be dedicated for the mitigation techniques.

9 Discussion

Attack analysis methodology
As mentioned in Section 4.5, there is no rule stating how an attack tree must be drawn and its design purely depends on the designer and the need of the company. So far, attack trees have been used in their classic mode, however, same as other modelling tools, attack trees can be customised due to the need of the user. In order to have a better attack tree in terms of quality and accuracy, at least two areas of expertise are needed: an architecture expert who knows the entire architecture and the existing interfaces that can later be used as attack surfaces, and a penetration testing specialist who knows how attacks are actually performed.

Aside from the reasons mentioned in Chapter 8, another reason of preferring attack trees to attack graphs may be the lower complexity of the algorithms run on attack trees given their simplified structure. We did not mention this as a reason in our result section because we did not investigate the mathematical aspects of each of these modelling techniques. This part is considered at the end of the discussion chapter as a future work.

Feasibility calculation
Aside from the parameters we used for feasibility calculation, other parameters may be possible to be added to the formula in future. Instances of some parameters to be included in the attack feasibility calculation can be: the number of steps an attacker must take to perform an attack, the attacker's motivation, the value of the asset and the resistance of the TOE. Although these parameters seem reasonable to be deployed, they must be carefully quantified or qualified. Here, the number of steps to realize an attack heavily depends on the drawing of the tree. In other words, the deeper the tree, the longer the attack path. Hence, since there is yet no standard to state what the best level is to cut a tree and what the atomic attacks should be, we decided not to include attack path length in the feasibility formula. The attacker's

motivation is considerably hard to model and rate, as except for the two extreme levels (highest and lowest), intermediate levels of motivation is hard to evaluate. The value of the asset and the resistance of the TOE are subjective parameters and can not be easily assessed.

Moreover, it should be mentioned that since we live in a dynamic world, rating of each parameter can change over time. For instance, because of the dramatic growth in technology, new tools and devices are introduced to the world almost every year. Hence, the equipment used by the attackers in future might be more advanced compared to the ones used nowadays. Another example can be the knowledge about the target of evaluation. Clearly, information regarding a newly-built product or system can not be easily obtained. Hence, the knowledge of the TOE can be evaluated as restricted in the beginning, but, as the system becomes more popular, more information would be accessible to the public. To conclude, attack feasibility parameters are not static and can change over time.

Parameters used for feasibility calculation are not always easy to rate. Interrelation is the main reason for this issue. Considering the elapsed time, the amount of time needed to perform an attack directly depends on the tools the attacker is using (equipment) and the skill level the attacker has (expertise). On the other hand, expertise itself is closely connected to the equipment and the knowledge about the target of evaluation. When the equipment is not advanced and the attacker's knowledge about the target is considered accessible by public, then the attacker is probably a non-expert. It is also believed that the window of opportunity is a multi-dimensional parameter. Hence, in the HoliSec approach, window of opportunity is divided into the *Access means* and the *Exposure time* [61]. We tried to keep our rating close to the reality by consulting industrial experts on them.

Regarding the weights assigned to the attack potentials, for the sake of simplicity and a better understanding we assumed that all the weights are equal (w=1) in the feasibility formula in Figure 5.1. However, some of the parameters can be considered to have a higher priority compared to the other ones. For instance, even if the equipment is advanced, a layman can not make use of such equipment since he does not have the required knowledge to do so. Thus, expertise should receive a higher priority (lower weight) compared to the equipment.

The bounds and ranges defined in the feasibility table are not strict. The values were assigned based on some discussions and some example scenarios. In other words, these numbers may change over time by consulting more experts involved in the domain. As an instance, to evaluate the higher bound of the "very high" category, the following scenario can be considered: A proficient attacker (expertise=1) using specialized equipment (equipment=1) may easily be able to mount an attack toward an asset in less than a day (elapsed time=0) from a remote location (window of opportunity=0) by using only the information available on the Internet (Knowledge about the TOE=0). Hence, a sum equal to 2 can be categorized under a very high feasibility.

Finally, it should be noted that the attack potentials were classified into four sub-categorised each to comply with the related standards and the current trend in the automotive industry. However, wherever possible, to reach a better accuracy, the parameters can be divided even more to provide fine-grained evaluation. For instance, the elapsed time can be broken down into less than a day, less than a week, one to two weeks, etc.

Use cases

One may argue that aside from confidentiality, integrity and availability, each of which was separately chosen as the root of the tree, there are also other security attributes that can be violated; attributes such as *Authenticity* and *Authorization*. It should be noted that such security aspects, which can also be compromised, can be modeled under either the confidentiality, integrity or the availability of the system. In other words, the reason why an attacker tries to pretend to be someone else or to escalate its privileges, is to illegally access or manipulate data or to make a service out of order. In addition, we mentioned that the technical goals of the attacker were preferred to the moral goals when drawing the tree. This does not question the correctness of the tree, since it impacts neither the attack path nor its corresponding feasibility.

Regarding the use cases themselves, to demonstrate how our nominated methodology works, we have been provided by some use cases on the HoliSec architecture. Since the information about this approach is open to public, not many details about the ECUs, their actual functionalities and other probable entities in the architecture were provided. Consequently, we had to propose some assumptions in order to to be able to discuss the scenarios slightly in more details. However, we tried to maintain these assumptions as close as possible to the real world by consulting industrial experts on how accurate the assumptions are.

In addition, attacks modeled by means of attack trees in each scenario were found during our investigations while reading related literature. Undoubtedly, as new attacks are found occasionally, it can not be claimed that the list of attacks modeled in each use case is thorough. Hence, more types of attacks may be possible in each scenario. Nevertheless, this will not question the correctness of our proposed methodology, since new attack paths can be added to the tree applying the systematic approach provided in previous section and that their feasibility can be calculated using the same parameters.

Eventually, it should be noted that when drawing the attack tree, there may appear some similar sub-trees in the main tree. The reason behind this repetition are some paths that are shared among different attacks. Since avoiding this repetition has not been a part of the thesis, it was not addressed, however, one may avoid existence of such repetitions, to some extent, by rethinking the design of the tree.

Future work

1- The criteria found for the assessing attack trees versus attack graphs have industrial basis. One can assess them from a mathematical point of view. For instance, the complexity of probable algorithms on attack trees and attack graphs.

2- The common criteria used for feasibility analysis where the ones currently used in the automotive industry. One may investigate other criteria, either the ones pointed at by the authors of this thesis that are difficult to be quantified, or any other new criterion.

3- For sake of having simplicity and higher accuracy, exploitation of an attack path was the only factor considered in the definition of the attack potentials. If adequate knowledge regarding the identification is available, one can consider two sets of values, one for the identification and one for the exploitation. In other words, the values assigned to attack potentials for the exploitation phase can be different from the values assigned to them in the identification phase. A linear formula can be used again to calculate the final value of an attack path feasibility.

10 Conclusion

Due to the recent development in the technology and by the advent of the Internet of Things the need for connectivity and automation is rising. The automotive industry has not been excluded from this trend. Autonomous vehicles has raised the need for inter-communication and intra-communication in modern vehicles. While the automotive industry is experiencing this paradigm shift, novel concerns regarding the security challenges of such industrial leap are also emerging.

In this thesis, we investigated existing attack analysis methodologies to find the one that is the most suitable for the automotive world and that can be later used as a part of the risk assessment process concerning modern vehicles. This comparison was done based on some criteria derived from studying related literature and interviewing academic and industrial automotive experts. Next, using this methodology, we explained how attack paths can be modelled and how the feasibility associated with each attack path can be calculated using our proposed formula.

Attack trees, due to their high readability, simplicity, flexibility and fame among the security experts, were proposed as the best-suited methodology for the automotive industry. In addition, by means of two automotive-related use cases, it was explained that how attack trees could be used to model attack scenarios and how the feasibility of each attack path could be assessed in each of the scenarios.

Bibliography

[1] L. Constantin. "Your Car's Computers Might Soon Get Malware Protection". [Online]. Available: `http://www.pcworld.com/article/3053501/security/your-cars-computers-might-soon-get-malware-protection.html`, 2016. Accessed on: Mar 12, 2019.

[2] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno. "Comprehensive Experimental Analyses of Automotive Attack Surfaces". *in 20th USENIX Conference on Security, San Francisco, California*, 2011.

[3] SAE International. *"SAE J3061-201601 Cybersecurity Guidebook for Cyber-physical Vehicle Systems"*. Society of Automotive Engineers (SAE), Jan, 2018.

[4] G. Burzio, G. F. Cordella, M. Colajanni, M. Marchetti, and D. Stabili. "Cybersecurity of Connected Autonomous Vehicles: a ranking based approach". *International Conference of Electrical and Electronic Technologies for Automotive, IEEE*, pages 1–6, July 2018.

[5] G. Dimitrakopoulos and P. Demestichas. "Intelligent Transportation Systems". *IEEE Vehicular Technology Magazine*, 5(1):77–84, March 2010.

[6] P. Kleberger and T. Olovsson. "Protecting vehicles against unauthorised diagnostics sessions using trusted third parties". *in 32Nd International Conference on Computer Safety, Reliability, and Security - Volume 8153*, page 70–81, September 2013.

[7] P. Kleberger, T. Olovsson, and E. Jonsson. "An in-depth analysis of the security of the connected repair shop". *in The Seventh International Conference on Systems and Networks Communications (ICSNC), Proceedings. Lisbon*, pages 99–107, November 2012.

[8] C. Miller and C. Valasek. "Remote Exploitation of an Unaltered Passenger Vehicle". *Black Hat USA*, August 2015.

[9] Andy Greenberg. "Hackers Hijack Big rig Trucks Accelerator and Brakes". [Online]. Available: `https://www.wired.com/2016/08/researchers-hack-big-rig-truck-hijack-accelerator-brakes/`, August 2016. Accessed on: Mar 7, 2019.

[10] McAfee department of cybersecurity. "White paper - Automotive Security Best Practices: Recommendations for security and privacy in the era of the next-generation car", June 2016.

[11] A. Lautenbach, T. Olovsson, and T. Rosenstatter. "A State-of-the-Art Report on Vehicular Security". *The HoliSec Consortium*, Version 1.0, 2017.

[12] A. Lautenbach and M. Islam. "HEAVENS: HEAling Vulnerabilities to ENhance Software Security and Safety". *Vinnova/FFI (Fordonsutveckling/Vehicle Development), Sweden*, March 2016.

[13] T. Casey, P. Koeberl, and C. Vishik. "Defining Threat Agents: Towards a More Complete Threat Analysis". *ISSE 2010 Securing Electronic Business Processes, Springer*, 2011.

[14] ISO. "Road vehicles - Functional safety". (ISO 26262), Geneva, Switzerland, 2011.

[15] C. Phillips and L. P. Swiler. "A graph-based system for network-vulnerability analysis ". *in the Workshop on New Security Paradigms, NSPW '98, New York, NY, USA: ACM*, pages 71–79, 1998.

[16] B. Schneier. "Attack trees". [Online]. Available: `https://www.schneier.com/academic/archives/1999/12/attack_trees.html`, December 1999. Accessed on: Feb 15, 2019.

[17] M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, and E. Weippl. "Dark clouds on the horizon: Using cloud storage as attack vector and online slack space". *USENIX Security Symposium, California, CA, USA*, August 2011.

[18] P. K. Manadhata and J. M. Wing. "An attack surface metric". *IEEE Transactions on Software Engineering*, 37(3):371–386, 2011.

[19] S. Caltagirone, A. Pendergast, and C. Betz. "The diamond model of intrusion analysis". *Center For Cyber Intelligence Analysis and Threat Research Hanover Md*, 2013.

[20] X. Lin, P. Zavarsky, R. Ruhl, and D. Lindskog. "Threat modeling for csrf attacks". *in International Conference on Computational Science and Engineering, IEEE, Vancouver, BC, Canada*, August 2009.

[21] E. M. Hutchins, M. J. Cloppert, and R. M. Amin. "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains". *Leading Issues in Information Warfare & Security Research*, 1(1), 2010.

[22] W. Zeng, M.Khalid, and S. Chowdhury. "A Qualitative comparison of flexray and Ethernet in vehicle network". *in IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE), Canada*, May 2015.

[23] D. K. Nilsson, P. H. Phung, and U. E. Larson. "Vehicle ECU Classification Based on safety-Security Characteristics". *IET Road Transport Information and Control - RTIC 2008 and ITS United Kingdom Members' Conference, Manchester, UK*, May 2008.

[24] F. Kargl, Z. Ma, and E. Schoch. "Security engineering for VANET". *in 4th Workshop on Embedded Security in Cars (escar), Citeseer*, January 2006.

[25] A. Karahasanovic. "Threat modeling of the AUTOSAR standard". *MSc thesis, Department of Computer Science and Engineering, Chalmers University of Technology and University of Gothenburg, Sweden*, November 2016.

[26] AUTOSAR. [Online]. Available: `https://www.autosar.org`, Accessed on: March 20, 2019.

[27] M. Wille. "Automotive Security: An overview of standardization in AUTOSAR". *31. VDI/VW-Gemeinschaftstagung Automotive Security, Wolfsburg*, October 2015.

[28] AUTOSAR. "Utilization of crypto services. AUTOSAR release 4.2.2". [Online]. Available: `http://www.autosar.org/standards/classic-platform/release-42/`, 2016. Accessed on: February 30, 2019.

[29] R. M. Daoud, H. H. Amer, H. M. Elsayed, and Y. Sallez. "Ethernet-based car control network". *Canadian Conference on Electrical and Computer Engineering, IEEE, Ottawa, Ont., Canada*, May 2006.

[30] A. Sawant, L. SVB, and D. Joshi. "CAN, FlexRay, MOST versus Ethernet for vehicular networks". *International Journal of Innovations & Advancement in Computer Science, IJIACS*, 7(4), April 2018.

[31] M. S. U. Alam. "Securing Vehicle Electronic Control Unit (ECU) Communications and Stored Data". *MSc thesis, School of Computing, Queen's University, Kingston, Ontario, Canada*, September 2018.

[32] Y. Huo, W. Tu, Z. Sheng, and V. C.M. Leung. "A survey of in-vehicle communication: Requirements, solutions and opportunity in IoT". *IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, Italy*, December 2015.

[33] C. Sandberg and A.Yadav. "HoliSec: Reference Architecture Holistic Approach to Improve Data Security". *The HoliSec consortium, Sweden's innovation agency (VINNOVA)*, 2018.

[34] W. Stallings and L. Brown. *"Computer Security: Principles and Practice"*. Pearson Education Limited, Upper Saddle River, NJ, USA, 4rd edition, 2018.

[35] A. R. Ruddle et al. "Security requirements for automotive onboard network based on dark-side scenarios. Deliverable D2.3: EVITA. E-safety vehicle intrusion protected applications". *Fraunhofer ISI*, January 2009.

[36] ISO. "Road vehicles - Functional safety - Part 3: Concept phase". (ISO 26262), Geneva, Switzerland, 2011.

[37] ISO. "Road vehicles - Functional safety - Part 4: Product development at the system level". (ISO 26262), Geneva, Switzerland, 2011.

[38] C. Schmittner, Z. Ma, C. Reyes, O. Dillinger, and P. Puschner. "Using SAE J3061 for Automotive Security Requirement Engineering". *in Computer Safety, Reliability, and Security. SAFECOMP 2016, Lecture Notes in Computer Science, Springer, Cham*, vol 9923, 2016.

[39] J. M. Wing and P. Manadhata. "Measuring a System's Attack Surface". *School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Pennsylvania, USA*, January 2004.

[40] C. Miller and C. Valasek. "A Survey of Remote Automotive Attack Surfaces". *Black Hat USA*, August 2014.

[41] T. Zhang, H. Antunes, and S. Aggarwal. "Defending connected vehicles against malware: Challenges and a solution framework". *IEEE Internet of Things Journal*, 1(1):10–21, February 2014.

[42] I. Roufa et al. "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study". *in 19th USENIX Conference on Security, Washington, DC*, 2010.

[43] J. Saarinen. "Students hijack luxury yacht with GPS spoofing". [Online]. Available: `https://www.itnews.com.au/news/students-hijack-luxury-yacht-withgps-spoofing-351659`, July 2013. Accessed on: April 18, 2019.

[44] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl. "Remote Attacks on Automated Vehicles Sensors: Experiments on Camera and LiDAR". *in Black hat Europe, Amsterdam, Netherlands*, 2015.

[45] A. Magar. "State-of-the-Art in Cyber Threat Models and Methodologies". *Sphyrna Security and Bell, Defence Research and Development, Ontario, Canada*, March 2016.

[46] M. Rosenquist. "Defense in depth optimizes security". *White paper - Intel Information Technology, USA*, September 2008.

[47] M. Rosenquist. "Prioritizing information security risks with threat agent risk assessment". *White paper - Intel Information Technology, USA*, December 2009.

[48] OWASP. "CLASP v1.2: comprehensive, lightweight application security process", 2011.

[49] M. Howard and S. Lipner. "The Security Development Lifecycle". *Microsoft Press Redmond*, 8, 2006.

[50] L. Verheyden. "The effectiveness of threat analysis tools." *MSc thesis, Faculty of Economics and Business Administration, GHENT University, Ghent, Belgium*, 2018.

[51] F. Swiderski and W. Snyder. *"Threat modeling (Microsoft Professional)"*. Microsoft Press, 1st edition, July 2004.

[52] S. Hernan, S. Lambert, A. Shostack, and T. Ostwald. "Uncover Security Design Flaws using The STRIDE Approach". *MSDN Magazine*, pages 68–75, January 2006.

[53] K. Wuyts, R. Scandariato, and W. Joosen. "Empirical Evaluation of a Privacy-Focused Threat Modeling Methodology". *Journal of Systems and Software, Elsevier*, 96:122–138, 2014.

[54] R. Khan, K. Mclaughlin, D. Laverty, and S. Sezer. "STRIDE-based threat modeling for cyber-physical systems". *in IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Torino, Italy*, September 2017.

[55] A. Karahasanovic, P. Kleberger, and M. Almgren. "Adapting Threat Modeling Methods for the Automotive Industry". *in 15th ESCAR Conference, Berlin*, 2017.

[56] R. Scandariato, K. Wuyts, and W. Joosen. "A descriptive study of Microsoft's threat modeling technique". *Requirements Engineering, Springer*, 20(2):163–180, 2015.

[57] S. Haque, M. Keffeler, and T. Atkison. "An Evolutionary Approach of Attack Graphs and Attack Trees: A Survey of Attack Modeling". *in The International Conference on Security and Management (SAM)*, pages 224–229, 2017.

[58] C. Alberts, A. Dorofee, J. Stevens, and C. Woody. "Introduction to the OCTAVE Approach". *Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA*, August 2003.

[59] T. Ucedavelez and M. M. Morana. "Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis". *Indianapolis, Indiana, John Wiley & Sons Inc*, 2015.

[60] L. Martin. "Gaining the Advantage: Applying Cyber Kill Chain Methodology to Network Defense". *Lockheed Martin Corporation*, 2015.

[61] C. Sandberg. “HoliSec: Tailoring the HEAVENS risk assessment methodology for improved performance”. *Volvo Group Trucks Technology*, 1.0, 2018.

[62] T. R. Ingoldsby. “Attack tree-based threat risk analysis”. *Amenaza Technologies Limited*, pages 3–9, 2010.

[63] J. B. Hong, D. S. Kim, and T. Takaoka. “Scalable attack representation model using logic reduction techniques”. In *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pages 404–411. IEEE, 2013.

[64] S. Bistarelli, F. Fioravanti, and P. Peretti. “Defense trees for economic evaluation of security investments”. In *First International Conference on Availability, Reliability and Security (ARES'06)*, pages 8–pp. IEEE, 2006.

[65] S.S Priya and PD. S. K. Malarchelvi. “Security Deliberations in Software Development Lifecycle”. *International Conference on Information and Communication Technologies*, 975:8887, 2014.

[66] X. Ou, W. F. Boyer, and M. A. McQueen. “A scalable approach to attack graph generation”. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 336–345. ACM, 2006.

[67] K. Edge, R. Raines, M. Grimaila, R. Baldwin, R. Bennington, and C. Reuter. “The Use of Attack and Protection Trees to Analyze Security for an Online Banking System”. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, Jan 2007.

[68] O. Sheyner and J. Wing. “Tools for generating and analyzing attack graphs”. In *International Symposium on Formal Methods for Components and Objects*, pages 344–371. Springer, 2003.