Leveraging Attack Graphs in Automotive Threat Analysis and Risk Assessment

Mera Nizam-Edden Saulaiman
BioTech Research Center, EKIK,
Óbuda University
Budapest, Hungary
Email:Mera.Abbassi@stud.uni-obuda.hu

Miklos Kozlovszky
BioTech Research Center, EKIK,
Óbuda University
Budapest, Hungary
Email:kozlovszky.Miklos@nik.uni-obuda.hu

Ákos Csilling
Academic Relations
Robert Bosch Kft.
Budapest, Hungary
Email:Akos.Csilling@hu.bosch.com

***Abstract*—With the increase in complexity of automotive network systems and the shift towards connected vehicles, cyber threats are constantly evolving, creating the need for advanced methodologies to assess and mitigate these threats and ensure the security of these systems. The ISO/SAE 21434 standard defines the Threat Analysis and Risk Assessment (TARA) methodology as a key activity for analyzing and assessing cybersecurity risks for a defined automotive system. In this paper, we introduce a Graph-based Attack Path Prioritization Tool (GAPP), which aims to introduce the concept of automation and address the limitations of manual TARA. GAPP automates the generation of attack paths, calculates the feasibility of each path, and identifies the most feasible attack paths within automotive networks. By providing a more dynamic, comprehensive, and automated means of analyzing network security, our approach aims to enhance TARA and offers a promising avenue for future research and development in the field of automotive cybersecurity.**

***Index Terms*—TARA, threat and risk analysis, automotive network, connected vehicles**

I. Introduction

In recent years, the automotive industry has witnessed a significant technological shift towards smart and connected vehicles that connect multiple embedded computers to form a complex advanced network [12].

The ISO/SAE 21434 standard [1] provides the technical basis for the cybersecurity engineering process of Electrical and Electronic (E \E) road vehicles and the requirements for cybersecurity management in the automotive industry. TARA is a core part of the security engineering process, which involves executing a comprehensive analysis, entailing the calculation of impact and attack feasibility values, leading to the derivation of the associated risk metrics. By implementing TARA, the automotive industry can proactively predict and identify potential security threats and vulnerabilities during the design phase, prioritize security measures, and ensure the safety and integrity of modern vehicles in the face of evolving cyber threats.

In our previous paper [8], we reviewed open-source attack analysis methodologies and frameworks from the IT domain and mapped their concepts to the automotive domain, highlighting that TARA is presently executed through manual effort by cybersecurity experts, a practice that has several inherent limitations and requires a significant amount of time and effort [9]. In [2] we proposed a generic model for automating the analysis and generation of attack paths within the TARA process. The objective is to seamlessly integrate this model into the TARA process, enhancing its efficacy in identifying potential threats.

In this study, we introduce the Graph-based Attack Path Prioritization Tool (GAPP), a tailored approach based on graphical modeling, leveraging the TARA methodology in alignment with the ISO/SAE 21434 standard. GAPP is designed to address security challenges specific to automotive systems with the primary objective of automating the generation of attack paths within a predefined network. By automating the analysis of attack paths and feasibility ratings, we utilize input data defined manually and employ an algorithm to calculate attack paths and their associated feasibility. Consequently, GAPP aims to provide an efficient means of assessing the security of modern automotive systems, capturing correlations between security events, and enabling quantitative reasoning for enhanced risk management in the ever-evolving landscape of automotive networks and connected vehicles.

The remainder of this paper is organized as follows. First, we provide a brief overview of modeling in security analysis in Section II. In Section III, the architecture and components of the GAPP tool are introduced. In Section IV, we conduct a comparative analysis between GAPP's results and the ISO TARA analysis to assess its effectiveness and efficiency. Finally, we conclude the paper and offer insights into future work in Section V.

II. Modelling in security analysis

Model-based security assessment methodologies offer a range of techniques for visual understanding and mapping of most likely threats. In threat modeling, different approaches and perspectives are used, which can be classified into three main categories:

Attacker-based: This approach revolves around understanding the motivations, capabilities, and strategies of potential attackers. This emphasizes how an attacker might target a system.

Asset based: Asset-based threat modeling begins with a focus on the critical assets or resources within a system. It aims to protect these assets by identifying threats that could target them. This is the approach used in TARA.

 ISBN: 978-1-68558-092-6

Vulnerability-based: This approach focuses on identifying and addressing vulnerabilities within a system, with a primary focus on weaknesses that could be exploited by attackers.

Furthermore, in terms of the structure, there are two main modelling categories: attack trees, and attack graphs.

Attack Trees provide a formal representation of potential attacks within a system [7]. In a hierarchical tree structure, the root represents the ultimate objective of the attacker. The branching paths from the leaves to the root symbolize the diverse strategies that an attacker might employ. [3].

In contrast to the tree structure, **Attack Graphs** are typically represented as Directed Acyclic Graphs (DAG) [6], and focus on vulnerabilities identified within a system. These graphs illustrate the interdependencies among the vulnerabilities of a system, providing a different perspective on system security [4].

With the GAPP methodology, we address a specific situation that often arises in automotive attack modeling. Each attack starts with an initial attack vector, which is one of several external interfaces to the system and continues through any number of internal interfaces connecting various internal subsystems, most often Electrical Control Units (ECUs), or even smaller components, such as firmware or data storage. The ultimate target is one of multiple security assets that require protection. The traditional methodology invites the analyst to draw up a list of all possible attack paths from all possible initial attack vectors to all assets and select those with the highest evaluated feasibility. However, for realistically complex systems, the number of possible combinations makes it practically impossible to perform a thorough analysis, and experts must rely on their expertise to find the most relevant attack paths. In the GAPP approach, the analyst only evaluates the direct attack steps from one subsystem to the next, and needs to consider only those directly connected to an internal interface. We assume that an attacker can combine multiple attacks in any sequence. Therefore, we evaluate all possible combinations that constitute an attack path. Fortunately, this part can be automated, so the construction of the actual paths, the evaluation of their feasibility, and ultimately, the ranking are fully automated using the GAPP tool.

III. GAPP Structure

The GAPP framework is designed to be easily defined and extendable, accommodating additional aspects that may emerge from various systems or scenarios. We followed the TARA process in ISO/SAE 21434, as discussed in our previous work [2]. The TARA process involves seven steps, each with a defined input and output, as shown in Fig. 1. In our current implementation in this study, GAPP addresses the attack path analysis and the attack feasibility rating steps of TARA, while the risk assessment and defense graph generation are currently out of scope.

The main inputs of GAPP are the list of assets, which become the nodes in the graph, their reachability via direct attack steps from one node to another, which become the edges on the graph, and the feasibility rating for each of these steps, which become weights on the edges. The main output of GAPP is the list of attack paths ranked by the combined feasibility of all steps along the path.

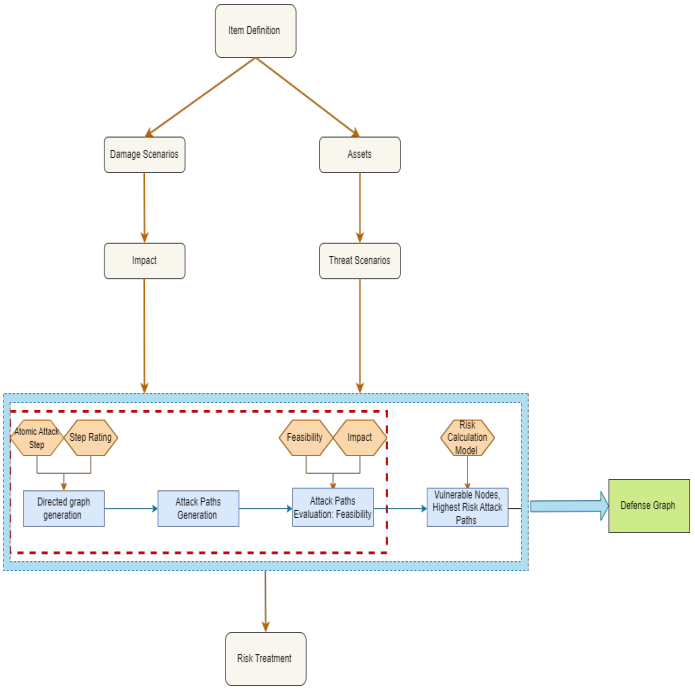

Fig. 1. Integration of the GAPP methodology into the TARA process, our current implementation is highlighted in the red rectangle.

In this section, we explain the essential concepts that serve as the foundation of GAPP and offer insights into the core principles of our methodology.

A. Attack steps

In the input layer, The GAPP tool primarily relies on the assets, network reachability, and attack steps. Assets are the output of the asset identification step in TARA, and include entities such as functions, data, and components that can affect the system and take part in a damage scenario. These assets are identified in the GAPP as nodes. These nodes are subsequently classified into distinct types: *Input nodes* represent external interfaces or attack entry points. *Internal nodes* are subsystems and components that might get compromised by the attacker. The *end nodes* represent the potentially targeted assets or vehicle functions.

Reachability refers to the potential ability of an attacker to traverse from one node to another with a single step. These atomic steps establish the pathways an attacker can take from an entry point (input node) to reach and potentially compromise the targeted assets (end nodes) within the system. The aggregation of these atomic steps from the input nodes to the end nodes signifies the potential routes that an attacker may take to gain control of individual assets.

B. Feasibility rating

The feasibility calculation is based on the attack potential method from the ISO/IEC 18045 [5], and includes these

criteria:

- **Elapsed time:** The duration required for an attacker to exploit the asset.
- **Specialist expertise:** The level of expertise an attacker would need to exploit the asset.
- **Knowledge of the item or component:** The amount of information an attacker would need about the asset to exploit it.
- **Window of opportunity:**The time frame during which an attacker could exploit the asset
- **Equipment:** The tools or resources an attacker would need to exploit the asset.

The ISO/SAE 21434 [1] defines the associated numeric values to the factors discussed previously, as shown in Fig. 2. The attack potential values are mapped to the Attack feasibility rating as shown in Fig. 3.

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								

Fig. 2. Attack potential values

Attack feasibility rating	Values
High	0 - 9
	10 - 13
Medium	14 - 19
Low	20 - 24
Very low	≥ 25

Fig. 3. Attack potential rating

In GAPP, a feasibility assessment is performed at the level of individual attack steps, rather than evaluating the entire attack path in a single assessment. This approach involves a granular examination of feasibility, in which each atomic attack step is individually assessed using a potential-based approach. Subsequently, these individual assessments are used to determine the overall feasibility of the complete attack path.

To calculate the overall feasibility of a path, we followed the maximum approach discussed in [10], in which we selected the maximum value per attack potential along the attack path. This concept has been partially used in the model described in [11].

In the context of the GAPP tool, let R denote the set of all tuples of attack feasibility factors. Function affmax : $R^* \rightarrow$ R represents a function that takes an arbitrary count $k \in \mathbb{N}$ of tuples of attack feasibility factors $r_1, \ldots, r_k$ as input and computes the maximum value for each attack feasibility factor. Specifically, employing an approach based on attack potentials as the attack feasibility factors in GAPP, the computation of affmax$[r_1, \ldots, r_k]$ is 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 $\max[v_{i,j}]$ refers to the maximum value of each attack feasibility factor j across the tuples $r_1, \ldots, r_k$.

C. Automated attack path generation

By automating this process, GAPP combines the defined atomic steps to construct comprehensive attack paths encompassing all possible communication routes that an attacker may follow. The primary objective of this automation is to eliminate the need for the manual enumeration of attack paths.

The GAPP tool performs two key tasks in its current process:

1) Enumeration of Attack Paths: The tool systematically enumerates all possible combinations of paths that lead from an input node to an asset or compromised function within the system.
2) Feasibility Calculation: GAPP calculates the combined feasibility along each enumerated path. This involves evaluating the feasibility of the individual attack steps and determining their cumulative impact on the overall path.

GAPP is currently implemented in python using standard libraries. So far no integration with other tool frameworks has been done. The principle of operation is described here.

- Directed Weighted Graph Generation: The GAPP tool creates a directed graph by connecting atomic attack steps, representing all possible paths. Entry points accessible to attackers are linked to starting nodes, and assets vulnerable to damage are the end nodes. Edges, including the virtual starting node, are weighted using feasibility ratings.
- Attack Path Analysis: The generated directed weighted graph is further analyzed to calculate the feasibility of each path by considering the weighted attack steps. These paths are computed starting from the starting nodes, leading to a comprehensive set of attack paths throughout the directed attack graph.

IV. EVALUATION

In this section we present the results of the evaluation of GAPP on the example system delineated in ISO 21434 [1], as this is a publicly available, well known basic example.

 ISBN: 978-1-68558-092-6

A. The system

The example contains a headlamp system designed to control the headlamp's operation based on the driver's input. In the high-beam mode, the system automatically switches to a low beam when it detects an oncoming vehicle and reverts to a high beam once the vehicle has passed. The system is connected to the gateway ECU, which in turn is linked to the navigation ECU through data communication. The navigation ECU has Bluetooth and cellular external communication interfaces, whereas the gateway ECU has an OBD-II interface. Fig. 4 shows a functional overview of the headlight system. It is assumed that both ECUs have security measures to prevent unauthorized data communication.

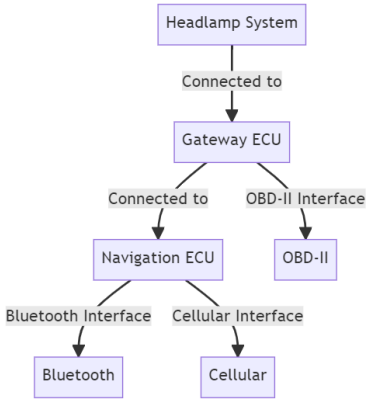

Fig. 4. Functional overview of the headlight system

B. Assets and damage scenarios

As shown in Fig. 1, the TARA process begins by identifying the assets and their damage scenarios. Assets in this system include data communication for lamp requests and oncoming car information as well as the firmware of the ECUs. Each asset is associated with its respective damage scenario and impact rating, which are out of scope in this study. These assets were then evaluated for the potential threat scenarios.

In the GAPP model, the external interfaces, assets and damage scenarios become the nodes of the graph, as shown in Table I.

TABLE I
GRAPH NODES IN GAPP

Nodes	Asset	Node type
Node 1	Physical access	Input node
Nodes 2	Bluetooth interface	Input node
Nodes 3	Cellular interface	Input node
Nodes 4	OBD port	Input node
Nodes 5	Navigation ECU	Internal Node
Nodes 6	GW ECU	Internal Node
Nodes 7	Data: DOS attack	End Node
Nodes 8	Data: spoofing the signal	End Node
Nodes 9	extract FW	End Node

C. Threat scenario and attack path analysis

The threat scenarios were identified for each damage scenario. As seen in Fig. 5 from the ISO standard [1], for each threat scenario, an attack path analysis is conducted to deduce all the possible paths that can lead to realizing the attack scenario.

Threat scenario	Attack path
Spoofing of a signal leads to loss of integrity of the data communication of the "Lamp Request" signal to the power switch actuator ECU, potentially causing the headlamp to turn off unintentionally	i. Attacker compromises navigation ECU from cellular interface. ii. Compromised navigation ECU transmits malicious control signals. iii. Gateway ECU forwards malicious signals to power switch actuator. iv. Malicious signals spoof the lamp request (OFF).
	i. Attacker compromises navigation ECU from Bluetooth interface. ii. Compromised navigation ECU transmits malicious control signals. iii. Gateway ECU forwards malicious signals to power switch actuator. iv. Malicious signals spoof the lamp request (OFF).
	i. Attacker gets local (see Table G.9) access to OBD connector. ii. Attacker sends malicious control signals from OBD connector. iii. Gateway ECU forwards malicious signals to power switch actuator. iv. Malicious signals spoof the lamp request (OFF).
Denial of service of oncoming car information	i. Attacker compromises navigation ECU from cellular interface. ii. Compromised navigation ECU transmits malicious control signals. iii. Gateway ECU forwards malicious signals to power switch actuator. iv. Attacker floods the communication bus with a large number of messages.
	i. Attacker attaches a Bluetooth-enabled OBD dongle to OBD connector when vehicle is parking unlocked. ii. Attacker compromises driver's smartphone with Bluetooth interface. iii. Attacker sends message via smartphone and Bluetooth dongle to Gateway ECU. iv. Gateway ECU forwards malicious signals to power switch actuator. v. Attacker floods the communication bus with a large number of messages.

Fig. 5. ISO Example attack paths for threat scenarios

With the GAPP approach, we only need to identify the individual steps that lead from one node to an other, leading to the attack steps discussed in the next subsection.

D. Attack steps and feasibility

We have applied the attack potential-based method to assess the feasibility of each atomic step in our attack paths. Table II presents the specific attack steps and their corresponding attack potential. Although this approach may not fully reflect the realistic risk and feasibility, it serves our research purposes and provides a more granular understanding of attack scenarios.

E. Attack Graph

GAPP generates the directed attack graph as shown in Fig. 6, representing all the possible paths from the attacker entry points that represent the start nodes.

F. Attack paths

In 7, GAPP utilizes system interconnections to generate distinct attack paths, resulting in 4 paths for the first attack scenario and 4 paths for the second attack scenario. Interactions between steps are considered, and each path is color-coded from start to end, enabling visualization and analysis.

 ISBN: 978-1-68558-092-6

TABLE II
ATTACK STEPS AND ATTACK POTENTIAL IN GAPP

Attack step	Edge	Attack potential				
		ET	SE	KoIC	WoO	EQ
Compromise OBD through physical access	1→4	1	6	7	10	4
Compromise navigation ECU from BLE Interface	2→5	1	6	5	4	4
Compromise navigation ECU from cellular interface	3→5	1	6	5	4	4
Compromise navigation ECU from OBD Interface	4→5	1	6	5	4	4
Compromise the GW to send malicious messages	5→6	1	6	7	4	4
Extract FW from GW	6→9	1	6	11	7	4
DOS of oncoming car information	6→7	1	3	7	4	4
Signal spoof of Headlamp data	6→8	1	3	7	4	4

TABLE III
ATTACK PATH GAPP

Attack Path	Max Attack potential					
	ET	SE	KoIC	WoO	EQ	Rating
[1→4→5→6→9]	1	6	11	10	4	low
[1→4→5→6→8]	1	6	7	10	4	low
[1→4→5→6→7]	1	6	7	10	4	low
[2→5 →6 →9]	1	6	7	4	4	Medium
[2→5→6 →8]	1	6	7	4	4	Medium
[2→5 →6 →7]	1	6	7	4	4	Medium
[3→5→6→9]	1	6	7	4	4	Medium
[3→5→6→8]	1	6	7	4	4	Medium
[3→5→6→7]	1	6	7	4	4	Medium

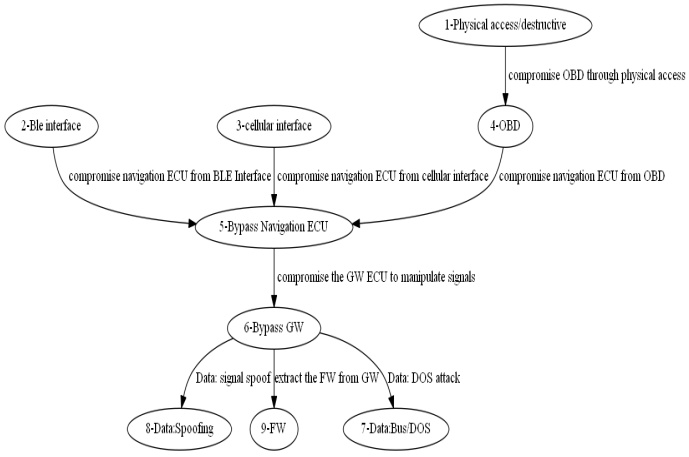

Fig. 6. GAPP Attack graph

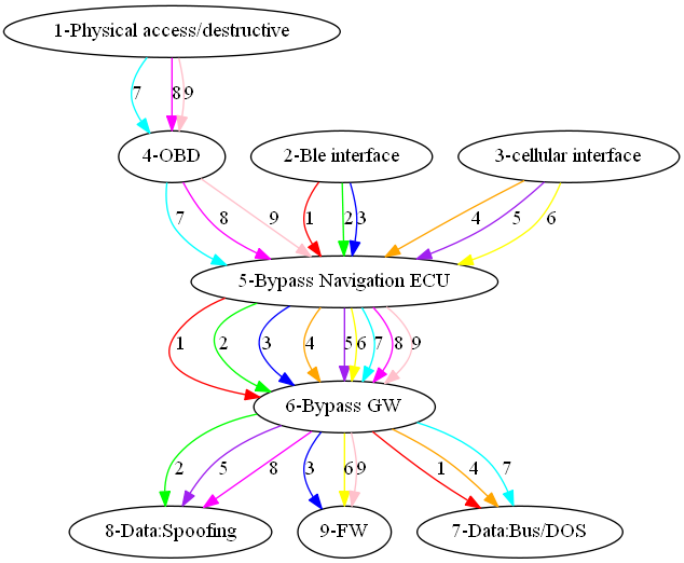

Fig. 7. GAPP attack graph highlighted path

G. Assessment

Table IV compares the attack paths generated by ISO and GAPP.

The evaluation of the GAPP tool in comparison with the ISO/SAE 21434 [1] standard involves assessing three main categories: attack path identification, feasibility assessment, and coverage and completeness.

In the attack path identification category, we compared the attack paths identified by GAPP to those mentioned in the ISO standard. We analyzed the sequence of steps and nodes involved and identified variations in the attack scenarios or paths in both analyses. GAPP provided more attack paths for each scenario, indicating its ability to capture a more comprehensive range of potential attack routes.

Next, in the feasibility assessment category, we compared the feasibility ratings assigned to the attack paths in GAPP with those provided in the ISO standard. Using averaging for each attack step and the same additional method as in the potential approach, GAPP yielded more accurate data. However, there is room for improvement to further enhance the accuracy. Finally, in the coverage and completeness category, we evaluated the coverage of attack scenarios and paths in GAPP compared to the ISO standard. We found that GAPP successfully covered the entire attack path and scenario, demonstrating its ability to encompass all relevant attack vectors and scenarios mentioned in the standard.

V. CONCLUSION & FUTURE WORK

The main benefit of our approach and the GAPP tool is to provide scalability for the identification of the most relevant attack paths in large systems. Instead of manually constructing and evaluating all possible paths, the tool only requires the manual evaluation of individual steps. The combination and ranking of these paths is performed by the tool, which provides

 ISBN: 978-1-68558-092-6

a list of the highest-ranking attack paths. Security engineers can concentrate on these issues in their TARA.

This is a small step in the full TARA process; nevertheless, the GAPP tool lays the foundation for further advancements in automotive cybersecurity. As an automated and efficient approach to attack path analysis, GAPP opens possibilities for future research and development. Here are some areas of future work to consider.

The current approach uses a crude combination of the feasibility rating of each individual step into a rating for the full path. Future work can focus on refining the feasibility assessment in GAPP by exploring alternative methods for calculating the feasibility values and considering more factors in the assessment process. Today, we recommend to set a wide cut-off value, and evaluate the edge cases in detail. A more refined formula could increase confidence in the evaluation and save effort.

The tool can be enhanced by integrating real-world data and real attack scenarios to provide more accurate and realistic results. In this study we used the publicly available ISO example to provide a comparison with an accepted standard evaluation. While similar studies on real products are surely confidential, a more extensive study could construct a larger imaginary example and provide more data for comparison.

In an industrial environment the GAPP tool would be integrated into the existing TARA framework, and reuse already evaluated attack steps from previous projects.

Enhanced Visualization: Improving the visualization capabilities of GAPP can help users better understand and interpret the generated attack paths and feasibility ratings.

TABLE IV
COMPARISON OF ATTACK PATHS BETWEEN THE MANUAL ANALYSIS IN [1] AND THE AUTOMATIC ANALYSIS WITH THE GAPP MODEL.

GAPP Attack Path	Feasibility	ISO attack Path	Feasibility
[1→4→5→6→8]	Low	[1→4→5→6→8]	Low
[2→5→6 →8]	Medium	[2→5→6 →8]	Medium
[3→5→6→8]	High	[3→5→6→8]	Medium
[1→4→5→6→7	Low	[1→4→5→6→7]	Low
[3→5→6→7]	Low	[3→5→6→7]	Medium
[2→5 →6 →7]	Medium	none	-

REFERENCES

[1] *Road vehicles – Cybersecurity engineering*, ISO/SAE 21434, 2021

[2] M. N. -E. Saulaiman, M. Kozlovszky, A. Banati and Á. Csilling, "Use Cases of Attack Graph in Threat Analysis And Risk Assessment for The Automotive Domain," 2022 IEEE 1st International Conference on Cognitive Mobility (CogMob), Budapest, Hungary, 2022, pp. 000085-000092, doi: 10.1109/CogMob55547.2022.10118297.

[3] V. Saini, Q. Duan, and V. Paruchuri, 'Threat Modeling Using Attack Trees', Journal of Computing Sciences in Colleges, vol. 23, 04 2008.

[4] A. Bánáti, E. Rigó, R. Fleiner and E. Kail, "Use cases of attack graph for SOC optimization purpose," 2022 IEEE 26th International Conference on Intelligent Engineering Systems (INES), Georgioupolis Chania, Greece, 2022, pp. 000143-000148, doi: 10.1109/INES56734.2022.9922617.

[5] Information security, cybersecurity and privacy protection — Evaluation criteria for IT security — Methodology for IT security evaluation, ISO/IEC 18045, 2022, Available: https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

[6] J. B. Hong, D. S. Kim, C.-J. Chung, and D. Huang, 'A survey on the usability and practical applications of Graphical Security Models', Computer Science Review, vol. 26, pp. 1–16, 2017.

[7] K. Edge, 'A Framework For Analyzing And Mitigating The Vulnerabilities Of Complex Systems Via Attack And Protection Trees', p. 219, 07 2007.

[8] M. N.-E. Saulaiman, M. Kozlovszky, Á. Csilling, A. Banati and A. Benhamida, "Overview of Attack Graph Generation For Automotive Systems," 2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC), Reykjavík, Iceland, 2022, pp. 000135-000142, doi: 10.1109/ICCC202255925.2022.9922866.

[9] D. Zelle, C. Plappert, R. Rieke, D. Scheuermann, and C. Krauß, 'ThreatSurf: A method for automated Threat Surface assessment in automotive cybersecurity engineering', Microprocessors and Microsystems, vol. 90, p. 104461, 2022.

[10] C. Plappert, D. Zelle, H. Gadacz, R. Rieke, D. Scheuermann, and C. Krauß, 'Attack Surface Assessment for Cybersecurity Engineering in the Automotive Domain', in 2021 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2021, pp. 266–275.

[11] D. Angermeier, H. Wester, K. Beilke, G. Hansch, and J. Eichler, 'Security Risk Assessments: Modeling and Risk Level Propagation', ACM Trans. Cyber-Phys. Syst., vol. 7, no. 1, Feb. 2023

[12] Mundhenk, P. "Security for Automotive Electrical/Electronic (E/E) Architectures;" Technical University of Munich: Munich, Germany, 2017.

 ISBN: 978-1-68558-092-6