AuSSE: A Novel Framework for Security and Safety Evaluation for Autonomous Vehicles

1st Nhung H. Nguyen
School of EECS
The University of Queensland
Brisbane, Australia
https://orcid.org/0000-0001-8607-2299

2nd Jin-Hee Cho
Department of Computer Science
Virginia Tech
Falls Church, VA, USA
https://orcid.org/0000-0002-5908-4662

3rd Terrence J. Moore
Network Science Division
US Army Research Laboratory
Falls Church, VA, USA
https://orcid.org/0000-0003-3279-2965

4th Seunghyun Yoon
Department of Energy Engineering
Korea Institute of Energy Technology
Falls Church, VA, USA
https://orcid.org/0000-0001-6264-976X

5th Hyuk Lim
AI Graduate School
Korea Institute of Energy Technology
Naju-si, South Korea
https://orcid.org/0000-0002-9926-3913

6th Frederica Nelson
Network Science Division
US Army Research Laboratory
Falls Church, VA, USA
https://orcid.org/0000-0001-8641-384X

7th Guangdong Bai
School of EECS
The University of Queensland
Brisbane, Australia
https://orcid.org/0000-0002-6390-9890

8th Dan Dongseong Kim
School of EECS
The University of Queensland
Brisbane, Australia
https://orcid.org/0000-0003-2605-187X

***Abstract*—Autonomous vehicles (AV) are becoming increasingly efficient and equipped with advanced technologies and connectivity. These include over-the-air software updates, connecting telematics data, and software-defined vehicles. However, these advancements also introduce new potential attack surfaces and increase AV security and safety risks. A critical challenge arises: How can we assess cyber attack impact on the operational safety of AV? To address this challenge, this paper proposes a novel framework named AuSSE (Autonomous Vehicle Security and Safety Evaluation).**

We developed a novel graphical security model called VHARM to identify attack paths and assess security risks. Additionally, we created CARLASec, an extension to the existing CARLA simulator. CARLASec enables the handling of in-vehicle networks, protocols, cyber attack injection mechanisms, and safety scores. As a result, it is suitable for a wider range of research on AV attack and defence. Additionally, recognising that AV security and safety cannot be separated, we introduce methods for comprehensive security and safety evaluations.

***Index Terms*—autonomous vehicle, autonomous vehicle attack, CARLA simulator, safety evaluation, security model.**

I. Introduction

With the advent of AV, in-vehicle networks can be divided into two distinct yet interconnected domains: the information domain and the automotive domain. The information domain includes various systems, such as infotainment, telematics, and Advanced Driver Assistance Systems (ADAS). These systems are powered by operating systems (OS) like Automotive Grade Linux (AGL), BlackBerry QNX, and Android Automotive. They also employ common communication technologies, including Wi-Fi, Cellular, Bluetooth, and USB ports [1]. On the other hand, the automotive domain focuses on the vehicle's core functional systems. This includes critical components, such as body control, engine control, airbags, and Anti-lock Braking Systems (ABS), interconnected through protocols like CAN, LIN, MOST, FlexRay, and Ethernet.

Traditional cyberattacks mainly targeted the automotive domain via vehicle protocols, notably the CAN protocol, which lacked essential security features [2]. Early research, Koscher (2010) [3] demonstrated how sniffing and injecting malicious CAN messages could manipulate automotive systems, such as unlocking doors, disabling brakes, and altering the speedometer display. With the advancement of AV and their information systems, the attack surface has increased significantly. Studies and reports on AV attack [4]–[10] have demonstrated how attackers can get into the system via multiple attack surfaces to gain complete control over the automotive domain of different vehicles, including Jeep Cherokee, Tesla, BMW, and Mercedes Benz.

These studies highlight significant security challenges in AV and present a research problem: How to assess the security of the combination of both information and automotive domains in the presence of real-world cyber attacks. In particular, the widely used golden triad - Confidentiality, Integrity, and Availability (CIA) - serves as the main attribute in traditional information system security. Moreover, when transitioning from pure information systems to physical systems like AV, an additional critical factor needs to be considered: Safety. This poses another research problem: How does the traditional CIA relate to safety in self-driving cars, and how does a cyber

2833-292X/24/$31.00 ©2024 IEEE
DOI 10.1109/DSN-S60304.2024.00012

attack affect the safety of an AV?

Conducting real-world testing for AV can be highly risky, costly, and sometimes impractical, especially when it involves scenarios with cyber attacks. Therefore, the analytical model and simulation approach are promising in this domain. Several studies [11]–[17] employed different techniques on AV security modelling, including qualitative and quantitative methods. However, they have not sufficiently explored the escalation of real-world vulnerabilities into automotive attacks, nor have they provided adequate methods to assess the impact of cyber attacks on AV safety.

To address these research problems, we propose AuSSE, a novel framework that combines analytical modelling and simulation to assess cyber attack risks on AV. By extending the CARLA simulation [18], AuSSE allows safety testing under various attack scenarios and profiles, thereby offering the analysis of cyber threats' impacts on AV safety.

The key contributions of this paper are as follows:

(1) We introduce a comprehensive framework for quantitatively evaluating the security and safety of AV. This framework is the first effort to explore the relationship between security and the safety attributes of AV using simulation techniques. The framework comprises two innovative components: VHARM and CARLASec. VHARM is utilised to identify potential attack vectors, evaluate security risks, and measure the effectiveness of cyber mitigation strategies. Meanwhile, CARLASec enables the simulation to assess the impact of cyber attacks on the safety of AVs.

(2) We contribute to the CARLA simulation environment by developing CARLASec. CARLASec enhances CARLA with the in-vehicle network architecture, as well as two key protocols: Controller Area Network (CAN) and Unified Diagnostic Services (UDS), enabling the simulation of various automotive cyber-attacks and the computation of safety metrics.

The structure of this paper is as follows: Section II provides an overview of our proposed framework. Section III outlines our approach to defining the AV networks and attacker profile, as well as discusses our vehicle security model, VHARM. Section IV describes our work on extending the CARLA simulator and calculating the safety score. The evaluation of the framework is presented in Section V. Section VI points out the limitations of our work and provides discussions on future directions. Finally, Section VII concludes the paper.

II. An Overview of Proposed Framework

In this section, we present the overall design of our proposed framework - AuSSE. AuSSE aims to offer the following main capabilities. 1) It provides a structured method for formally defining AV networks and attack behaviours. 2) It defines the security model, enabling the identification and visualization of all potential attack paths. 3) It offers a method and simulation environment to evaluate the safety impact of cyber attacks on AV systems. 4) It introduces formal security and safety metrics to assess the overall security and safety score of AV systems. 5) It can determine the most effective mitigation strategies associated with cyber threats.

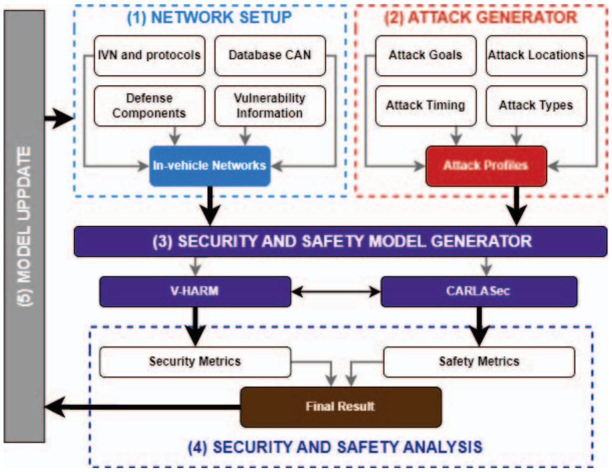

Fig. 1: The architecture of AuSSE framework

As shown in Figure 1, AuSSE comprises five phases:

(1) Network setup: This initial phase involves defining the components, defence mechanisms, and topology of the AV network. This information is then stored in a JSON format, for subsequent phases. Additionally, this phase includes defining the structure of CAN messages and identifying vulnerabilities in each component. Given the similarities between AV and computer networks, vulnerability scanners can be used to identify potential vulnerabilities.

(2) Attack generator: The attack profile is defined in this phase to configure the attack generator for subsequent processes. Further details on AV network and Attack profile definition are available in Section III.

(3) Security and safety model generator: This extensive phase includes two modules, VHARM and CARLASec, for assessing the AV system's security and safety. Inputs such as Network Information in JSON format, the CAN message file, and the Attack Profile are used to generate the Security model and set up the simulation environment. VHARM and CARLASec are detailed in Section III and Section IV, respectively.

(4) Security and safety analysis: In this phase, the metrics poll is used to automatically calculate the security score and safety score of the entire system.

(5) Model update: After analysis, the security decision-maker can update the defence components in the Network Setup and/or the Attack Profile. The framework can then be rerun until the desired system security and safety score is achieved. This approach, therefore, allows for continuous improvement and assessment of the effectiveness of various security strategies and the risk of different attack profiles.

III. Definition and The vehicle security model

This section outlines our definitions of the AV network's architecture and the attack profiles that could be employed by attackers. Additionally, we present our security model, namely VHARM, within this context.

A. The AV Networks

The AV In-Vehicle Network (IVN) is defined by $IVN = (ECU, BUS, VUL)$. Specifically, this network comprises a finite set of Electronic Control Units (ECU), a finite set of connecting buses (BUS) that link these ECUs, and a finite set of vulnerabilities (VUL).

B. Attack Profiles

The attack profile, denoted as AP, has a critical role in our framework. It is used for defining the potential behaviour of an attacker and serving as a basis for automatically generating attacks. A malicious actor may employ one or several attack profiles.

Each profile $ap \in AP$ comprises four elements: a goal ap_{goal}, which typically is the specific components within the IVN; a location to the goal $ap_{loc} \in \{Network, Adjacent, Local, Physical\}$; an attack time $ap_{time} = (t_{start}, t_{duration}, t_{wait}, t_{interval})$, which specifies the start time of the attack (t_{start}), the duration of the attack ($t_{duration}$), the pause period before the next attack (t_{wait}), and the frequency of the attacks ($t_{interval}$); and a finite set of attack types ap_{attype}.

C. VHARM - The Vehicle Security Model

VHARM builds upon the principles of the original Hierarchical Attack Represent Model (HARM) [19] with a new definition of AV domains.

VHARM takes parameters of the in-vehicle networks alongside attacker profiles to automatically generate visual graphs. Furthermore, VHARM introduces metrics to quantitatively assess the overall security of the system. The structure of VHARM is comprised of three distinct layers, each providing a unique perspective on system vulnerabilities and potential attack vectors. The upper layer employs an attack graph to outline potential routes an attacker might use to breach the AV, moving from one ECU to another. The middle layer offers an in-depth view of these attack paths by breaking down the ECUs into their components. At the lower layer, the model employs an Attack Countermeasure Tree (ACT) [20] to show vulnerability information associated with each component identified in the middle layer. Each ACT allows the calculation of the likelihood of a successful attack. This calculation considers various factors, including the probability of vulnerability exploitation, the chances of detection, and the efficacy of mitigation strategies.

By using a combination of three attack graphs and an attack tree, VHARM enables a comprehensive evaluation of security risks. It allows for the calculation of attack routes, the number of potential attack paths, and the assessment of security risks at the component, ECU, and pathway levels, leading up to an overarching system-level security score.

IV. CARLASec: The Vehicle Attack Simulator

This section discusses the new features of CARLASec and how we use it to assess the safety impacts that arise from cyber-attacks.

CARLA has primarily focused on artificial intelligence (AI) and autonomous driving research, without the support of simulating critical technical components of actual vehicles by default. CARLASec enhances the functionality of CARLA by adding extra modules that include an in-vehicle network (IVN) with CAN and UDS protocols, as well as a new metric for evaluating safety scores. These enhancements enable the simulation of cyber-attack scenarios, the observation of vehicle behaviours in such situations, and the automated measurement of safety impacts.

A. In-vehicle Network in CARLASec

To incorporate the IVN modules into CARLASec, we present a model of an IVN detailed in Section III, represented as a JSON file. Upon loading this JSON file, the system dynamically creates objects that reflect the components defined within the IVN. These objects are not only descriptive but also functional, allowing us to simulate their corresponding behaviour during the simulation.

B. Protocols in CARLASec

In CARLASec, we employ a Database CAN (DBC) file to simulate the transmission of CAN messages. The DBC file provides a structured way for simulated vehicles to accurately interpret and respond to CAN messages. To support the processing and handling of CAN messages, our implementation incorporates two Python libraries – python-can [21] and cantools [22], along with the can-utils tools [23]. To mimic the real operational environment of a CAN network in CARLASec, we have configured two distinct CAN buses, each connected to the Central Gateway. One bus is designated for critical functions while the other handles non-critical functions. Furthermore, we have implemented a mechanism for setting CAN Identifier (CAN ID) priorities and the broadcast nature of CAN messages.

For handling UDS messages, we used the udsoncan [24] and isotp [25] Python libraries. Using these resources, we define six UDS services and establish a UDS Server on critical ECUs and a UDS Client on the infotainment system. As a result, the infotainment system becomes a local diagnostic tester and can send diagnostic commands to control critical vehicle functions.

With these fundamental extensions for the security domain, CARLASec enables the simulation of five types of main automotive attacks against AV, namely, CAN DoS, CAN Fuzzing, CAN replay, UDS fuzzing, and UDS replay.

C. Driving Safety Rating in CARLASec

To evaluate the potential impact of a cyber attack on the operation of a car, we introduce a metric called Driving Safety Rating (DSR) to rate driving safety. DSR builds upon the concept of the Driving Score found in the Carla Leaderboard [26], which is primarily used for ranking the performance of autonomous driving algorithms in competitions. The original Driving Score on the Carla Leaderboard does not take into account the severity of collisions, which can lead to misleading results, especially in cases where critical collisions occur. Therefore, we enhance the original metric by incorporating Collision Intensity while retaining the other components.

Equation 1 presents our DSR metric, which is computed based on two fundamental factors: Driving Behavior (DB) and Collision Severity (CS). The DB includes events such as opening doors while running, running red lights, ignoring stop signs, and off-road incidents with the CARLA vehicle. The CS is determined by the Collision Type (CT) and Collision Intensity (CI), where CT includes three types: collision with a pedestrian, collision with a vehicle, and collision with others (including buildings, walls, and static objects), and CI represents the collision intensity for each type. RC presents the average percentage of routes completed.

$$DSR = \prod_{i=1}^{n} p_i^{DB} \times \prod_{j=1}^{m} \frac{p_j^{CT}}{CI_j} \times RC \tag{1}$$

Here, n represents the total number of driving behaviour infractions in a single run, m stands for the total number of collisions in a single run, p_i is the penalty value assigned to each driving behaviour infraction and p_j is the penalty value assigned to each collision. The value of p_i and p_j is obtained from the CARLA leaderboard.

V. Evaluation

For the demonstration purpose, we focus on a specific demonstration scenario involving a BMW car, as documented by Cai (2019) [10] and Jungebloud (2023) [17]. The scenario illustrates an attack where the attacker gains root access to both the head unit and the telematics unit (TU), enabling them to send CAN and UDS messages and control various car functions. We developed a VHARM model based on the described IVN and attack profiles, leading to a preliminary quantitative analysis. This analysis resulted in 10 attack paths and a high-security risk score for the entire system.

Further, we simulate the most critical attack path identified by the VHARM analysis using CARLASec. This path begins when the attacker gains root access to the TU after exploiting 2 vulnerabilities CVE-2018-9311 (v_1) and CVE-2018-9318 (v_2). Subsequently, the attacker targets three specific ECUs to manipulate the vehicle: the Electrical Digital Motor Electronics (EDME) for throttle control, Electronic Power Steering (EPS) for steer and the Body Control Unit (BDC) for door control.

To evaluate the safety impact of different attack types on these three ECUs, we develop three attack profiles:

$$ap_1 = (EDME, \text{Adjacent}, (0,5,5,5), \{CR, CF, UR, UF\})$$
$$ap_2 = (EPS, \text{Adjacent}, (0,5,5,5), \{CR, CF, UR, UF\})$$
$$ap_3 = (BDC, \text{Adjacent}, (0,5,5,5), \{CR, CF, UR, UF\})$$

These profiles target the manipulation of the EDME, EPS and BDC with fixed timing: starting attack at vehicle start-up for 5 seconds, followed by a 5-second pause and repeated it five times. The attack methods deployed are CAN Replay (CR), CAN Fuzzing (CF), UDS Replay (UR) and UDS Fuzzing (UF). In the Can Replay and UDS Replay scenarios, it is assumed the attacker can sniff and replay the correct message to control the ECUs. For Can Fuzzing, the attacker exploits prior knowledge of the CAN ID from replay attacks targeting throttle, steering, and door controls and only fuzzes the data frame (8 bytes). UDS Replay and UDS Fuzzing involve using Routine Control Services to manipulate throttle, steering, and door functions. These attacks presuppose that the attacker has successfully generated accurate seeds to bypass the Security Assess service and change the Diagnostic Session Control Service. In UDS Fuzzing, the Service ID, Routine Control Type and Routine Identifier remain constant, only the routine status data (1 byte) is varied. Given the attacker has already obtained root privileges of the TU, the attack location is categorized as 'Adjacent'.

We conducted three attack profiles in a controlled environment using CARLAsec, specifically Map 10, with 12 vehicles and 8 pedestrians involved. The targeted vehicle followed a predetermined route across all scenarios. After running these simulations, we calculated DSR using Equation 1 and performed a comparative analysis of these scores, as shown in Figure 2. A score of 100 indicates completely safe driving conditions, with no risks, collisions, or violations of driving behaviour. Conversely, a score of 0 represents highly unsafe driving, characterized by frequent collisions or significant violations of driving norms.

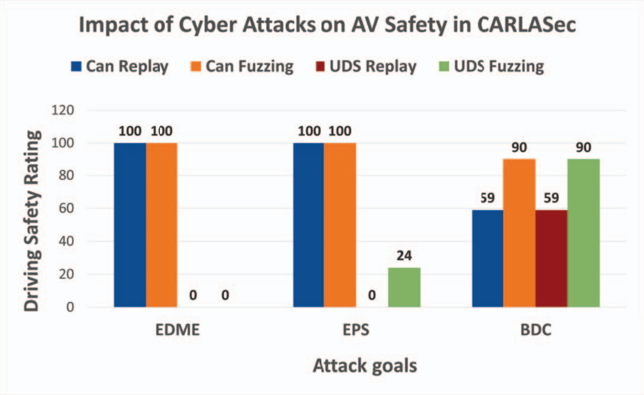

Fig. 2: The variation in Driving Safety Rating for different attack goals and attack types.

Figure 2 illustrates the impact of cyber attacks on AV safety is significant. The results indicate that the CAN Replay and CAN Fuzzing do not affect AV safety, since the gateway restricts transmission across different CAN bus systems. This design significantly enhances the safety of AVs during cyber attacks. However, the diagnostic functions can pose a significant threat to AV safety if used maliciously, such as through UDS attacks. Furthermore, our analysis indicates that the impact on driving safety can vary depending on the specific types and goals of automotive attacks, even when the attacker exploits the same security vulnerabilities (v_1 and v_2). Importantly, the evaluation demonstrates that AuSSE framework is effective in analysing and evaluating the security and safety of AV under cyber attack conditions.

VI. Limitations and Future works

This section outlines the limitations of our current approach and proposes possible enhancements for future research.

At present, AuSSE is limited to scenarios with a single attacker. However, real-world situations often involve multiple attackers collaborating against a single target, as seen in complex diversionary attacks. Therefore, our future research aims to extend the attack profile to capture scenarios with multiple attackers. Additionally, there is currently a lack of integration between VHARM and CARLASec. VHARM only provides static, initial security assessments, while CARLASec offers a more dynamic approach by adapting to the attacker's actions and environmental changes. To improve this, our goal is to enhance our framework by developing a Temporary VHARM that can smoothly integrate into the dynamic phases of CARLASec. Lastly, our current analysis relies on a single safety metric to determine the vehicle safety state during cyber attacks. To provide a more comprehensive evaluation of cyber threats across both security and safety domains, we aim to develop a suite of safety metrics that can be integrated with existing security metrics.

VII. Conclusions

In this paper, we present a new framework for evaluating the security and safety of vehicles. Our framework combines VHARM for security score assessment and CARLASec for simulating and assessing the safety of AV during cyber attacks. The analytical viewpoint of our methods does not yet cover all real-life cyber attacks, as we use a simulator platform and operate under its simplified constraints. However, we propose a novel perspective on this matter, providing researchers, car manufacturers or security decision-makers with new approaches and tools to evaluate AV security and understand how cyber attacks can impact the driving safety of the car.

References

[1] C. Miller and C. Valasek, "A Survey of Remote Automotive Attack Surfaces," *Defcon 22*, pp. 1–90, 2014. [Online]. Available: http://illmatics.com/remote%20attack%20surfaces.pdf.

[2] M. Bozdal, M. Samie, S. Aslam, and I. Jennions, "Evaluation of can bus security challenges," *Sensors (Switzerland)*, vol. 20, no. 8, 2020, ISSN: 14248220. DOI: 10.3390/s20082364.

[3] K. Koscher, A. Czeskis, F. Roesner, *et al.*, "Experimental security analysis of a modern automobile," *Proceedings - IEEE Symposium on Security and Privacy*, pp. 447–462, 2010, ISSN: 10816011. DOI: 10.1109/SP.2010.34.

[4] S. Checkoway, D. McCoy, B. Kantor, *et al.*, "Comprehensive experimental analyses of automotive attack surfaces," *Proceedings of the 20th USENIX Security Symposium*, pp. 77–92, 2011.

[5] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Defcon 23*, vol. 2015, pp. 1–91, 2015. [Online]. Available: http://illmatics.com/Remote%20Car%20Hacking.pdf.

[6] S. Nie, L. Liu, and Y. Du, "Free-fall: hacking tesla from wireless to can bus," *Defcon*, pp. 1–16, 2017.

[7] S. Nie, L. Liu, Y. Du, and W. Zhang, "Over-The-Air: How we remotely compromised the gateway, BCM, and Autopilot ECUs of TESLA cars," *BlackHat USA 2018*, vol. 1, pp. 1–19, 2018.

[8] Tencent Keen Security Lab, "Experimental Security Assessment of BMW Cars: A Summary Report," *Keen Security Lab*, p. 26, 2018.

[9] Tencent Keen Security Lab, "Mercedes-Benz MBUX Security Research Report," 2021.

[10] Z. Cai, A. Wang, W. Zhang, M. Gruffke, and H. Schweppe, "0-days mitigations: Roadways to exploit and secure connected BMW cars," *Black Hat USA*, vol. 2019, p. 39, 2019.

[11] A. R. Ruddle, H. Mira, and S. Information, "EVITA - Security requirements for automotive on-board networks based on dark-side scenarios . intrusion protected applications," no. February 2016, 2009.

[12] J. P. Monteuuis, A. Boudguiga, J. Zhang, H. Labiod, A. Servel, and P. Urien, "SarA: Security automotive risk analysis method," *CPSS 2018 - Proceedings of the 4th ACM Workshop on Cyber-Physical System Security, Co-located with ASIA CCS 2018*, no. May, pp. 3–14, 2018. DOI: 10.1145/3198458.3198465.

[13] A. Behfarnia and A. Eslami, "Risk Assessment of Autonomous Vehicles Using Bayesian Defense Graphs," *IEEE Vehicular Technology Conference*, vol. 2018-Augus, 2018, ISSN: 15502252. DOI: 10.1109/VTCFall.2018.8690732. arXiv: 1903.02034.

[14] K. Karray, J. L. Danger, S. Guilley, and M. Abdelaziz Elaabid, "Attack tree construction and its application to the connected vehicle," *Cyber-Physical Systems Security*, pp. 175–190, 2018. DOI: 10.1007/978-3-319-98935-8_9.

[15] Z. Petho, I. Khan, and Á. Torok, "Analysis of Security Vulnerability Levels of In-Vehicle Network Topologies Applying Graph Representations," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 37, no. 5-6, pp. 613–621, 2021, ISSN: 15730727. DOI: 10.1007/s10836-021-05973-x. [Online]. Available: https://doi.org/10.1007/s10836-021-05973-x.

[16] S. Khalid Khan, N. Shiwakoti, and P. Stasinopoulos, "A conceptual system dynamics model for cybersecurity assessment of connected and autonomous vehicles," *Accident Analysis and Prevention*, vol. 165, no. December 2021, p. 106 515, 2022, ISSN: 00014575. DOI: 10.1016/j.aap.2021.106515. [Online]. Available: https://doi.org/10.1016/j.aap.2021.106515.

[17] T. Jungebloud, N. H. Nguyen, D. S. Kim, and A. Zimmermann, "Hierarchical Model-Based Cybersecurity Risk Assessment During System Design," *38TH International Conference on ICT Systems Security and Privacy Protection (IFIPSEC 2023)*, 2023.

[18] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[19] J. Hong and D. S. Kim, "HARMs: Hierarchical attack representation models for network security analysis," *Proceedings of the 10th Australian Information Security Management Conference, AISM 2012*, pp. 74–81, 2012. DOI: 10.4225/75/57b559a3cd8da.

[20] A. Roy, D. S. Kim, and K. S. Trivedi, "Poster abstract: ACT: Attack countermeasure trees for information assurance analysis," *Proceedings - IEEE INFOCOM*, pp. 4–5, 2010, ISSN: 0743166X. DOI: 10.1109/INFCOMW.2010.5466633.

[21] *Python-can 4.3.1 documentation*. [Online]. Available: https://python-can.readthedocs.io/en/stable/.

[22] *Can bus tools — cantools 39.4.3.dev10+gcc02988 documentation*. [Online]. Available: https://cantools.readthedocs.io/en/latest/.

[23] *Linux-can/can-utils: Linux-can / socketcan user space applications*. [Online]. Available: https://github.com/linux-can/can-utils.

[24] *Python implementation of uds standard (iso-14229) — udsoncan 1.21 documentation*. [Online]. Available: https://udsoncan.readthedocs.io/en/latest/.

[25] *Isotp 2.0 documentation*. [Online]. Available: https://can-isotp.readthedocs.io/en/latest/isotp/examples.html.

[26] *Carla autonomous driving leaderboard*. [Online]. Available: https://leaderboard.carla.org/.