Integrated Automation for Threat Analysis and Risk Assessment in Automotive Cybersecurity Through Attack Graphs

Mera Nizam-Edden Saulaiman[1], Akos Csilling[2] and Miklos Kozlovszky[3]

[1]Doctoral School of Applied Informatics and Applied Mathematics, Óbuda University, Bécsi út 96/b, 1034 Budapest, Hungary; mera.abbassi@stud.uni-obuda.hu

[2]Robert Bosch Kft., Gyömrői út 104, 1103 Budapest, Hungary; akos.csilling@hu.bosch.com

[3]John von Neumann Faculty of Informatics, Óbuda University kozlovszky.miklos@nik.uni-obuda.hu, Bécsi út 96/b, 1034 Budapest, Hungary, Medical Device Research Group, LPDS, Institute for Computer Science and Control (SZTAKI), Hungarian Research Network (HUN-REN), kozlovszky.miklos@sztaki.hu, Kende u. 13-17, 1111 Budapest, Hungary

Abstract: Attack graphs contribute to the evaluation of network security vulnerabilities, offering a visualization of possible attack paths. Despite their common use in IT security for analyzing system vulnerabilities, attack graphs are not commonly used in the automotive sector. As smart vehicles increasingly rely on 5G networks for high-bandwidth, low-latency communication – necessary for advanced vehicle-to-everything (V2X) services and sensor data processing – security concerns escalate. The complexity of 5G-enabled vehicles significantly expands a vehicle's attack surface. The ISO/SAE 21434 standard establishes a framework for securing road vehicle systems. The Threat Analysis and Risk Assessment (TARA) process, a vital part of this standard, helps identify and mitigate security risks. However, the current TARA process relies heavily on manual effort to identify potential attack vectors and assess risks. This can be time consuming, resource-intensive, and prone to human error. This paper discusses the concept of an automated attack graph generation tool specifically designed for automotive threat analysis. We propose a new Graph-based Attack Path Prioritization tool (GAPP), tailored for automotive networks. GAPP focuses on generating attack paths, assessing their feasibility, and identifying the most likely attack scenarios. This aims to enhance the efficiency, comprehensiveness, and accuracy of the TARA process in evaluating network security.

Keywords: Automotive security; 5G; Threat analysis and risk assessment; Attack graph; ISO/SAE 21434

1 Introduction

In the rapidly advancing world of automotive technology, vehicles are no longer just means of transportation; they have transformed into sophisticated, interconnected networks, equipped with advanced technologies like GPS, Wi-Fi, Bluetooth, and Vehicle-to-everything (V2X) communications [1]. 5G is a key enabler for the various V2X communication use cases, including Vehicle-to-Vehicle (V2V), Vehicle-to- Infrastructure (V2I), Vehicle-to-Network (V2N), and Vehicle-to-Pedestrian (V2P). 5G New Radio (NR) uses Uu and PC5 radio interfaces for V2X communications, the Uu radio interface is employed for V2N communications, while the PC5 radio interface (sidelink) allows direct V2V, V2P, and V2I communications within the infrastructure coverage and beyond for V2V and V2P. 5G-V2X communications are anticipated to significantly impact the automotive industry, supporting diverse use cases, including safety, non-safety, and infotainment applications such as fully automated driving, cooperative manoeuvring, and cooperative perception [2].

Traditionally, securing a vehicle focused on preventing physical break-ins. However, with the advancement of smart cars, the security paradigm has expanded dramatically. The attack surface of a smart car extends far beyond the traditional threats, encompassing a complex array of cybervulnerabilities, a concern that has become more evident in recent years [3] [4].

Recognizing the importance of cybersecurity in this connected environment, the automotive industry has adopted key standards like the ISO/SAE 21434, driven by the UNECE WP.29 regulation [5]. The ISO/SAE 21434 standard outlines requirements for managing cybersecurity risks throughout various stages, from concept and product development to production, operation, maintenance, and finally decommissioning. It provides a framework to specify the requirements for cybersecurity processes throughout the vehicle life cycle.

The process of Threat and Risk Assessment (TARA), mentioned in Clause 15 of the standard aims to identify and predict potential threats early in the design process, helping to lower the risk of cyberattacks on new vehicles. Its main goal is to list all possible threats and assess how they could affect users, guiding decisions on how to prevent these threats early enough in the development process.

Currently, the implementation of TARA in the automotive industry largely involves manual efforts [6] Experts manually define threat scenarios, often using attack trees to map out and assess potential attack paths. This manual approach is both time-intensive and susceptible to human error. It tends to analyse vulnerabilities independently, neglecting the potential compound effect of multiple vulnerabilities interacting, which can lead to severe system compromises.

To address these issues, we propose the Graph-based Attack Path Prioritization Tool (GAPP). GAPP's initial concept was introduced in our previous paper [7], a novel approach that integrates graphical modelling with the TARA methodology.

This approach aims to automate the generation of attack paths in a predefined network, by using the data input specified in the TARA process and applying GAPP's logic to generate an attack graph and calculate the feasibility of the generated attack paths.

This paper is structured as follows: it begins with an overview of modern automotive technologies, and the corresponding cybersecurity risks. In Section 2 we present different modelling approaches in cybersecurity, exploring the use of graph-based models for security analysis. In Section 3 we present various challenges of security in automotive networks, including potential attack surfaces like in-vehicle communication systems and external communication channels. In Section 4, we outline the TARA process as defined in the ISO/SAE 21434 standard. Finally, in Section 5 we introduce the GAPP methodology, its components, concept, logic, and application in automating attack path generation and assessment. Then, we conclude the paper with a discussion on the implications of the findings and potential areas for future research.

2 Modeling in Security Analysis

Modeling in security provides a structured approach for identifying and analyzing potential threats, vulnerabilities, and attack vectors within a system. Through creating detailed models of a system's architecture, and documenting and evaluating the potential security threats. Security experts, developers, and management decision-makers can anticipate how an attacker might exploit weaknesses, thus making informed decisions. Consequently, it helps identify issues early in the development life cycle that might be missed in traditional testing.

2.1 Graph-based Security Models

Graph-based modelling is used in cybersecurity to visualize and analyse potential vulnerabilities in the system. This approach uses graph theory to map out system components and connections, representing them as nodes and edges. By using graphs, it becomes easier to identify vulnerabilities, understand how various parts of the network are interconnected, and how an attacker could move through the system. It's particularly useful in complex networks where understanding the relationships and potential attack vectors is challenging.

2.1.1 Threat Modeling

Threat modelling aims at understanding and visualizing potential threats or attack paths within a system. This process involves mapping out all possible routes an adversary might lead to successfully compromise a system and realize a threat. Researchers in [8] classified attack modelling methods into ”manual/automatic” and ”formal/graphical”. As discussed in [6], Formal-based methods use mathematical models like tables, texts, and formulas whilst graph-based methods use various graph-based models to analyse the threats and risks of the system. Attack modelling methods can be both formal and graphical [9]. The researchers in [8] found, based on a comprehensive survey of the literature in the IT domain that 18 out of 20 articles used manual modelling, while only 4 used automatic modelling, which makes the topic of automation in this field a challenging and a hot topic for researchers.

Xu et al. (2012) [10] used formal threat models to automate security test generation, but manual analysis was still needed. Arsac et al. (2011) [11] automated protocol validation with SATMC, but their MA threat model was manual. Baquero et al. (2015) [12] used Microsoft's SDL Tool for automated analysis, but the modeling process was manual. Musman and Turner (2018) [13] introduced algorithms to automate expert tasks, but some analysis remained manual. Graph-based Security Models (GrSMs) are primarily categorized into two types: Graph-based and Tree-based models [14].

Attack graph (AG) technique models all the possible paths an attacker might take to achieve his goals. It consists of nodes representing the network’s state during an attack and edges indicating state transitions. Unlike other methods, attack graphs consider the possibility of combining multiple vulnerabilities to form an attack.

Attack trees (AT) use a hierarchical structure with a root representing the attack target and nodes below depicting events leading to this target. There are LEAF nodes (elementary attack steps), OR nodes (indicating alternative attack paths), and AND nodes (where all child nodes must occur for the attack to happen).

There are different types of trees in GrSMs, including attack trees (AT), defence trees (DT), attack countermeasure trees (ACT), and attack fault trees (AFT). Also, different types of attack graphs can be considered, including logical attack graphs, state attack graphs, dependency attack graphs, and multi-prerequisite graphs [15].

Several attack-graph-based tools have been developed. For example, MulVAL [6] derives logical attack graphs by associating vulnerabilities extracted from scans with the probability that an adversary could successfully conduct a successful attack to exploit them.

3 Automotive Security

The attack surface in modern vehicles includes a range from wired in-vehicle communication to cellular communication. We can categorize these into in-vehicle communication systems and external communication systems.

In-Vehicle Communication: wired communication protocols, including FlexRay, Local Interconnect Network (LIN), and Controller Area Network (CAN), connecting various Electronic Control Units (ECUs) in the vehicle [6].

External Communication: wireless communication protocols mainly support the advanced features in modern vehicles, like GPS, Wi-Fi, Bluetooth, cellular connections, and 5G. V2X is a relatively new technology that connects the vehicle to its surroundings, it includes different communication technologies such as Dedicated Short-Range Communications (DSRC), Cellular-V2X (C-V2X) that includes Long-Term Evolution V2X (LTE-V2X), and 5G (5G-V2X). These external links create a wide attack surface for cyberattacks [16].

Each technology presents unique security challenges. We can categories the automotive attack surface depending on the level of access required [6]:

Physical Access: It requires Direct interaction with the vehicle system through interfaces like OBD-II ports and ECU connections, even DVD players can be a physical entry point. In addition, we have the ECUs which depend on software and firmware which might present vulnerabilities and thus potential entry points.

Short-Range Access: It depends on using short-range wireless technologies such as Bluetooth and NFC, which are beneficial for features like keyless entry (unlocking doors or starting the engine) but vulnerable to attacks like interception or spoofing.

Long-Range Access: Remote communication capabilities like cellular data, Wi-Fi, and 5G in V2N communication, where it connects to the 5G edge, 5G core, and 5G data networks to provide a variety of services to users, from comfort and infotainment to remote driving.

4 Threat Analysis and Risk Assessment TARA

The ISO/SAE 21434 standard [5] is critical for the approval of automotive electrical and electronic (E/E) systems, it mandates a rigorous approach to identify and manage potential cybersecurity threats within the vehicle. This standard introduces the Threat Analysis and Risk Assessment (TARA) process as an essential component. This process involves a thorough examination of the system to identify potential threats that could lead to security breaches, it evaluates the likelihood and potential impact of these threats, encompassing all conceivable attack scenarios on the system [17].

This process visualizes possible attacks in the form of attack trees, the goal is to outline the individual and sequential steps an attacker might take to compromise and violate security properties of the system, i.e., confidentiality, integrity, and availability, which in turn will endanger important assets in the system such as safety features or intellectual property.

TARA uses existing knowledge about known attacks which can be available through vulnerability databases or security expert knowledge. This existing attack knowledge serves as a reference point to identify and mitigate similar vulnerabilities in new systems, preventing the recurrence of known security issues [16]. An automated model-based approach for attack analysis would make the process more accurate and flexible along the development life cycle [7].

The TARA takes input from the previous steps in the ISO/SAE standard as input. The process involves seven steps where each step has a defined input and a defined output as shown in Figure 1.

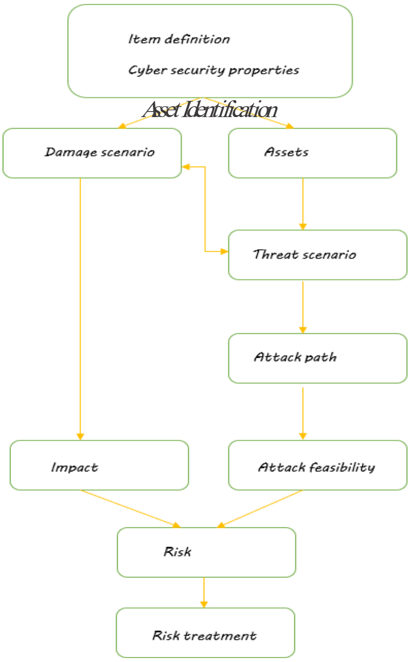


Figure 1
TARA process according to the ISO/SAE 21434

4.1 Asset Identification

The first step of the TARA process is the asset identification. Building on inputs from the preceding phases of the security engineering process, namely the item definition and security properties, this step is important in determining key system elements and potential damage scenarios, where damage scenarios are the possible adverse consequences to an asset, describing the harm it causes to the road user.

Assets refer to any system components vulnerable to security threats, that could be involved in a damage scenario, ranging from data to functional units. Each asset is linked to a cybersecurity property i.e., integrity, confidentiality, or availability. The primary outputs of this step are the identification of assets and the definition of potential damage scenarios.

4.2 Threat Scenario Identification

This process begins with the input of the previous step, i.e., identified assets, and their associated damage scenarios. Using these inputs, security experts engage in collaborative discussions or brainstorming sessions to define possible threat scenarios. These scenarios represent different situations or use cases where the cybersecurity properties of an asset could be compromised to realize a damage scenario.

4.3 Impact Rating

The Impact rating describes the damage rate of a damage scenario across defined impact categories, namely safety, financial, operational, and privacy. For each category, the impact of a potential damage scenario is rated on a scale ranging from severe to negligible. This process of impact rating helps in quantifying the degrees of impact that different scenarios might have on the system.

4.4 Attack Path Analysis

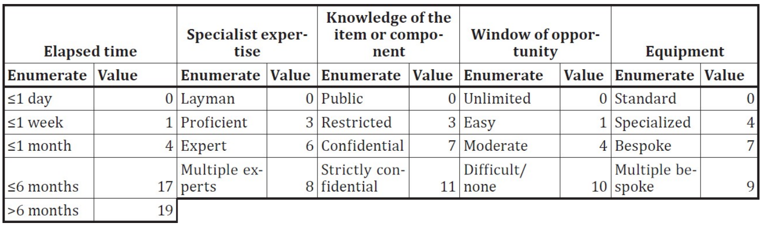
There are two primary methodologies for enumerating these paths: the top-down approach, using techniques like attack trees, and the bottom-up approach, which constructs attack paths starting from identified vulnerabilities. This process aims at finding all the attacks that realize a threat.

4.5 Attack Feasibility Rating

This step involves evaluating the feasibility of each identified attack path, categorized as high, medium, low, or very low. The assessment is based on different approaches as defined in the standard [5].

- Attack Potential-Based Approach:
  - Elapsed time: This involves estimating the time it takes for an attacker to identify a vulnerability, develop an attack strategy, and then execute the attack Figure 2 .
  - Specialist expertise: Evaluating the attacker’s level of technical skill and practical knowledge required by an attacker to successfully exploit a vulnerability. It considers both theoretical understanding and practical application capabilities in cybersecurity.

- Knowledge of the item or component: Understanding the attacker's familiarity with the system's principles product type, or attack methods relevant to the system.
- Window of opportunity: The time available for the attacker to exploit a vulnerability. It accounts for the availability and accessibility of the system to the attacker, and the time frame in which the attack can be effectively executed.
- Equipment: This includes the specific tools, software, or hardware required for the attacker to identify or exploit a vulnerability.



Elapsed time		Specialist expertise		Knowledge of the item or component		Window of opportunity		Equipment	
Enumerate	Value	Enumerate	Value	Enumerate	Value	Enumerate	Value	Enumerate	Value
≤1 day	0	Layman	0	Public	0	Unlimited	0	Standard	0
≤1 week	1	Proficient	3	Restricted	3	Easy	1	Specialized	4
≤1 month	4	Expert	6	Confidential	7	Moderate	4	Bespoke	7
≤6 months	17	Multiple experts	8	Strictly confidential	11	Difficult/ none	10	Multiple bespoke	9
>6 months	19								

Figure 2

Attack feasibility rating for Attack potential-based approach

- CVSS-Based Approach: This approach quantifies the severity and risk of a vulnerability using metrics that are divided into three groups, Basic group, Temporal group, and environmental group, we discussed them in detail in our previous paper [18]. Some of the main groups in the Base metrics can be summarized in the following list:
  - Attack vector: Assessing how an attack is initiated, whether it's through a network, locally, or via a physical medium.
  - Attack complexity: The evaluation of the difficulty level involved in executing an attack, considering the sophistication of techniques required.
  - Privileges required: Determining the level of access or permissions required for an attacker to successfully execute the attack.
  - User interaction: Considering the extent to which user action plays a role in the successful execution of the attack.
- Attack Vector-Based Approach: This focuses on evaluating the primary method or channel through which the attack is initiated. It assesses the ease or complexity of initiating an attack based on the chosen vector, influencing the overall feasibility of the attack.

4.6 Risk Value Determination

Involves evaluating the risk level of each identified threat scenario based on its impact and feasibility. The outcome of this evaluation is quantified as risk values, rated on a scale from 1 to 5. A value of 1 indicates minimal risk, reflecting a lower likelihood or impact of the threat scenario on road users.

To determine these risk values, two primary methods are used:

- Risk Matrix: This method involves a matrix system where risk levels are determined based on a combination of factors such as the likelihood of the threat and its potential impact. The matrix allows for a quick and systematic assessment of risk levels.
- Risk formulas: This approach uses specific formulas to calculate the risk value. These formulas consider various factors like the severity of the impact, the likelihood of occurrence, and the feasibility of attack execution.

4.7 Risk Treatment Decision

In this phase, appropriate risk management strategies are determined for each threat scenario, based on their assessed risk values.

The defined options for risk treatment include:

- Avoiding the risk
- Reducing the risk
- Sharing the risk
- Retaining the risk

By concentrating on the highest risk scenarios, the residual risk can be reduced effectively.

5 GAPP Methodology

The GAPP methodology employs a systematic approach to model and analyse the security of automotive systems, focusing on generating attack graphs. The process is structured into distinct phases. In the initial phase, the system is modelled, defining nodes representing assets, external interfaces, and internal components, each with its associated privilege states. The connectivity between the nodes captures the network topology. Subsequently, attack steps are defined, each with conditions specifying its logic and potential effects. The system model serves as inputs to GAPP to enumerate and evaluate all possible attack paths, prioritizing them based on severity and potential impact. The attack graph illustrates how attackers could progress from external interfaces, through internal nodes, to target assets. Throughout the process, threats are assigned to assets, and the privilege model guides the representation of privilege escalation.

Figure 3 shows how the GAPP automation model fits within the wider TARA framework.

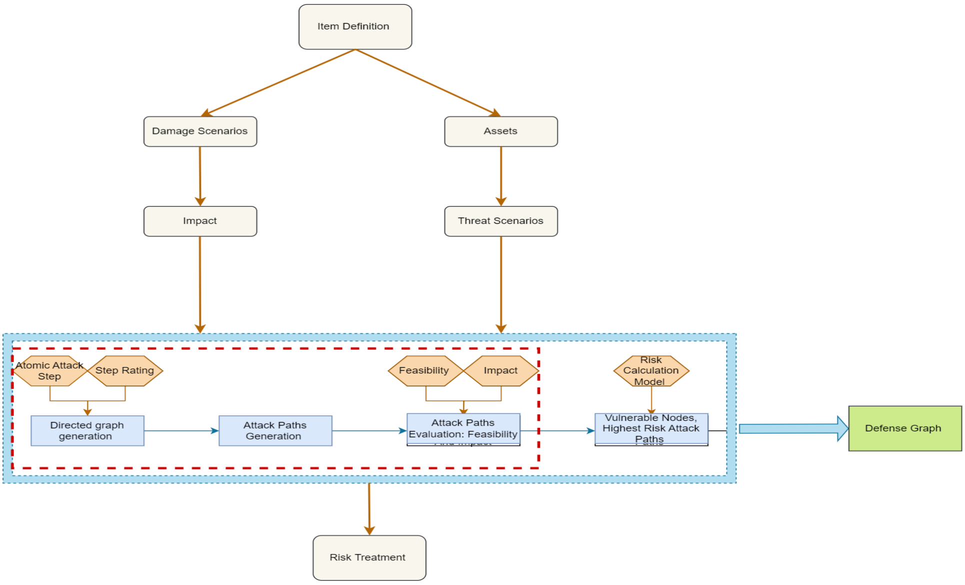


Figure 3
The GAPP automation model within the TARA framework [5]

5.1 System Model

The system to be modelled corresponds to the item definition according to ISO/SAE 21434, it must include enough of context to represent the attack vectors to be considered. The identified assets and other components are represented by nodes in the system model, including any external and internal connections among these assets.

The level of detail must be chosen appropriately to have a meaningful representation of the system. While the automation provided by the GAPP tool allows for the hierarchical representation of multiple levels of detail, the current study only considers a single level.

5.2 Example System

For illustration, we show the example system from the ISO/SAE 21434 standard.

This system illustrated in Figure 4 is designed to adjust the headlamp operation based on driver input, and switches between high and low beams in response to oncoming traffic. It is connected to the navigation ECU, which links to the gateway ECU via CAN communication. The navigation ECU is connected to

Bluetooth and 5G interfaces, while the gateway ECU is connected directly to an onboard diagnostics (OBD-II) interface.

The system model for our example is shown in Figure 5, Note that the CAN bus communications between the various ECUs are represented by nodes in the model.

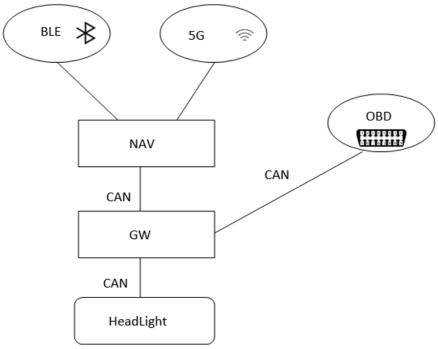


Figure 4
Head Light system

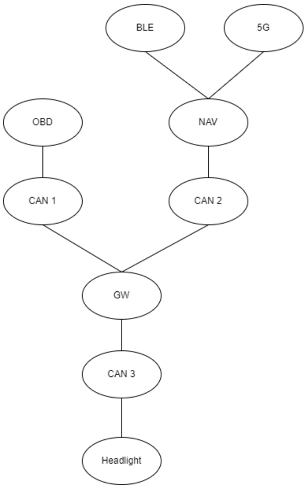


Figure 5
System model in GAPP representing the headlight system.

5.3 Nodes

In the GAPP implementation, we define various node types that will make up the system model:

Asset nodes: correspond to the assets identified earlier in the TARA process.

These are the assets that need protection, and they will represent the end nodes in our attack graph. When an attack path reaches an asset node, this means that we identified an attack path that breaks one of the security properties of an asset.

External interface: nodes represent the attack surfaces identified previously and correspond to the external interfaces of our system. These are the entry points for the attacker and will become the starting points of the attack paths, e.g., the BLE and 5G interfaces.

Internal nodes: correspond to the internal components and the networks connecting them. These include the NAV ECU, the Gateway ECU, and the CAN busses connecting them.

Each node has a state represented by the privilege level, representing the level of compromise reached by the attacker. The state changes with each attack step, representing transitions in the attack graph. This approach considers the privileges an attacker gains on network nodes as states, allowing for the analysis of multiple attack sequences by chaining various states depending on the pre- and post-conditions of the attack step. The nodes corresponding to the Headlight system shown above are listed in Table 1.

Table 1

List of nodes in the example headlight system

Node	Type
OBD	External Interface
BLE	External Interface
Cellular	External Interface
CAN1	Network - Internal node
NAV	Component - Internal node
CAN2	Network - Internal node
GW	Component - Internal node
CAN3	Network - Internal node
Headlight oncoming data	Asset
Headlight lamp request	Asset
Headlight FW	Asset

5.4 Connectivity

The concept of connectivity in this model describes the topology by specifying the reachability of each node within the network. In GAPP, connectivity refers to the persistent links between nodes. Unlike factors such as vulnerabilities and privileges, connectivity remains constant throughout the analysis. In our logic, a single node may form connections with multiple other nodes.

5.5 Threats

Threats that are defined early in the TARA process as discussed in 4 are assigned to the asset nodes. For threat classification, we follow the STRIDE classification of threats. STRIDE is a threat modelling framework used for risk assessment [19] STRIDE stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. Each classification targets one or more of the security properties: authenticity, integrity, non-repudiation, confidentiality, availability, and authorization [20], [21]. The threats are assigned to the assets in our example system as shown in Table 2.

Table 2

Threats assigned to the asset nodes according to the damage scenarios and threat scenarios defined

Node	Type	Threat
Headlight oncoming data	Asset	Spoofing, DOS
Headlight lamp request	Asset	Spoofing, DOS
Headlight FW	Asset	Information disclosure, Tampering

5.6 Privileges

GAPP's privilege model is based on the framework introduced by researchers in [22], which identifies five distinct types of privilege:

- Read/Write (Functional Communication Link): Involves reading and writing on communication links like CAN-Bus.
- Execute (Functional Component): Ability to trigger functions of a component without detailed knowledge.
- Read (Functional Component): Involves reading information or data from a component.
- Write (Functional Component): Ability to modify data of a component.
- Full Control (Functional Component): Total control over a vehicle component, similar to root privileges in IT systems.

Transition from State to State:

The transition between these privileges is hierarchical, escalating from lower to higher (e.g., from Read to Full Control).

The transition process involves exploiting vulnerabilities at each level to achieve a higher privilege level. For example, exploiting a vulnerability at the Read level may grant Execute privileges, this is defined by the logic of the tool as conditions.

This model enables the representation of a chain of privilege escalations, illustrating potential attack paths through a vehicle’s network.

5.7 Attack Steps

This concept is key for identifying and understanding attack paths within the system under analysis. An attack step is a distinct phase in a multi-stage cyberattack, each representing a specific action or exploit for a vulnerability within the network. The advantage of breaking down an attack into these steps is twofold. Firstly, it allows for the combination of steps from different sources to uncover new, previously unknown attack paths. Secondly, it enhances the adaptability of these steps to new system architectures. Isolating individual steps enables a more accurate assessment of an attack's feasibility, thereby allowing for a more precise determination of the attack's risk value. Essentially, understanding these steps helps identify and strengthen the weakest links in a system's security. An attack step involves moving from one node in a given state to the next, resulting in a change in the level of access (privilege). In our approach, each attack step has 5 parameters defined: (source node, destination node, network, privilege required, privilege gained). The source and destination node can be an interface, ECU, or asset. An attack step can be assigned to multiple nodes, for instance, a specific attack step, such as sniffing CAN communication, can be assigned to all the CAN channels in our network. Similarly, forwarding malicious messages through an ECU can be allocated to nodes like the GW node and the navigation node. Currently the attack steps which are defined are mapped to the corresponding node using the defined properties of the step.

In addition, the attack step has defined feasibility, this is currently defined by the security expert through the attack potential approach as discussed in [17]. Attack steps are derived from previously known attacks and from experts' knowledge, which will enable us to create a database of attack steps. This database encapsulates generalized attack steps, rendering them applicable across various automotive systems rather than being system specific. Integrated into cyber-security tools, this will be further discussed in our future work.

In Table 3 the attack steps exemplify the practical application of the GAPP methodology, the attack steps are extracted from the ISO/SAE 21434, and the description of attack steps is intentionally concise for simplicity. However, it's important to note that in practical applications, attack steps can be more granular and detailed. The level of detail can be extended to provide a comprehensive understanding of the specific actions and processes involved in each attack step, enhancing the accuracy and depth of the attack modelling process. The attack path begins with the step that exploits the vulnerabilities in the external interface nodes from the system model and finishes with the attack step targeting the asset of the headlight system.

Table 3
Attack steps for the headlight system in GAPP

Attack step	Node	Pre-condition	Post-condition
Compromise OBD	External Interface	1	5
Spoof the CAN communication	Network	1	2
Compromise Interface (BLE, 5G)	External Interface	1	2
Compromise ECU (navigation, GW)	ECU	2	4
Extract FW from GW	Asset	3	3
DOS of oncoming car information	Asset	1	2
Signal spoof of Headlamp data	Asset	1	2

5.8 Transition Model

The GAPP methodology initiates its assessment from the external interface node accessible to the attacker and meeting predefined conditions. The tool establishes the initial state of the network during the system model initialization. Upon locating the external interface node, GAPP's logic defines transitions between states aligning with the attack step conditions. During each transition, the tool checks the conditions of the attack step, if the step is successful it results in a change in the privilege level at the affected node, dynamically representing the attacker's progress.

In scenarios where the network contains nodes with access levels surpassing the minimal requirement, the model considers the privilege conditions fulfilled.

Each transition signifies a shift in the model's state, often resulting in a reduction in the network's overall security. These transitions have the potential to introduce new vulnerabilities to the affected nodes.

Collectively, these sequential attack steps compile the attack path, leading to the threat.

5.8.1 Feasibility

In the context of assessing attack feasibility in the GAPP framework, individual attack steps are evaluated for their potential of exploitation, rather than assessing the entire attack path as a whole, we evaluate feasibility according to the attack potential approach defined in the ISO/SAE 21434 as discussed in 4.5.

For determining the overall feasibility of an attack path, the GAPP tool uses a 'maximum value approach' as outlined in [23]. This method involves selecting the highest value of attack potential for each factor along the attack path, a concept also referenced in [24] and [7]. The application of attacker types in GAPP is contextualized within this assessment, where the selection of attacker types contributes to fixed ratings for expertise, knowledge, and equipment, with subsequent adjustments for time and access.

Mathematically, in GAPP, let R denote the set of all tuples of attack feasibility factors. Function affmax: R∗ → R represents a function that takes an arbitrary count k ∈ N of tuples of attack feasibility factors r1, . . ., rk as input. The function computes the maximum value for each attack feasibility factor, expressed as follows:

$$\text{affmax } [r_1, \ldots, r_k] :=$$

$$(\max\ [v_{1,1}, \ldots, v_{k,1}], \ldots, \max\ [v_{1,f}, \ldots, v_{k,f}])$$

where max [vi, j] refers to the maximum value of each attack feasibility factor j across the tuples r1, . . ., rk.

5.8.2 Attack Graph

In the final phase, the process involves listing the attack paths, evaluating their feasibility, and prioritizing the top highest-risk paths. Using the state transition model, GAPP systematically scans the model for all attack paths and calculates their feasibility.

Based on the state transition model and starting from the OBD Port, 5G, BLE interfaces, attack steps in Table 3 are associated with the nodes of the system model. For the sake of simplicity, we employed generic attack steps. This could involve relaying diagnostic messages either from the OBD port or from the GW and NAV ECU. In this attack scenario, the GW ECU forwards malicious messages leading to a Denial of Service (DOS) on the headlight data bus, compromising the availability security property of this asset. The graphical representation indicates three distinct paths that can exploit this threat. Notably, all three paths converge at the GW ECU, designating it as a high-risk component Figure 7.

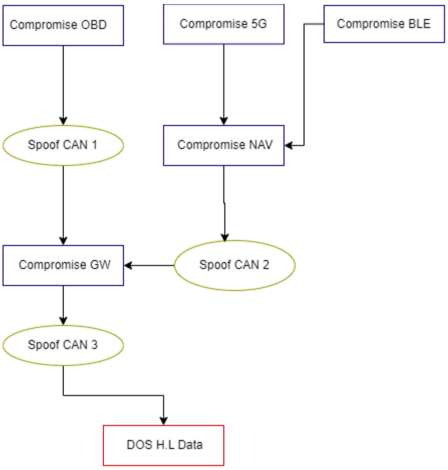


Figure 6
Attack graph for a DOS attack

The feasibility would be calculated for each of these paths by using the concept here 5.5.1, but for this paper, it's out of scope.

6 Discussion

The essence of creating an attack graph in GAPP is to systematically organize all relevant multi-stage attacks aimed at achieving a specific threat. This involves breaking down the overall attack goal into smaller sub-goals and, eventually, elementary actions necessary to accomplish the goal. The result is a systematic enumeration of all possible attack paths leading to the given threat. When iterated over all possible threats, the graph will contain all possible attack paths, allowing the selection of the highest risk scenarios.

In addition to focusing attention on the highest-risk attack path, the constructed graph also allows the identification of the attack steps with the largest contribution to the overall risk in the system. This in turn indicates vulnerable points in the architecture of the system that may need additional protection.

The GAPP methodology includes a hybrid approach combining both manual and automated aspects, the initial steps, such as asset identification, threat scenario definition and impact rating which are the inputs from initial steps in the TARA process. The manual input provides the foundation for the upcoming automated process. GAPP uses this information in addition to the attack step database, this database is coming from previous evaluation and input from security experts, the construction of this database can be initiated by breaking down previous attack paths into their atomic steps. As a next step in our research, we intend to publish a basic database gathered from previously published attacks.

While Sommer and Dürrwang in [25] addressed the need for an automotive vulnerability database tailored for automotive needs, their work, while valuable, lacked the properties necessary for our automation and use cases. Our research aims to bridge this gap by developing a database model which will facilitate integration and automation of vulnerability data into the GAPP tool.

In addition, a dedicated software tool to realize the creation, analysis, and derivation of attack paths within the GAPP framework will also be published. Evaluation through a comprehensive case study will further validate the effectiveness and applicability of GAPP's attack paths across diverse automotive architectures.

In a wider scope, the current work can be linked to other activities aimed at increasing the efficiency of security engineering work. In particular, the GAPP model can be linked with model-based system engineering methods that provide an abstraction of the system. The same model can then be used within the two frameworks.

Obtaining an accurate and complete data from the graph is a challenging task as we might face cases with false positives due to the incomplete knowledge of vulnerabilities, system configurations, or incomplete databases. In order to minimize this, we currently incorporate the review and feedback of security experts for the attack graph and attack paths found by the tool, in addition to continuously refining the algorithm. Another issue we might face is false negatives in cases of missed attack paths, a solution for this could be to always keep our vulnerability database up to date. Future work to explore the use of machine learning for the graph generation.

On the threat modelling side, the inclusion of attacker profiling, including their goals and methods can be incorporated into the model. Furthermore, a catalogue of threats, while certainly used implicitly in the form of professional experience, could be explicitly included in the attack step database. While all possible attacks on all possible automotive systems would be impossible to enumerate, the number of possible attack steps that could be included in a database with sufficient granularity for useful applications.

Conclusions

In concluding this paper, we highlight the GAPP methodology in advancing automotive cybersecurity analysis. Our approach, cantered around the dissection of cyberattacks into discrete steps, offers a refined lens for examining vulnerabilities in complex automotive systems. Through isolating each attack step, GAPP enables a more granular and precise assessment of potential threats, facilitating the identification of previously unrecognized attack paths. This methodology also proves invaluable in adapting existing security measures to new architectural paradigms, enhancing the transferability of security insights. The automated nature of GAPP in generating and evaluating attack paths represents a paradigm shift in threat analysis, providing a more streamlined, accurate, and efficient process. This paper's contributions lay the groundwork for further research and development in the field, aiming to fortify the cyberresilience of modern automotive systems against evolving threats.

Acknowledgment

The authors would like to thank the 2019-1.3.1-KK-2019- 00007 "Innovacios szolgaltato bazis letrehozasa diagnosztikai, terapias es kutatasi celu kiberorvosi rendszerek fejlesztesere" national project for the financial support.

The authors would like to thank AIAM (Applied Informatics and Applied Mathematics) doctoral school of Óbuda University, Budapest, Hungary for their support in this research.

References

[1] Mohammed, D.; Horváth, B. Vehicle Automation Impact on Traffic Flow and Stability: A Review of Literature. *Acta Polytech. Hung.* **2023**, *20*, 129-148

[2] Saulaiman, M. N.-E.; Kozlovszky, M.; Csilling, Á. A Survey on Vulnerabilities and Classification of Cyber-Attacks on 5G-V2X. In Proceedings of the 2021 IEEE 21st International Symposium on Computational Intelligence and Informatics (CINTI); November 2021; pp. 000235-000240

[3] Valasek, C.; Miller, C. Remote Exploitation of an Unaltered Passenger Vehicle. *Black Hat USA* **2015**, *2015*, 1-91

[4] Barik, M.; Sengupta, A.; Mazumdar, C. Attack Graph Generation and Analysis Techniques. *Def. Sci. J.* **2016**, *66*, 559, doi:10.14429/dsj.66.10795

[5] 14:00-17:00 ISO/SAE 21434:2021 Available online: https://www.iso.org/standard/70918.html (accessed on 22 December 2023)

[6] Saulaiman, M. N.-E.; Kozlovszky, M.; Csilling, Á.; Banati, A.; Benhamida, A. Overview of Attack Graph Generation For Automotive Systems. In Proceedings of the 2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC); July 2022; pp. 000135-000142

[7] Saulaiman, M.; Csilling, A.; Kozlovszky, M. Leveraging Attack Graphs in Automotive Threat Analysis and Risk Assessment.; Porto, Portugal, September 23 2023

[8] Xiong, W.; Lagerström, R. Threat Modeling – A Systematic Literature Review. *Comput. Secur.* **2019**, *84*, 53-69, doi:10.1016/j.cose.2019.03.010

[9] Xu, D.; Nygard, K.E. Threat-Driven Modeling and Verification of Secure Software Using Aspect-Oriented Petri Nets. *IEEE Trans. Softw. Eng.* **2006**, *32*, 265-278, doi:10.1260/136943306776986994

[10] Automated Security Test Generation with Formal Threat Models | IEEE Journals & Magazine | IEEE Xplore Available online: https://ieeexplore.ieee.org/abstract/document/6155723 (accessed on 12 July 2024)

[11] Arsac, W.; Bella, G.; Chantry, X.; Compagna, L. Multi-Attacker Protocol Validation. *J. Autom. Reason.* **2011**, *46*, 353-388, doi:10.1007/s10817-010-9185-y

[12] Baquero, A.O.; Kornecki, A.; Zalewski, J. Threat Modeling for Aviation Computer Security. *crosstalk* **2015**, *28*, 21-27

[13] MUSMAN, S.; Turner, A.J. A Game Oriented Approach to Minimizing Cybersecurity Risk. *Saf. Secur. Stud.* **2018**, *27*

[14] Hong, J. B.; Kim, D. S.; Chung, C.-J.; Huang, D. A Survey on the Usability and Practical Applications of Graphical Security Models. *Comput. Sci. Rev.* **2017**, *26*, 1-16, doi:https://doi.org/10.1016/j.cosrev.2017.09.001

[15] Yi, S.; Peng, Y.; Xiong, Q.; Wang, T.; Dai, Z.; Gao, H.; Xu, J.; Wang, J.; Xu, L. Overview on Attack Graph Generation and Visualization

Technology. In Proceedings of the 2013 International Conference on Anti-Counterfeiting, Security and Identification (ASID); October 2013; pp. 1-6

[16] Bogdanoski, M.; Shuminoski, T.; Hadji-Janev, M.; Risteski, A.; Janevski, T. Future 5G Mobile Broadband Networks Using Cloud-Based Services with Advanced Security and QoS Framework. *Acta Polytech. Hung.* **2020**, *17*, 27-46, doi:10.12700/APH.17.10.2020.10.3

[17] Saulaiman, M. N.-E.; Kozlovszky, M.; Banati, A.; Csilling, Á. Use Cases of Attack Graph in Threat Analysis And Risk Assessment for The Automotive Domain. In Proceedings of the 2022 IEEE 1st International Conference on Cognitive Mobility (CogMob); October 2022; pp. 000085-000092

[18] Saulaiman, M.; Takács, M.; Kozlovszky, M.; Csilling, A. Fuzzy Model for Common Vulnerability Scoring System. In Proceedings of the 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI); May 2021; pp. 419-424

[19] Archiveddocs The STRIDE Threat Model Available online: https://learn.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20) (accessed on 7 January 2024)

[20] Shostack, A. *Threat Modeling: Designing for Security*; Wiley: Indianapolis, IN, 2014; ISBN 978-1-118-80999-0

[21] Loren, K.; Praerit, G. *The Threats to Our Products*; 1999

[22] Sommer, F.; Kriesten, R. Attack Path Generation Based on Attack and Penetration Testing Knowledge.; January 31 2023

[23] Plappert, C.; Zelle, D.; Gadacz, H.; Rieke, R.; Scheuermann, D.; Krauß, C. Attack Surface Assessment for Cybersecurity Engineering in the Automotive Domain.; IEEE, March 1 2021; pp. 266-275

[24] Angermeier, D.; Wester, H.; Beilke, K.; Hansch, G.; Eichler, J. Security Risk Assessments: Modeling and Risk Level Propagation. *ACM Trans. Cyber-Phys. Syst.* **2023**, *7*, 8:1-8:25, doi:10.1145/3569458

[25] Sommer, F.; Dürrwang, J. Comprehensive Collection of Automotive Security Attacks 2019