Computers & Security 133 (2023) 103391

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

STRIDE threat model-based framework for assessing the vulnerabilities of modern vehicles

Zaina Abuabed, Ahmad Alsadeh *, Adel Taweel

Faculty of Engineering and Technology, Birzeit University, P.O. Box 14, Birzrit, Palestine

ARTICLE INFO

Keywords:
Attack tree
Cybersecurity for vehicles
Exploitability
Impact rating
Risk analysis
Risk matrix
Threat modeling

ABSTRACT

Modern automobiles are becoming increasingly sophisticated with enhanced features. Modern car systems have hundreds of millions of lines of code, which increase the attack surface. To address this concern, this paper proposes a new cybersecurity analysis framework that complies with the ISO/SAE 21434:2021 standard. The framework uses the Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privileges (STRIDE) threat model, the Attack Tree Analysis (ATA) approach, and the Common Vulnerability Scoring System (CVSS) as a key score exploitation matrix to rate identified potential threats. To evaluate, the framework was applied, to real-life scenarios, to examine the possible cyber threats in Advanced Driver-Assistance Systems (ADAS). To assess, a tool was implemented to automate threat impact ratings, according to safety, operational, financial, privacy, and legislative metrics. It also automates attack feasibility ratings considering attack vectors, complexity, authentication, and risk level identification based on a five-by-five risk matrix. As a result, 199 potential threats were identified and addressed in, four targeted, ADAS-related use cases. For the Lane-Keeping safety-critical use case, as an example, five security requirements were elicited as countermeasures. These results show that ADAS in modern vehicles are vulnerable to cyberattacks.

1. Introduction

A vehicle system is considered a closed system where no connection exists between the vehicle and the outside world except through the on-board diagnostics (OBD) physical port. Securing vehicle systems was, thus, not a pressing issue. However, this is not the case for modern vehicles. In the last decade, vehicle manufacturers have shifted towards introducing vehicles with highly advanced features that require connecting the vehicle with other external devices physically or remotely. Moreover, many features depend on collecting data from the outside environment using custom media and technologies, such as WiFi, the global positioning system (GPS), camera streaming, radar or light detection and ranging (LiDAR), and many more.

Today's vehicle systems contain more than 100 million lines of code (Charette, 2009). Furthermore, as automotive software is growing at an extensive rate, semi-autonomous vehicles (level 3) will contain more than 300 million lines of code and 500 million lines of code for fully autonomous vehicles (level 5) (SAE International, 2020; Oka, 2021). Vehicle software complexity is consequently increasing along with the increased number of connection media, which expands the attack surface. Security mechanisms will, therefore, rise and it will become necessary for security risks to be identified early and iteratively in the automotive development life cycle to mitigate the risks of potential cyber attacks (Aksu and Aydin, 2022; Ghosh et al., 2023; Luo et al., 2021; Wang et al., 2021).

Several threat analysis risk assessment (TARA) approaches have been introduced for the IT sector, some were adapted to suit the automotive industry, such as STRIDE (Howard and Lipner, 2006), which classifies threats into six categories: spoofing, tempering, repudiation, information disclosure, denial of service, and elevation of privileges. Other TARA approaches were designed based on published vehicle road safety and security standards for automotive use. From a software perspective, threat modeling is a strategic process followed to identify exploitable vulnerabilities and probable attack scenarios within the software under evaluation (UcedaVelez and Morana, 2015). Historically, threat modeling was first used in military armies as an analytical approach to determine how enemies could execute effective strikes. This kind of analysis included determining attack scenarios, the attacker's capabilities and motivation, and attack likelihood. Thus, armies can build strategies to mitigate these potential attacks (Kuehn, 2017; Tzu, 2017).

* Corresponding author.
E-mail addresses: zaina.shtaiwi.zs@gmail.com (Z. Abuabed), asadeh@birzeit.edu (A. Alsadeh), ataweel@birzeit.edu (A. Taweel).

https://doi.org/10.1016/j.cose.2023.103391
Received 12 April 2023; Received in revised form 14 July 2023; Accepted 17 July 2023
Available online 22 July 2023

On the other hand, risk assessment focuses on determining the level of risk by utilizing pre-identified metrics in order to manage and prioritize risks.

Contributions. We propose a systematic TARA framework based on the workflow of the ISO/SAE 21434:2021 standard (ISO/SEA, 2021). STRIDE threat model is used to classify the addressed threats, and damage impact is rated according to safety, operational, financial, privacy, and legislative metrics. Accordingly, attack scenarios are formulated relying on the Attack Tree Analysis (ATA) approach. CVSS v2.0 key score exploitation matrix is used to rate the feasibility of each attack, considering attack vectors, attack complexity, and authentication metrics. Lastly, a five-by-five risk matrix is designed to assess an attack risk level (LOW, MEDIUM, HIGH, or EXTREME). The output of the proposed framework is a set of security requirements that should be considered while developing the vehicle software. However, risk analysis should be conducted iteratively, since new threats are likely to emerge due to continuous updates from the field. Additionally, the implemented countermeasures from previous risk analysis iterations may contain new unaddressed threats.

The applicability and usability of the developed framework are demonstrated using, real-life, scenarios of the Advanced Driver-Assistance Systems (ADAS) that contain safety-critical functionalities, which are used as running examples. The framework was applied to four use cases, which are: the Navigation system, Lane-Keeping system, Adaptive Cruise Control (ACC) system, and Anti-lock Braking System (ABS). As a result, in applying the proposed framework workflow and metrics, 199 threats were identified and addressed for the above four use cases, which were targeted in this study. In the Lane-Keeping use case, for example, ten potential threats were identified and addressed, which were used to formulate three different attack scenarios. In the first round of risk analysis, five security requirements were elicited to countermeasure the attacks, which are relatively considerable for a simple yet safety-critical use case. These results show that modern automobiles are, therefore, susceptible to possible security threats that an attacker may take advantage of. This, hence, calls for the need to secure vehicle internal and external communication by applying suitable risk mitigation countermeasures.

A threat analysis tool of the proposed framework was implemented, based on the Microsoft Threat modeling Tool (MS TMT), to generate potential threats. The tool automates manual steps in the framework that are related to the impact and attack feasibility metrics ratings and risk level identification for the generated threats. The tool also automates attack risk level identification based on the designed risk matrix. The tool enables a non-expert analyst to use the generated results to assess different scenarios by modifying respective metrics values in proportion to threat situations.

2. Related work

E-safety vehicle intrusion-protected applications (EVITA) are part of a research project funded by the European Commission (Henniger et al., 2009). In EVITA, TARA was used in order to elicit and prioritize security requirements for designing a secure, cost-effective architecture for automotive on-board systems. In EVITA, attack trees were adopted as the main approach for the threat analysis phase. Then, all attack trees were restructured to identify all attack patterns in the risk analysis phase. Subsequently, the risk assessment is performed according to the severity of the risk outcomes, the probability of attacks, and the controllability.

Healing Vulnerabilities to Enhance Software Security and Safety (HEAVENS) proposes a framework for TARA in order to elicit security requirements for automotive software (Islam et al., 2016). Many automotive organizations adopted HEAVENS since it was explicitly recommended in SAE J3061 (SAE International, 2016). The framework starts with defining the system by identifying the details of the components and the communication of the architecture. Then, threat analysis is conducted using STRIDE threat modeling (Howard and Lipner, 2006). The next phase is risk assessment, which aims to estimate the security level depending on the threat level and impact level values. The second version of HEAVENS is released in 2021 (Lautenbach et al., 2021) which contains 17 updates to make old HEAVENS meet the new ISO/SAE 21434:2021 standard.

Operationally critical threat, asset, and vulnerability evaluation (OCTAVE) is a systematic context-driven risk evaluation approach (Alberts and Dorofee, 2001). It employs workshops to conduct discussions within the analysis team members in order to exchange information about system assets, threats, vulnerabilities, strategies, and countermeasures. The output is a set of risk mitigation plans that should be implemented. Managers should use the approach results to implement a review plan that must be executed continuously to ensure the validity of the security status.

Threat vulnerability risk analysis (TVRA) is a quantitative approach that aims to identify the risk by quantifying the likelihood and impact of each attack that exploits a system vulnerability (Rossebo et al., 2007). TVRA starts by identifying the system objectives and then determines the system assets. For each asset, vulnerabilities are identified, consequently identifying threats and their associated incidence. Then, the likelihood and impact of each threat are assessed according to the ISO/IEC 15408 standard. Finally, the risk is defined, and a suitable countermeasure is determined.

STRIDE threat model was officially released by Microsoft in 2002 (Howard and Lipner, 2006). The goal of STRIDE is to identify various types of threats based on the design of the software. For this purpose, threats are classified into six categories: identity spoofing, data tampering, repudiation, information disclosure, denial of service, and elevation of privilege. Threats can be identified by building a data flow diagram (DFD) which determines the system entities, trust boundaries, communication types, events, etc. The more accurate the DFD, the more successful threats will be extracted. As a result, countermeasures to mitigate the addressed vulnerabilities should be implemented.

Attack tree analysis (ATA) is a tree-based approach that aims to identify potential attacking strategies that an attacker may perform against the system under evaluation (Ren et al., 2011). Before starting, the system should be clearly described in detail in terms of components, communication, and assumptions. Then, tree construction is started by identifying the top event, which represents the attack goal. The logical relationships (AND, OR) between events will determine how events are linked together. For risk assessment, three metrics are estimated for each leaf node: attack cost, technical difficulty, and the probability of being discovered. Finally, attack scenarios are established and documented. Ebrahimi et al. utilize ATA to design a threat modeling approach to construct attack paths for connected vehicles based on an identified attack surface (Ebrahimi et al., 2022). The proposed model is evaluated using a testing platform for autonomous driving functions called SPIDER (GmbH, 2022).

The security automotive risk analysis (SARA) method is an attacker-based framework designed to address new threats that are generated from the adoption of new technologies (Monteuuis et al., 2018). These threats include malicious observer threats and altered road infrastructure threats. STRIDELC is used, which extends the traditional STRIDE threat model by adding two threat categories, which are linkability and confusion. Additionally, attack capability is proposed as a metric to measure the strength of an attack. The outcome is applying countermeasures to minimize the risk level, and the framework is repeated until the risk level reaches an acceptable level.

Security-aware hazard and risk analysis (SAHARA) combines the STRIDE threat model with the hazard analysis risk assessment (HARA) approach to identify security threats to safety goals (Macher et al., 2015). Thus, automotive safety integrity levels (ASILs) analysis is the main concept that SAHARA relies on. Threats are estimated according

to three metrics: needed resources (R), required know-how (K), and the criticality of the threat (T). Then, the security level ($SecL$) is determined using a determination matrix approach. At the end, the $SecL$ is converted into the automotive safety integrity level (ASIL) which is identified by ISO 26262 as having four levels (A, B, C, and D) (ISO, 2011). Process for attack simulation and threat analysis (PASTA) is a risk-based framework that consists of seven stages, and each stage contains multiple activities (UcedaVelez and Morana, 2015). PASTA uses a variety of tools and techniques within its activities, such as abstract architectural diagrams, data flow diagrams (DFDs), and attack trees. Fortunately, a rich document is provided that contains a detailed explanation for how to apply each activity in each stage. The output of PASTA is the establishment of mitigation strategies and countermeasures.

Common vulnerability scoring system (CVSS) and automotive safety integrity levels (ASIL) approaches also use STRIDE threat modeling (Wolf, 2019). It combines security and safety by using CVSS ratings and ASIL safety ratings for the purpose of penetration testing. TARA+ is a novel threat model, that proposes to perform cybersecurity analysis for automated driving systems (Bolovinou et al., 2019). The model is verified following the SAE J3016 standard by applying it to highway, highway traffic jams, and urban AD functions. The main addition in TARA+ is considering the controllability of the attack as a factor while calculating the threat severity. Kong et al. propose a risk assessment framework for smart cars (Kong et al., 2018). It relies on the IT security management standard GMITS (ISO:IEC 13335) (ISO/IEC, 2004), and the attack tree analysis technique to assess the addressed threats by calculating the vulnerability level. The framework was demonstrated for vehicle velocity increases and personal information leakage in smart cars using the attack tree approach.

Autonomous vehicles require cybersecurity risk analysis to identify potential threats and vulnerabilities. This includes analyzing the system architecture, identifying attack points, and evaluating their likelihood and impact. A number of recent studies have focused on these issues. Kim et al. (2021) analyzed 151 studies from 2008 to 2019 on autonomous vehicle attacks and defenses, categorizing them into control systems, drive system components, and vehicle-to-everything communications. The defenses are categorized as security engineering, intrusion detection, and anomaly detection. In another study, Khan et al. (2022) analyzed attacks and defenses against autonomous vehicles, with an emphasis on vehicle-to-everything connectivity technology. Developed a system dynamics conceptual model for analyzing cybersecurity in deployments of Connected and Autonomous Vehicles (CAVs). The model integrates six critical approaches, including communication framework, secure access, human factors, CAV penetration, regulatory laws, policy framework, and trust, promoting an improved, self-regulatory, and resilient CAV system. Algarni and Thayananthan (2023) proposed an approach that uses intelligent cybersecurity options to protect all services used in automated vehicles from underlying threats in the event of cyber attacks. Benyahya et al. (2022) focused on security, data protection, and standards while analyzing cybersecurity threats and data privacy related to Automated City Shuttles (ACS) integration in smart cities. In addition to identifying cyber attacks and defining mitigation methods, it also highlights the importance of cyber security laws and data privacy solutions.

3. The proposed framework

The ISO/SAE 21434:2021 standard, which provides a road map for TARA framework designers, is adopted in this study. The work of Plappert et al. outlines the general workflow of the ISO/SAE 21434-based TARA approach (Plappert et al., 2021). It recommends to use risk analysis methods, best practices, and metrics in order to yield the best results. Accordingly, we propose our systematic TARA framework, which consists of five major steps: use case analysis, threat analysis, attack analysis, risk assessment, and risk treatment. Fig. 1 depicts the general workflow of our framework and the activities that correspond to each phase.

3.1. Use case analysis

Use case analysis contains two main activities: Assets Identification and Attackers identification. The main goal of this phase is to identify all possible entities and interactions in the system (use case) under evaluation that may lead to a risk. For this reason, a high-level design of the use case along with design assumptions is needed as a prerequisite before starting the use case analysis. The high-level design is a comprehensive description of the system components that may include a preliminary architecture. It should describe system entities, interactions, protocols, data, communication means, etc. However, if TARA is applied very early in the concept phase, the focus will be on identifying generic assets and the associated risks of attacks.

3.1.1. Assets identification

In asset identification, the high-level design should be analyzed to determine if an item can be considered an asset, such as sensors, actuators, firmware stores, communication components inside the vehicle, vehicle functions, algorithms, etc. Besides, all the necessary details of the assets should be provided at this step. However, unnecessary details can be misleading, so the analyzer should be aware of the level of abstraction that should be used, which depends on the time at which TARA is conducted in SDLC. Next, based on asset identification and use case assumptions, a DFD should be drawn as accurately as possible because if the DFD is wrong, the threat analysis will be wrong. DFD elements are: process, data store, data flow, entity, and trust boundary.

3.1.2. Attackers identification

The next activity is attacker identification, which aims to classify probable attackers based on their motivation, the skills they possess, the available resources, and the access media they use. Experts' opinions and previous attacks' literature on similar use cases provide a good source to complete this activity. Correct attacker' identification can be considered complementary to DFD to identify successful threats and damage scenarios, as we will describe later in the next section.

3.2. Threat analysis

Threat analysis is the most important phase that should be conducted with the aid of threat analysis tools and a collaborative effort with domain experts. The output of this analysis is to formulate potential threat scenarios and estimate the impact of damage scenarios on cybersecurity properties. Threat analysis consists of two main activities: threat identification and impact rating.

3.2.1. Threat identification

After analyzing the use case under evaluation, the STRIDE threat modeling method is used to identify and classify potential threats. STRIDE utilizes the previously prepared DFD diagram to determine threats for each asset. Microsoft determines the threat types associated with each element type in DFD (Howard and Lipner, 2006). Accordingly, potential threat scenarios are formulated and prepared to be evaluated in the attack analysis phase. Also, a damage scenario is formulated, which represents the result of exploiting a potential threat by an attacker. It is recommended to formulate damage scenarios collaboratively with domain experts.

3.2.2. Impact rating

The main purpose of this activity is to rank the impact of a damage scenario against predefined cybersecurity properties. In general, the impact metrics capture the effects of successful vulnerability exploitation on a critical asset. ISO/SAE 21434:2021 standard recommends using four parameters related to the security objectives that are used in the

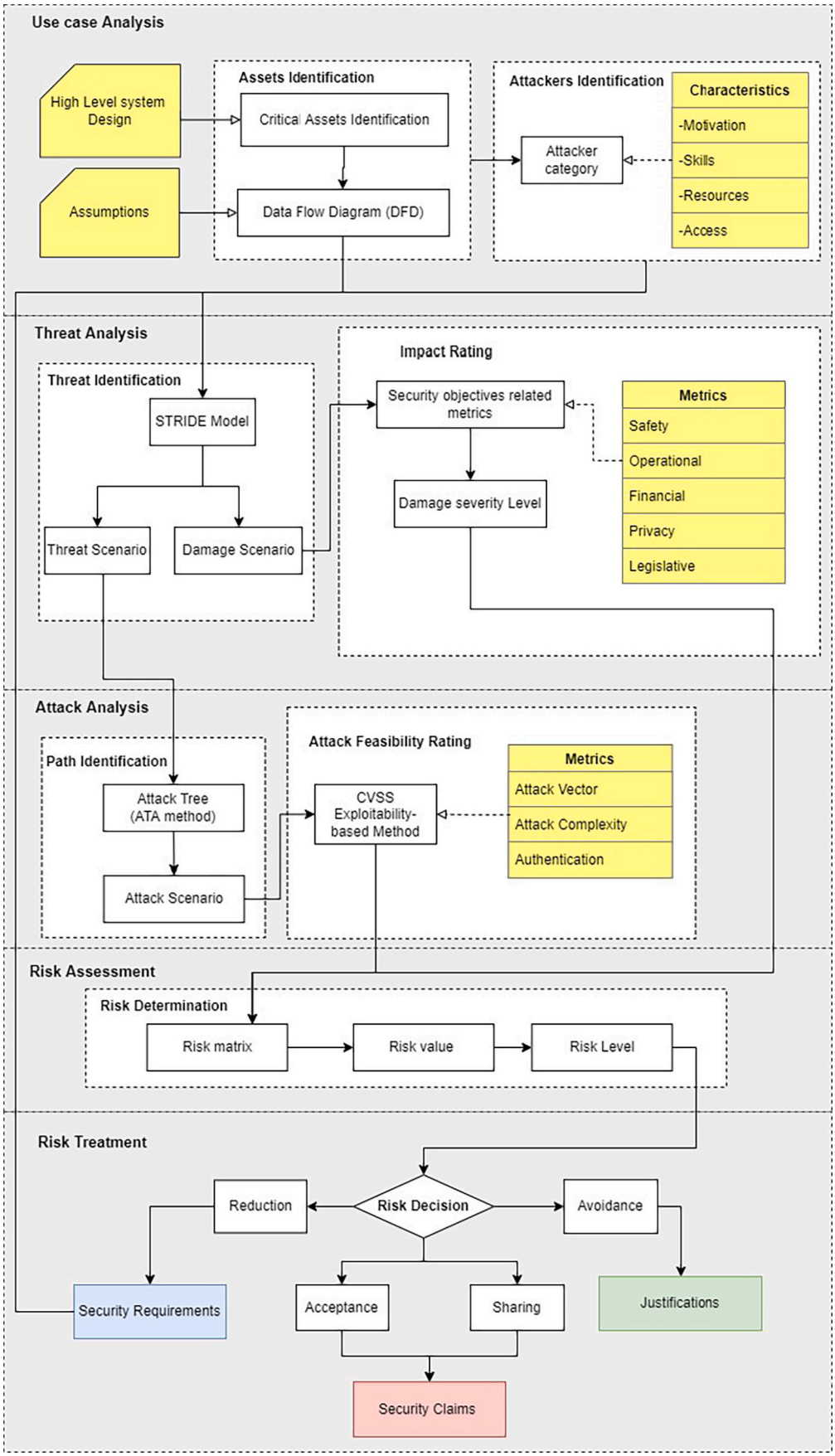

Fig. 1. ISO/SAE 21434:2021 compatible framework.

Table 1
Mapping impact score to impact value (Islam et al., 2016).

Impact Score (IS)	Impact Level (IL)	Impact Value (IV)
0	None	1
1-19	Low	2
20-99	Medium	3
100-999	High	4
Greater than 1000	Critical	5

HEAVENS framework, which are safety (i_s), financial (i_f), operational (i_p), privacy (i_p) and legislative (i_l). These parameters are described in HEAVENS (Islam et al., 2016).

In HEAVENS, privacy and legislation are considered as one parameter. However, HEAVENS 2.0 (Lautenbach et al., 2021) split them into two separate parameters since each has a different meaning. Parameter values can be estimated to be None, Low, Medium, and High, and each level is translated into a qualitative value of 0, 1, 10, or 100, respectively. The estimation depends on the damage impact degree that corresponds to the security objective.

After damage impact parameter estimation, equation (1) is used to calculate the overall impact score (IS) (Islam et al., 2016) (Lautenbach et al., 2021).

$$IS = w_s i_s + w_f i_f + w_o i_o + w_p i_p + w_l i_l \tag{1}$$

The weight value (w_x) is a number between 1 and 10 that represents the importance of the parameter relative to the other parameters. As a suggestion, safety and financial are considered the most important for vehicular systems; therefore, the weight for them will be assigned to 10 as an example (Islam et al., 2016). While the remaining weights can be assigned to 1.

$$w_s = w_f = 10 \tag{2}$$

$$w_o = w_p = w_l = 1 \tag{3}$$

Then, the damage impact severity level can be determined by mapping the calculated impact score (IS) from equation (1) into its associated impact level (IL) and impact value (IV), as Table 1 describes.

3.3. Attack analysis

In attack analysis, potential attack scenarios are identified based on threat scenarios identified in the analysis. The attack tree method is adopted to identify attack paths, and then each attack is rated against its feasibility. Attack analysis consists of two main activities: attack path identification and attack feasibility rating, which are described next.

3.3.1. Path identification

The main purpose of this activity is to identify potential attack scenarios. ATA method (Ren et al., 2011) is the most suitable approach for path analysis. Based on the threat scenario, the goal of the attacker is identified, which represents the root node in the tree. Then, the events that lead to the goal are identified and arranged in a hierarchical tree (parent-child). The leaf node is an event that cannot be divided and represents the starting point from which an attacker can start his attack. To get realistic attack scenarios, attack trees must be constructed carefully and as accurately as possible. Next, attack scenarios are formulated by reading the attack tree from leaf to root.

3.3.2. Attack feasibility rating

It is important to know that one threat scenario can yield many attack scenarios, so an attack feasibility rating is essential to evaluating each attack scenario. Therefore, we adopt CVSS v2.0 as it is recommended by ISO/SAE 21434 for attack feasibility rating. The base score exploitability matrix for v2.0 is described publicly in a published guidebook (Mell et al., 2007). Attack feasibility rating also needs collaborative effort with automotive domain experts to rate each attack scenario according to CVSS base impact metrics. When rating is finished, the exploitability value can be calculated using equation (4) which is proposed by CVSS v2.0 (Mell et al., 2007).

Table 2
Mapping score to severity level according to CVSS v2.0 - adapted.

Score Range	Severity
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.4
Critical	8.5-10.0

$$Exploitability = 20 * AV * AC * AU \tag{4}$$

where the access vector (AV) indicates how the vulnerability can be exploited. AC stands for access complexity, which measures the degree of complexity of vulnerability exploitation after being accessed. Authentication (AU) measures how many times an attacker must authenticate to exploit a vulnerability. The calculated exploitability score is a number between 0 and 10. After that, the exploitability severity level can be determined by mapping the exploitability value to its associated level, as described in Table 2. To provide more consistency between the impact rating and feasibility rating, we added the critical severity level to the original levels that are provided by the CVSS v2.0 guidebook.

3.4. Risk assessment

In this phase, risk determination is performed using the risk matrix approach. A semi-quantitative 5-by-5 risk matrix is used to categorize the impact and attack feasibility to quantify the qualitative ratings. Before starting the construction of the matrix, we give each cell a name (row, column). Rows are numbered from 1 to 5, and columns are also numbered from 1 to 5. For the matrix construction, first we assign the values 1, 2, 3, 4, and 5 to the rows from top to bottom, respectively, and the same values from left to right. These values represent the severity levels of none, low, medium, high, and critical, respectively. Secondly, a score for each cell is calculated by multiplying the row with the column values (Ni et al., 2010). Finally, cells are categorized into four categories: low, medium, high, and extreme. Categorization is based on cell score, where values less than 4 are considered low, values from 4 to 8 are considered medium, values from 9 to 16 are considered high, and values greater than 16 are considered extreme. These categories are assigned based on subjective judgement that seems relatively suitable in the context of automotive systems. The resultant matrix is shown in Fig. 2.

3.5. Risk treatment

Once the risk level is determined, a risk treatment method must be selected to manage this risk. In general, there are four known risk management methods available and recommended by ISO/SAE 21434: risk acceptance, risk sharing or transfer, risk avoidance, and risk reduction. Risk acceptance means that the risk can be managed without additional countermeasures. For example, if there is a built-in strategy in the architecture that solves this risk, then by default it will be mitigated. Or, if the risk has no impact or its exploitability is limited, then the risk can be accepted. Risk sharing or transfer means that the risk will be shifted to a third party to deal with, especially if the risk cannot be managed internally. For example, if the risk is a financial one, a contract with an insurance company can mitigate this risk. However, these types of risks are rare in the automotive industry because of the strong relationship between risk and safety. For both risk acceptance and risk sharing, cybersecurity claims must be documented.

On the contrary, risk avoidance is required to eliminate the source of risk since no available countermeasure can mitigate it. If the risk is

Table 3
ADAS identified assets.

Sensors	In-vehicle modules	Data stores	External data sources
Wheel speed	Human-machine interface (HMI)	Local maps data store	Environmental data
Radar/LiDar	On-board diagnostics (OBDII)	Engine control unit (ECU)	Mechanic
Cameras	Telematic control unit (TCU)		Driver
Brake pedal	Engine		Maps update server
	Electronic braking system (EBS)		
	Adaptive cruise control (ACC)		
	Lane-keeping		
	Brake actuators		

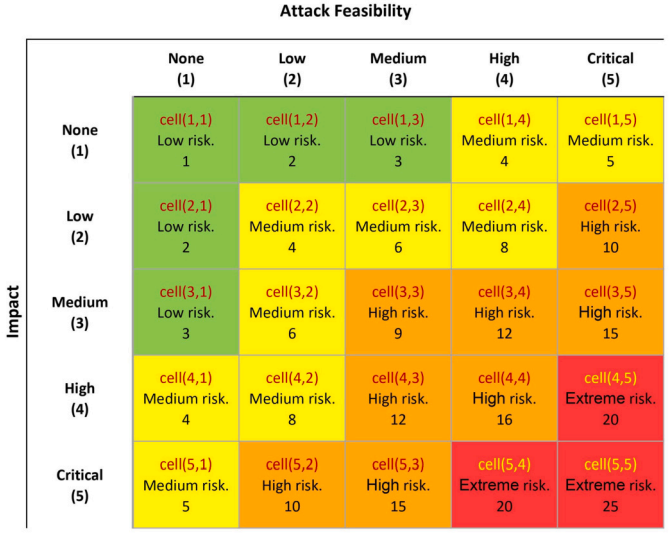

Fig. 2. 5 by 5 risk matrix.

associated with a required feature, risk avoidance may lead to the abandonment of the feature. So, this decision must be taken carefully. Risk reduction means that a countermeasure to mitigate this risk must be developed. Consequently, if risk reduction is the selected decision, then the associated security requirement must be documented. After that, risk analysis should be repeated to investigate if the new requirements may expose the system to new threats.

4. Applying the framework for ADAS

To verify the applicability of the framework presented in Section 3, advanced driver assistance system (ADAS) is selected to be analyzed considering cybersecurity risks.

4.1. Use case analysis for ADAS

ADAS is decomposed into four main ADAS applications (Winner et al., 2018): (1) navigation systems, (2) adaptive cruise control (ACC). (3) Lane-keeping and blind spot systems (4) anti-lock braking systems (ABS). These four applications are used for our framework evaluation. However, more applications are defined by Synopsys for ADAS (Synopsys, 2022).

Some domain experts from Ford describe the ACC and the navigation applications (Group et al., 2005) as follows: The ACC depends on the GPS data, which comes from the GPS receiver that is managed by the navigation application, the local map data, and the radar and LiDAR data to predict the upcoming events within about two kilometers. Based on these predictions, the ACC adjusts the speed and acceleration of the vehicle. In order to be up-to-date, the map data is updated with the over-the-Air (OTA) updates feature, which is controlled via the telematic module. Lane-Keeping is the capability to keep the vehicle within the borders of the road using a vision camera sensor. This application is considered one of the most difficult features within ADAS because the road lines have different colors and shapes. Moreover, the vision differs between day and night. The worst is the absence of road lanes in some cases (Kukkala et al., 2018). Canny based on the Otsu algorithm, is one of the newest methods to detect road lanes. It depends on pre-processing the image's edges and can detect road lanes efficiently in daytime or nighttime (Li et al., 2016). Another feature is ABS, which is used in almost all modern vehicles to provide safety against sudden accidents by preventing tires from locking. It depends on speed sensors that are connected to the wheels to control the wheel slip ratio (Algadah and Alaboodi, 2019).

In the work of Miller and Valasek (2014), 24 different car model architectures have been investigated. As a result, the authors noticed that cars from the same region have almost the same architecture. Consequently, three architectures are highlighted: European, American, and Asian architectures. The work of Winsen investigates these architectures and draws general high-level interconnection diagrams for each (Winsen, 2017). We select the European vehicular architecture to apply the use-case analysis to.

4.1.1. Assets identification for ADAS

Depending on the system description, we can identify the use-case assets. These assets are classified into four groups: sensors, in-vehicle modules, data stores, and external data sources. Table 3 describes the classified ADAS assets. After identifying use case assets, we can draw the DFD diagram accordingly.

MS TMT 2016 is used to draw the DFD. The assumed information flow between the identified assets was reflected to draw a complete DFD. To customize the tool, we use the template for automotive that was designed by NCC-group (Nccgroup, 2016). It created the first automotive template in 2016. It is worthy of mention that the template was used in many works in the literature and was strongly recommended for research purposes. Additionally, the template can be easily adapted to suit any vehicle use case by adding the additional modules and assigning their attributes. Moreover, if there is any additional bus or technology, it can be added to the template with its attributes and associated threats.

However, we noticed that there are some missing components, such as the engine and brake modules. The NCC group template was modified by adding the missing modules and flows (Wolf, 2019). Therefore, we decided to use the modified template. While drawing the DFD, it is important to consider that each ECU needs to be connected to the firmware data store in order to import the basic settings and for recovery purposes. Moreover, each sensor gets its information from the external environment, so an environment boundary is needed to separate the car system from the environment. Fig. 3 depicts a full DFD for the applications under evaluation.

4.2. Threat analysis for ADAS

After finishing the DFD diagram, a threat report has been generated from MS TMT as an HTML file. It describes the potential threats for each ECU module through the connected flows. In the threat analysis phase,

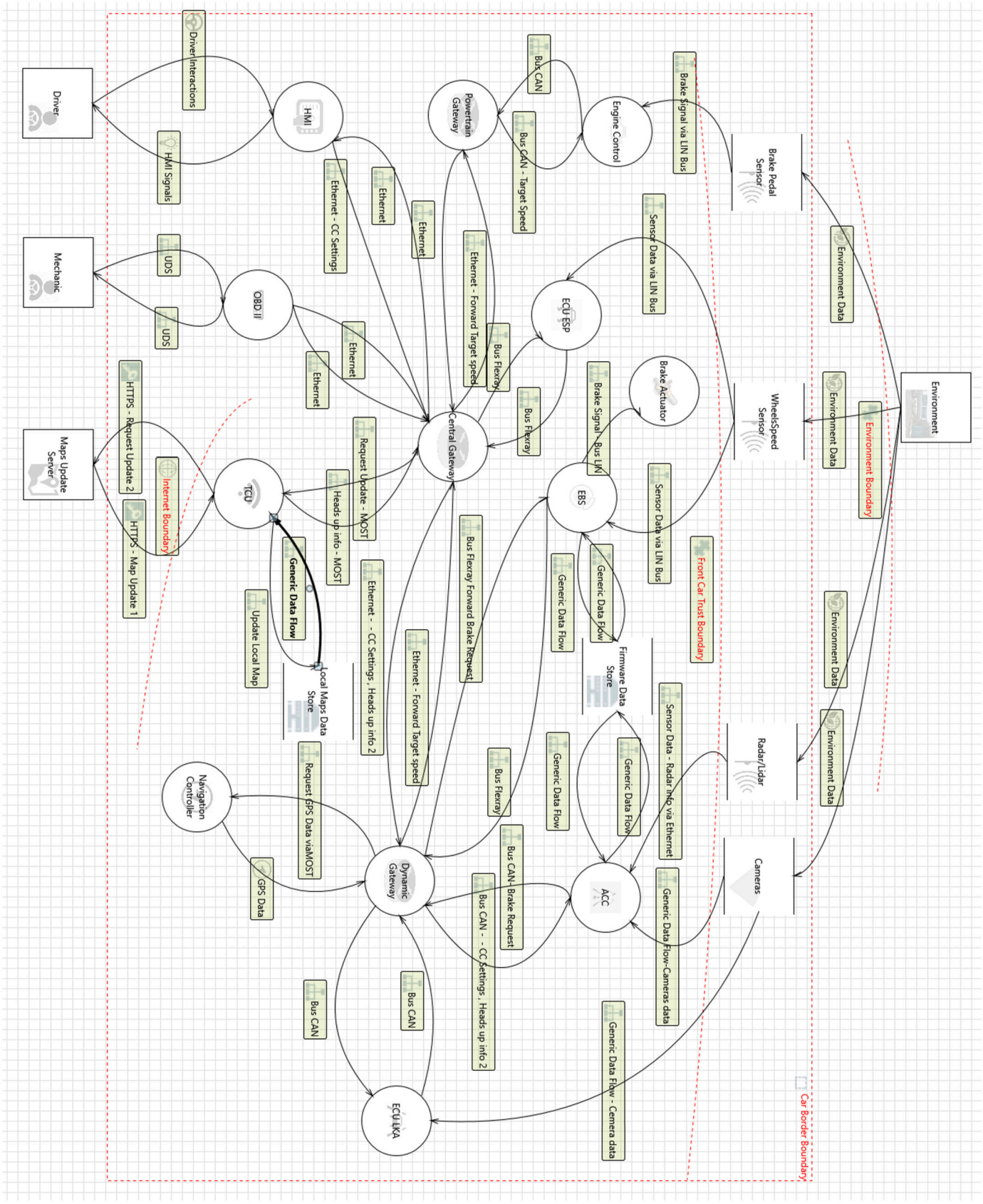

Fig. 3. The proposed DFD for ADAS applications.

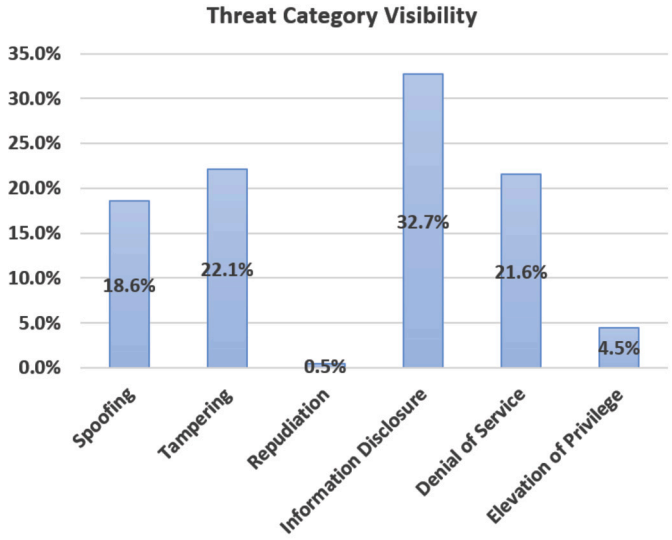

Fig. 4. Threats visibility of STRIDE categories in ADAS.

Table 4
Impact rating for Lane-Keeping ECU.

Threat id	Impact Metrics					Impact Score	Impact Level	Impact Value
	is	if	io	ip	il			
4	1	0	1	0	0	11	Low	2
213	10	1	1	0	1	112	High	4
214	1	0	0	1	1	12	Low	2
215	100	100	10	0	1	2011	Critical	5
241	100	100	10	0	10	2020	Critical	5
242	1	1	10	0	10	40	Medium	3
243	10	1	1	100	10	221	High	4
244	0	10	0	0	0	0	None	1
245	0	10	0	100	10	210	High	4
246	10	10	0	0	1	201	High	4

the risk analyst will depend on the generated report to perform threat identification and impact rating activities.

4.2.1. Threat identification for ADAS

The threats generated by MS TMT are classified based on the STRIDE approach. For each threat, the tool provides a description, category, attack method, source, target, and recommendations to mitigate the threat. In this step, 212 threats are identified by the tool. Next, the report is exported to an Excel sheet file to be filtered in order to remove any redundancy and inapplicable threats

For analysis, threats that relate to the same target and interaction are grouped. Then, we check the threat STRIDE categories that exist for each target. The generated report addresses 22 different targets; among them, three are not considered real targets in the realistic automotive industry. The driver, mechanic, and map-update server could not be targets for a serious attack because the driver and the mechanic are outside the boundary of the system and there is no software component. Also, the map-update server is outside the boundary, and its protection is not the responsibility of the vehicle system. So, the threats that are associated with these targets (13 threats) are considered inapplicable threats. Consequently, 199 threats were left after neglecting the inapplicable ones. Fig. 4 depicts the visibility of the generated STRIDE threat categories.

4.2.2. Impact rating for ADAS

Table 4 depicts a sample of damage impact rating results associated with the Lane-Keeping module. However, this rating is approximate and could contain inaccurate values, but the purpose of this step is to estimate how much time and effort are needed to complete the rating process. Based on the estimated impact score, the impact level and values are calculated in order to prepare for the risk matrix in the risk assessment phase.

4.3. Attack analysis for ADAS

In the Attack analysis phase, attack path identification is performed for the Lane-Keeping use case following the ATA approach. Then, attack feasibility is rated for each potential attack. Lane-Keeping use-case functionalities depend basically on cameras that are lying on the vehicle's body.

4.3.1. Attack path identification for ADAS

For Lane-Keeping use case, the asset under evaluation is the captured images by the cameras. Some of camera's attacks are described in the work of Hamad and Prevelakis (2020). Based on that, we identified two attackers' goals: Goal 1 is manipulating the stored images, and Goal 2 is disclosing the stored images. For Goal 1, there are three potential attacks. First, AT1 is confusing camera functionality by physical interference by a thief attacker to yield illegal money.

Second, AT2 is modifying images by writing the wrong bits. Third, AT3 is manipulating the image processing algorithm that operates within the Lane-Keeping ECU. AT2 and AT3 may be of interest to a researcher for research and testing purposes. Considering the STRIDE classification, Goal 1 attacks belong to the tempering and spoofing categories and affect the data integrity security property.

For Goal 2, there are two potential attacks. First, AT4 extracts the stored images and sends them to a remote location. This attack may be of interest to external organizations for tracking and spying purposes. AT3, as previously described, is also relevant to Goal 2. Considering the STRIDE classification, Goal 2 belongs to the information disclosure category and affects the data confidentiality property. Based on that, we draw an attack tree as depicted in Fig. 5. Finally, three potential attack scenarios for ADAS-Lane-Keeping use case are formulated as listed below:

- Scenario 1: A thief or criminal may try to confuse the proper operation of the camera with physical interference, such as covering the camera with tape, shining intense light towards the camera, or replacing the camera in order to yield illegal money. This may cause a potential accident or prevent the owner from locating his vehicle.
- Scenario 2: Automotive researchers or penetration testers may try to manipulate captured images to cause a wrong environment perception. This attack requires controlling the ECU where the images are maintained and processed via an image processing algorithm.
- Scenario 3: A competing organization may target the recorded data from the camera for spying or tracking purposes. The data may contain sensitive information about other vehicles or the car's surroundings. This involves manipulating the Lane-Keeping ECU to access the storage and send the data to a remote destination using the Telematics module or fake device.

4.3.2. Attack feasibility rating for ADAS

To reduce analysis time, we consider the Lane-Keeping application from ADAS while building attack trees and formulating attack scenarios. We rate attack feasibility according to the CVSS 2.0 exploitability metric. The attack feasibility value is calculated based on the exploitability equation (4). Table 5 describes the results of this activity. Finally, the attack feasibility level is determined based on Table 2.

4.4. Risk assessment for ADAS

The risk level is calculated for each potential Lane-Keeping threat using the risk matrix approach. For example, for threat ID 4, the impact value equals 3, and the attack feasibility value equals 2. Consequently,

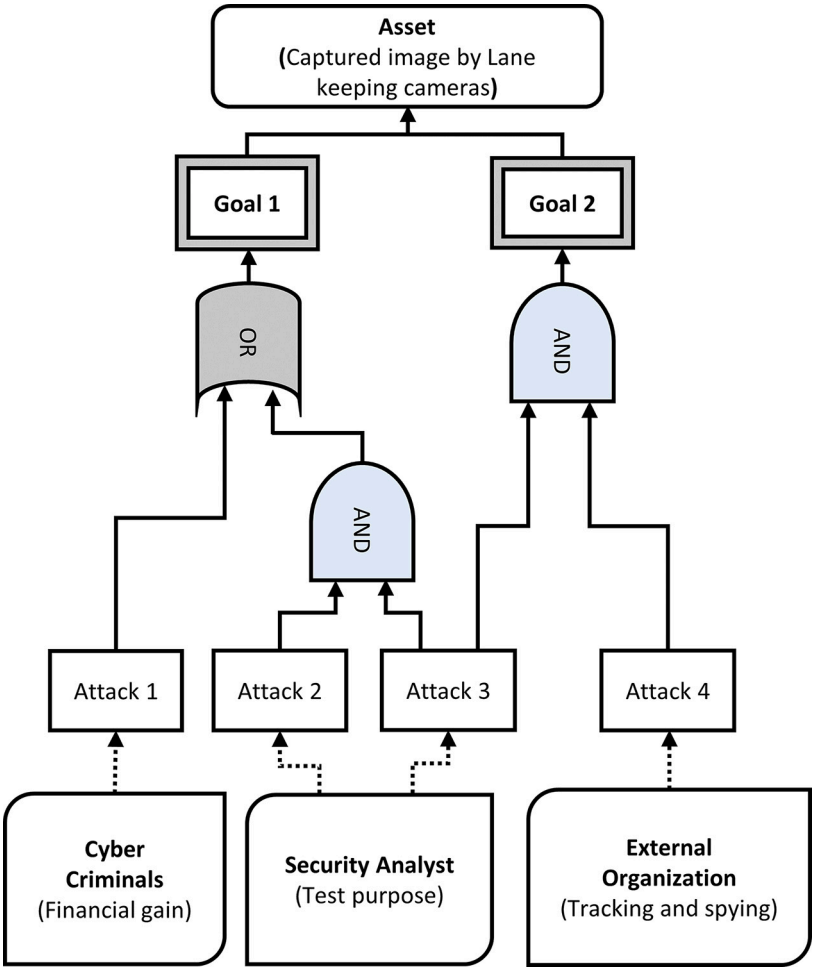

Fig. 5. Lane-keeping application attack tree.

Table 5
Lane-Keeping attacks feasibility rating.

	Id	Title	Attack method	Source	AV		AC		Au		Attack feasibility Score	Attack feasibility level	Attack feasibility value
Goal 2 - AT4	243	Data flow sniffing	Man-in-the-middle using attack using hardware or software.	Dynamic Gateway	N	1	L	0.71	N	0.704	9.9968	Critical	5
	244	Other connected parts of a Gateway may acquire more information than allowed	Misbehavior of the Gateway. Or connecting to one and faking a device.	Dynamic Gateway	A	0.646	H	0.35	S	0.56	2.53232	Low	2
	245	CAN Bus reveals all messages to everyone on the Bus	Attach a device to the CAN Bus, and sniff all messages.	Dynamic Gateway	N	1	L	0.71	N	0.704	9.9968	Critical	5
Goal 1-AT1	213	Data send over Generic Data Flow - Camera data can be tampered with	Gain access either to the physical network (lines) or control over one device connected to it.	Cameras	N	1	M	0.61	S	0.56	6.832	Medium	3
	4	Data send over Environment Data can be tampered with	Gain access either to the physical network (lines) or control over one device connected to it.	Environment	N	1	M	0.61	S	0.56	6.832	Medium	3
Goal 1-AT2 AT3	241	Claiming to be a ECU LKA and therefore controlling the traffic	Attach to the network, copy the address of the physical layer of your target, disconnect it and replace your device with it.	Dynamic Gateway	L	0.395	L	0.71	N	0.704	3.948736	Low	2

Note: (1) Id, Title, Attack Method, Source are generated by MS TMT 2016. (2) AV, AC, and AU are estimated by domain experts. (3) Score and level are calculated based on CVSS guidebook. (4) Level is assigned to be used in the risk matrix.

Table 6
Risk assessment for Lane-Keeping risks.

Threat id	Impact value	Attack feasibility value	Matrix cell location	Risk value	Risk level
	(row r)	(column c)	(r , c)		
4	3	2	(3 , 2)	6	Medium
213	3	4	(3 , 4)	12	High
241	2	5	(2 , 5)	10	High
243	5	4	(5 , 4)	20	Extreme
244	2	1	(2 , 1)	2	Low
245	5	4	(5 , 4)	20	Extreme

the corresponding matrix cell is c(3, 2), and the risk value is 6. Following the matrix coloring scheme in Fig. 2, the risk level will be medium. Table 6 demonstrates the risk levels for Lane-Keeping threats following the risk matrix approach.

4.5. Risk treatment for ADAS

For the Lane-Keeping use case, the information disclosure threat where connected parts of the dynamic gateway acquire more information than allowed (ID 244) will be accepted because of its low risk level. The remaining risk will be reduced by adopting a mitigation countermeasure. However, risk analysis should be repeated later, when the development process progresses and more details are provided by the upcoming design. Therefore, we wrote five security requirements (Sreq) to resolve these risks, as follows:

- *Sreq1*: The CAN bus flow between the dynamic gateway and Lane-Keeping ECU should be encrypted.
- *Sreq2*: The accessibility of the CAN bus between the dynamic gateway and the Lane-Keeping ECU should be as limited as possible.
- *Sreq3*: Should use signatures with certificates and private keys in special trust zones for communication between the dynamic gateway and the Lane-Keeping ECU?
- *Sreq4*: The Lane-Keeping ECU should rely on another source of data besides cameras.
- *Sreq5*: Should use authentication with certificates and a trust zone for the private key for the communication between the dynamic gateway and the Lane-Keeping ECU.

5. Risk analysis automation tool

Our proposed framework has many activities that can be automated. MS TMT provides automated threat generation based on the designed DFD. However, it does not provide automated and modifiable generations for impact ratings, attack feasibility ratings, and risk assessment for the generated threats that accommodate the ISO 21434:2021 standard recommended metrics. To overcome that, we implement a tool that extends the work of MS TMT and automates ratings and risk assessment activities according to recommended metrics. The tool is a prototype that was written in plain Java as a standalone application.

Before using the tool, we prepared the risk assessment template, which contains a sample of NCC-Group risks that were assessed according to our framework metrics. The generated threat report that is exported from MS TMT as an Excel file needs to be imported into the tool. Then, the template can be applied to automatically assess each threat, as shown in Fig. 6. Additionally, metric values can be changed for each threat individually and recalculated again. By clicking the View Selected Row button, the user can see further details about the selected threat and can change the risk metrics values in the opened frame. After finishing the risk assessment task, the user can export his work for future modifications.

In order to gain more confidence in the proposed framework from the point of view of others and to see its usefulness, two specialized experts participated in a risk survey (it was hard to find more experts in the field to get them involved in our study). The first works in a vocational school in the field of car education. The second is an engineer who specializes in an automotive maintenance workshop that provides software services to car customers. After reviewing and testing the framework, we asked them to complete a questionnaire to get an idea of the usability and usefulness of the proposed framework in terms of its difficulty, time, and effort required to use it. The evaluation includes the implementation of the manual activities as well as the automation tool. The result of the evaluation is described in Table 7. By analyzing the questionnaire results, we notice that the framework's usability is, in general, easy to medium for almost all activities except for attack path analysis, which is considered hard and time- and effort-consuming. This is reasonable since such an activity needs security expert participation to yield the best results in the shortest amount of time and effort.

6. Discussion

The framework is designed in structured phases following the ISO/SAE 21434:2021 standard recommendations. Also, the previous frameworks constitute a solid base to rely on when considering phase arrangement, best practices, metrics usage, and evaluation methods. In this section, a discussion about the proposed framework concerning benefits and challenges is provided. There are some insights that are worthy of being emphasized:

- CVSS v2.0 is recommended by the ISO/SAE 21434:2021 standard to rate attack feasibility. However, version 3.1 is newly released, but it includes new metrics such as scope (S) and user interaction (UI) which may be difficult to estimate and will add unnecessary complexity to the framework. Moreover, the maximum value of the overall score is 6.8, which is considered not normalized to 10 as v2.0. Additionally, the use of CVSS v2.0 is fair enough for concept-phase TARA analysis.
- We propose an attacker-based framework where threats are presented from the attacker's perspective. Therefore, the elicited security requirements focus on making it hard to compromise vehicles software and reducing the consequences of an expected attack. On the other hand, there are other unknown attacks that must be considered when adopting security solutions that make the entire vehicle as secure as possible. This can be achieved by performing iterative TARA analysis at each phase of the automotive SDLC. This mechanism provides the ability to conduct TARA analysis again at any level of detail. Therefore, additional fine-grained assessment can be performed later on by decomposing a process into sub-processes.
- For impact and attack feasibility ratings, we provide initial ratings for each threat. Automotive security experts should be involved in feasibility ratings. Safety experts should also be involved in impact ratings to make the analysis more accurate and useful. However, in our work, two domain experts are involved in the rating process for both threats damage impact and attack feasibility. The first is to work in the auto education department of a vocational school. The second is a professional engineer who runs an auto repair shop and provides customers with software services such as vehicle upgrades.
- The main purpose of the automation tool is to automate a set of activities that may consume time and effort from the risk analyst. Moreover, if a security expert prepared the template with cooperation from a safety expert, then a non-expert risk analyst can rely on the tool to resolve the risk assessment for the generated threats. Besides that, metric modifiability provides more agility so that the tool will suit any additional assumptions.
- In framework evaluation, DFD drawing and STRIDE analysis were conducted using the MS TMT. However, Sion et al. argue that DFD are not enough for security threat analysis (Sion et al., 2020). They

Table 7
Framework usability questionnaire results.

		Difficulty High/Medium/Low		Consumed Time High/Medium/Low		Submitted Effort High/Medium/Low	
	Expert1	Expert2	Expert1	Expert2	Expert1	Expert2	
Use case analysis	Assets Identification	E	E	M	L	M	L
	DFD	M	E	M	M	L	M
	Attacker Identification	E	E	L	L	L	L
Threat analysis	STRIDE	E	E	L	L	L	L
	Threat Scenario	M	M	L	M	L	L
	Damage Scenario	H	M	M	L	M	M
	Impact Rating	M	M	M	M	M	H
Attack analysis	Path Analysis	H	H	H	H	H	H
	Feasibility Rating	M	M	M	M	H	M
Risk assessment	Risk Value	E	E	L	L	L	L
	Risk Level	E	E	L	L	L	L
	Simulator Tool	E	M	L	L	L	L
	Risk Treatment	M	E	L	L	L	M

discuss some DFD weaknesses, such as data modeling, deployment information, security concepts, and abstraction levels. But DFD is still easy to use, and other diagramming languages that consider these weaknesses need to be evaluated in terms of efficiency and time and effort costs.

7. Conclusion

Modern vehicle software becomes vulnerable to the risk of cybersecurity attacks. This work proposes a systematic threat analysis and risk assessment framework that is designed according to the activities, workflow, and recommendations of ISO/SAE 21434:2021 Road Vehicles—cybersecurity Engineering Standard. We conclude that the framework is effective and helpful for risk analysts to address threats and assess risks early in the vehicle software development life cycle. However, to eliminate the issue of rating uncertainty with which we were faced in the running example, a crowd of cross-functional experts should be involved for use case description and rating activities. The crowd should include domain, security, and safety experts.

Most importantly, we emphasize the necessity of utilizing automation tools to facilitate risk analysis. We found that the Microsoft Threat Modeling Tool is a very useful tool that we strongly recommend being used for threat analysis in the automotive industry. On the other hand, a realistic MS TMT template should be designed carefully by the Original Equipment Manufacturer (OME) to generate more concrete threats. Additionally, our tool contributes to automating the entire process by extending MS TMT in order to automate the whole framework's activities.

Regarding ADAS use cases, we found that modern vehicles are vulnerable to potential safety-related threats. Therefore, for designing secure vehicles, in-vehicle secure communication is first required. Secondly, risk mitigation countermeasures are also required to resolve risks that threaten the vehicle from external communication. Nevertheless, even after securing internal and external communication, there are unknown types of risks that have not been addressed yet. Consequently, risk analysis should be performed iteratively to overcome as many potential threats as possible.

For future work, it would be useful to create one tool that automates all the ISO/SAE 21434:2021-based proposed framework activities instead of using the MS TMT and its extension. In addition to source, target, and interaction, the tool should take the nature of the data into consideration while designing a data flow diagram.

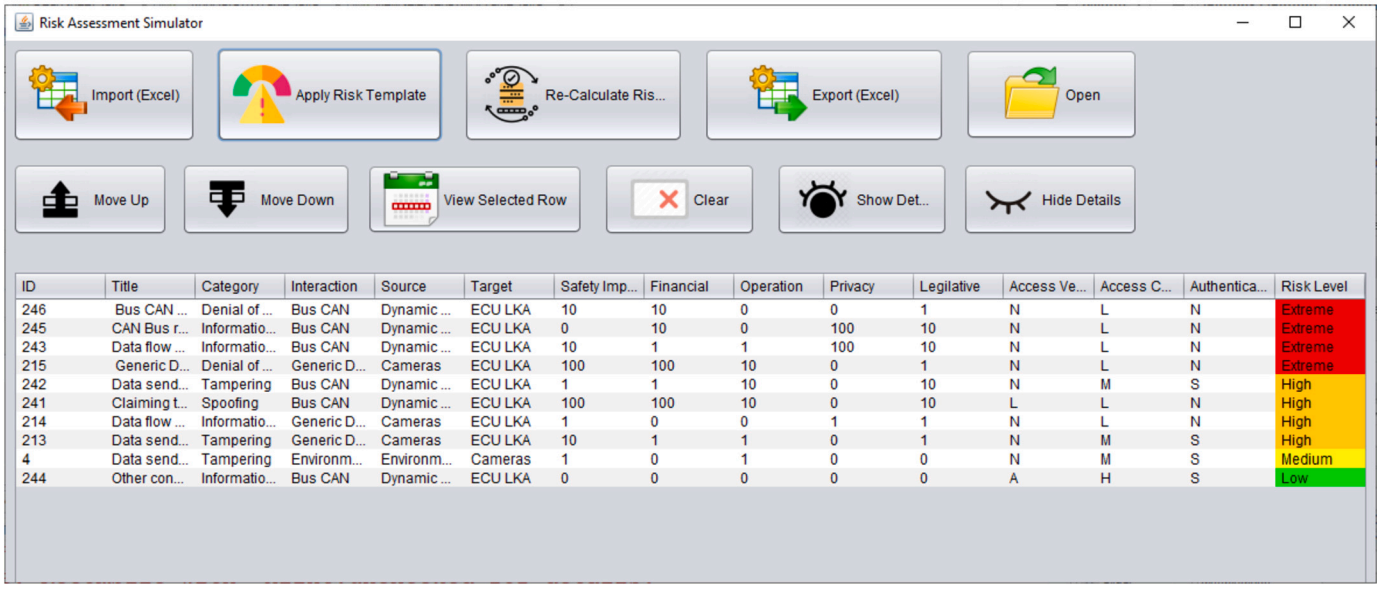

Fig. 6. Simulator main frame.

CRediT authorship contribution statement

Zaina Abuabed: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – Original Draft Preparation, Visualization, Data Curation. **Ahmad Alsadeh:** Conceptualization, Validation, Writing – Review & Editing, Supervision, Data Curation, Investigation. **Adel Taweel:** Conceptualization, Writing – Review & Editing.

Declaration of competing interest

We, [Zaina Abuabed, Ahmad Alsadeh, and Adel Taweel], declare that we have no conflicts of interest regarding the submission of this manuscript to the Journal of Computers & Security. Any organization or institution that might have a stake in the outcome of this research does not currently employ us. Furthermore, we have not received any financial support from any organization or institution related to this research.

Data availability

Data will be made available on request.

Acknowledgements

We express gratitude to Mahmoud Dappose and Mohammad Al-Ashqar, local automotive experts, for their invaluable contributions to the evaluation of the proposed framework. Mahmoud, an automotive workshop owner for vehicle software maintenance and enhancements, provided invaluable insights to ensure the framework's accuracy and effectiveness. Mohammad, an automotive engineer at the Automotive Section of the Ministry of Education Industrial Secondary School, contributed to the comprehensive evaluation of threats and their potential impact on the use case. Their dedication and support have significantly enhanced the quality and reliability of the framework.

References

Aksu, D., Aydin, M.A., 2022. Mga-ids: optimal feature subset selection for anomaly detection framework on in-vehicle networks-can bus based on genetic algorithm and intrusion detection approach. Comput. Secur. 118, 102717.

Alberts, C.J., Dorofee, A.J., 2001. OCTAVE Method Implementation Guide Version 2.0. Volume 1: Introduction. Technical Report. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

Algadah, K.M., Alaboodi, A.S., 2019. Anti-lock braking system components modelling. Int. J. Innov. Technol. Explor. Eng. 9, 3969–3975.

Algarni, A.M., Thayananthan, V., 2023. Autonomous vehicles with a 6g-based intelligent cybersecurity model. IEEE Access 11, 15284–15296.

Benyahya, M., Collen, A., Kechagia, S., Nijdam, N.A., 2022. Automated city shuttles: mapping the key challenges in cybersecurity, privacy and standards to future developments. Comput. Secur. 122, 102904.

Bolovinou, A., Atmaca, U.I., Sheik, A.T., Ur-Rehman, O., Wallraf, G., Amditis, A., 2019. Tara+: controllability-aware threat analysis and risk assessment for l3 automated driving systems. In: 2019 IEEE Intelligent Vehicles Symposium (IV), pp. 8–13.

Charette, R.N., 2009. This car runs on code. IEEE Spectr. 46, 3.

Ebrahimi, M., Striessnig, C., Triginer, J.C., Schmittner, C., 2022. Identification and verification of attack-tree threat models in connected vehicles. arXiv preprint. arXiv: 2212.14435.

Ghosh, S., Zaboli, A., Hong, J., Kwon, J., 2023. An integrated approach of threat analysis for autonomous vehicles perception system. IEEE Access 11, 14752–14777.

GmbH, V.V.R., 2022. Spider a mobile hil platform for fast, flexible reproducible adas or sensor tests. https://www.v2c2.at/spider/.

Group, U.S.S.S.W., et al., 2005. Adaptive cruise control system overview. In: 5th Meeting of the US Software System Safety Working Group, pp. 1–7.

Hamad, M., Prevelakis, V., 2020. Savta: a hybrid vehicular threat model: overview and case study. Information 11, 273.

Henniger, O., Ruddle, A., Seudié, H., Weyl, B., Wolf, M., Wollinger, T., 2009. Securing vehicular on-board it systems: the evita project.

Howard, M., Lipner, S., 2006. The Security Development Lifecycle. Microsoft Press, USA.

Islam, M.M., Lautenbach, A., Sandberg, C., Olovsson, T., 2016. A risk assessment framework for automotive embedded systems. In: Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security. Association for Computing Machinery, New York, NY, USA, pp. 3–14. https://doi.org/10.1145/2899015.2899018.

ISO, 2011. ISO 26262: Road vehicles-Functional safety. Technical Report. International Organization for Standardizatio.

ISO/IEC, 2004. ISO/IEC 13335:2004 Information technology — Security techniques — Management of information and communications technology security. Technical Report. International Organization for Standardization.

ISO/SEA, 2021. ISO/SAE 21434:2021 Road vehicles — Cybersecurity engineering. Technical Report. International Organization for Standardization. https://www.iso.org/standard/70918.html.

Khan, S.K., Shiwakoti, N., Stasinopoulos, P., 2022. A conceptual system dynamics model for cybersecurity assessment of connected and autonomous vehicles. Accid. Anal. Prev. 165, 106515.

Kim, K., Kim, J.S., Jeong, S., Park, J.H., Kim, H.K., 2021. Cybersecurity for autonomous vehicles: review of attacks and defense. Comput. Secur. 103, 102150.

Kong, H.K., Hong, M.K., Kim, T.S., 2018. Security risk assessment framework for smart car using the attack tree analysis. J. Ambient Intell. Humaniz. Comput. 9, 531–551.

Kuehn, J.T., 2017. Threat analysis and military history. https://networks.h-net.org/node/12840/blog/hand-grenade-week/164631/threat-analysis-and-military-history.

Kukkala, V.K., Tunnell, J., Pasricha, S., Bradley, T., 2018. Advanced driver-assistance systems: a path toward autonomous vehicles. IEEE Consum. Electron. Mag. 7, 18–25.

Lautenbach, A., Almgren, M., Olovsson, T., 2021. Proposing heavens 2.0–an automotive risk assessment model. In: Proceedings of the 5th ACM Computer Science in Cars Symposium, pp. 1–12.

Li, Y., Chen, L., Huang, H., Li, X., Xu, W., Zheng, L., Huang, J., 2016. Nighttime lane markings recognition based on canny detection and hough transform. In: 2016 IEEE International Conference on Real-Time Computing and Robotics (RCAR). IEEE, pp. 411–415.

Luo, F., Jiang, Y., Zhang, Z., Ren, Y., Hou, S., 2021. Threat analysis and risk assessment for connected vehicles: a survey. Secur. Commun. Netw. 2021, 1–19.

Macher, G., Sporer, H., Berlach, R., Armengaud, E., Kreiner, C., 2015. Sahara: a security-aware hazard and risk analysis method. In: Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition. EDA Consortium, San Jose, CA, USA, pp. 621–624.

Mell, P., Scarfone, K., Romanosky, S., 2007. A complete guide to the common vulnerability scoring system version 2.0.

Miller, C., Valasek, C., 2014. A survey of remote automotive attack surfaces. Black hat USA, 94.

Monteuuis, J.P., Boudguiga, A., Zhang, J., Labiod, H., Servel, A., Urien, P., 2018. Sara: security automotive risk analysis method. In: Proceedings of the 4th ACM Workshop on Cyber-Physical System Security. Association for Computing Machinery, New York, NY, USA, pp. 3–14. https://doi.org/10.1145/3198458.3198465.

Nccgroup, 2016. Ncc group template for the Microsoft threat modeling tool 2016 for automotive security. Available online at https://github.com/nccgroup/The_Automotive_Threat_Modeling_Template.

Ni, H., Chen, A., Chen, N., 2010. Some extensions on risk matrix approach. Saf. Sci. 48, 1269–1278.

Oka, D.K., 2021. Building Secure Cars: Assuring the Automotive Software Development Lifecycle. John Wiley & Sons.

Plappert, C., Zelle, D., Gadacz, H., Rieke, R., Scheuermann, D., Krauß, C., 2021. Attack surface assessment for cybersecurity engineering in the automotive domain. In: 2021 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), pp. 266–275.

Ren, D., Du, S., Zhu, H., 2011. A novel attack tree based risk assessment approach for location privacy preservation in the vanets. In: 2011 IEEE International Conference on Communications (ICC), pp. 1–5.

Rossebo, J.E.Y., Cadzow, S., Sijben, P., 2007. Etvra, a threat, vulnerability and risk assessment method and tool for eEurope. In: Proceedings of the Second International Conference on Availability, Reliability and Security. IEEE Computer Society, USA, pp. 925–933.

SAE International, 2016. SAE J3061: Cybersecurity guidebook for cyber-physical vehicle systems. Technical Report. Society of Automotive Engineers.

SAE International, 2020. SAE J3016 automated-driving graphic. Technical Report. Society of Automotive Engineers. https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic.

Sion, L., Yskout, K., Van Landuyt, D., van Den Berghe, A., Joosen, W., 2020. Security threat modeling: are data flow diagrams enough? In: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, pp. 254–257.

Synopsys, 2022. What is adas (advanced driver assistance systems)? Overview of adas applications. https://www.synopsys.com/automotive/what-is-adas.html.

Tzu, S., 2017. The Art of War-Sun Tzu. CreateSpace Independent Publishing Platform. Translated by Lionel Giles, M.A.

UcedaVelez, T., Morana, M.M., 2015. Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis, 1st ed. John Wiley & Sons.

Wang, Y., Wang, Y., Qin, H., Ji, H., Zhang, Y., Wang, J., 2021. A systematic risk assessment framework of automotive cybersecurity. Automot. Innov. 4, 253–261.

Winner, H., Prokop, G., Maurer, M., 2018. Automotive Systems Engineering II, vol. 1. Springer.

Winsen, S., 2017. Threat modelling for future vehicles: on identifying and analysing threats for future autonomous and connected vehicles. Master's thesis. University of Twente.

Wolf, M., 2019. Combining safety and security threat modeling to improve automotive penetration testing. Master's thesis. Ulm University.