Revisiting Automotive Attack Surfaces: a Practitioners' Perspective

Pengfei Jing*, Zhiqiang Cai†, Yingjie Cao*, Le Yu*, Yuefeng Du†, Wenkai Zhang†
Chenxiong Qian‡, Xiapu Luo*§, Sen Nie†, Shi Wu†
*Department of Computing, The Hong Kong Polytechnic University
†Keen Security Lab, Tencent
‡Department of Computer Science, University of Hong Kong

2024 IEEE Symposium on Security and Privacy (SP) | 979-8-3503-3130-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/SP54263.2024.00080

***Abstract*—As modern vehicles become increasingly complex in terms of both external attack surfaces and internal in-vehicle network (IVN) topology, ensuring their cybersecurity remains a challenge. Existing standards and regulations, such as WP29 R155e and ISO 21434, attempt to establish a baseline for automotive cybersecurity, but their sufficiency in addressing the evolving threats is unclear. To fill in this gap, we first carried out an in-depth interview study with 15 experts in automotive cybersecurity, uncovering the particular challenges encountered during security activities and the limitations of current regulations. We identified 20 key insights from the interview data, ranging from the challenges and gaps in the existing automotive security industry to the limitations and recommendations for current regulations. Notably, we discovered that the quality of threat cases provided by existing regulations is unsatisfactory, and the Threat Analysis and Risk Assessment (TARA) process is often highly inefficient due to the lack of automatic tools. In response to the above limitations, we first built an improved threat database for automotive systems using the collected interview data, which enhanced the existing database both quantitatively and qualitatively. Additionally, we present CarVal, a datalog-based approach designed to infer multi-stage attack paths in IVNs and calculate risk values, thereby making TARA more efficient for automotive systems. By applying CarVal to five real vehicles, we performed extensive security analysis based on the generated attack paths and successfully exploited the corresponding attack chains in the newly gateway-segmented IVN, uncovering new automotive attack surfaces that previous research failed to cover, including the in-vehicle browser, official mobile app, backend server, and in-vehicle malware.**

1. Introduction

In recent years, modern vehicles have experienced significant advancements in technology, resulting in the increased complexity in terms of both external attack surfaces and internal in-vehicle network (IVN) topology. Specifically, as manufacturers are bringing more and more advanced functions to modern vehicles (e.g., remote control and Over-The-Air (OTA) update), the corresponding attack surfaces have been greatly expanded compared with previous vehicles with fewer interfaces [21, 43]. Additionally, the IVN is also getting increasingly complex. On the one hand, the number of Electronic Control Units (ECUs) are growing rapidly to meet the demand on advanced functions (e.g., Advanced Driver-Assistance System - ADAS); on the other hand, the IVN topology is also getting more sophisticated to assist more efficient in-vehicle information exchange (e.g., switching to the gateway-segmented design [36, 42], or even more advanced zonal design [1]). Due to the above facts, modern vehicles are significantly different from the ones in previous research [21, 43], and it remains a challenge how to ensure the cybersecurity of modern vehicles in such context.

In light of these emerging challenges, regulatory bodies have introduced a series of standards and regulations, such as WP29 R155e [7] and ISO 21434 [6], in an attempt to establish a baseline for automotive cybersecurity. These regulations aim to provide a framework for the industry to follow, ensuring the security and safety of automotive systems. However, it remains unclear whether these regulations are sufficient in offering a solid foundation for addressing the ever-evolving cybersecurity threats faced by modern vehicles.

To address this gap, we first conducted an in-depth semi-structured interview study with 15 automotive cybersecurity experts to reveal the practitioners' perspective with a focus on the cybersecurity regulations. From our interview study, we extracted 20 key insights, spanning from the challenges and gaps in the existing automotive security industry to the limitations and recommendations for current regulations. In summary, two major limitations were identified from the interview. Firstly, we found that the specific threat cases given by current regulations suffer from various limitations and cannot offer a sufficient guideline to follow. Secondly, we identified that current standard [6] only presented the high-level methodology for Threat Analysis and Risk Assessment (TARA), and the specific implementation of TARA in industry can be very inefficient due to the lack of automatic tools.

§ *The corresponding author.*

DOI 10.1109/SP54263.2024.00080

Guided by the above findings, we conducted the following works to fill the identified gaps:

An improved automotive threat database. In response to the unsatisfactory quality of the existing threat database, we constructed an improved automotive threat database using the collected interview data, which improved the existing database both quantitatively and qualitatively. Specifically, this improved database is constructed by practical threats collected from interviews, with the hierarchical structure including 7 themes (i.e., high-level groups), 28 codes (i.e., low-level groups), and 119 detailed threat descriptions. We also analyzed the relations among these threats and further proposed a Knowledge Graph (KG) based representation as an interconnected database.

An automatic tool for TARA. Additionally, to address the low efficiency of TARA, we introduce CarVal, the first Datalog-based approach to automatically reason attack paths in IVNs and calculate corresponding risk values. CarVal is capable of reasoning multi-stage attack paths in increasingly complex IVNs and generating logical attack paths to guide subsequent analysis (e.g., security testing). To demonstrate CarVal's utility, we applied it to five real vehicles and generated realistic attack paths. Guided by the generated attack paths, we performed extensive security analysis on five vehicles, and successfully exploited various attack paths in the gateway-segmented IVN. Particularly, a series of *new* automotive attack surfaces were exploited, including the 1). In-Vehicle Infotainment (IVI) browser, 2). official mobile app, 3). backend server, and 4). IVI malware.

In summary, this paper makes the following contributions:

- An in-depth interview study with 15 automotive security experts, identifying 20 key points ranging from challenges in conducting security activities to specific limitations of existing regulations.
- An improved threat database for automotive cybersecurity, developed using the data collected from the interviews, which enhances the existing database both qualitatively and quantitatively.
- The design and development of CarVal, a novel Datalog-based approach to infer attack paths and assess corresponding risk values in modern IVNs. CarVal is capable of inferring multi-stage attacks and prioritizing attack paths based on the calculated risk values.
- Extensive security analysis on five real cars based on attack paths discovered by CarVal, which led to the identification of new attack chains that previous works failed to cover, from new attack surfaces to the ECUs behind the gateway.

The remainder of this paper is structured as follows. §2 provides the necessary background information. We introduce the methodology of our interview study in §3, and present our key findings in §4. The improved database for automotive systems is shown in §5. We introduce our Datalog-based approach for attack path reasoning in §6, and present our experimental analysis in §7. Finally, we present the discussion in §8 and conclude the paper in §9.

2. Background

2.1. Threat Analysis and Risk Assessment for Automotive Systems

The increasing computerization and complexity of modern vehicles have led to the emergence of new attack surfaces and corresponding cyberattacks [21, 43, 59, 73]. This necessitates conducting TARA on contemporary vehicles to identify potential threats, vulnerabilities, and associated risks within the system. By comprehending these risks, vehicle manufacturers can implement appropriate mitigation strategies. However, security assessment guidelines provided by current regulations, such as WP29 R155e [7] and ISO 21434 [6], exhibit limitations in delivering comprehensive security assessments. These guidelines are often too generic and fail to provide specific guidance on addressing security risks related to a particular system [24]. Furthermore, existing regulations [7, 9] only enumerate discrete threats that manufacturers should consider, leaving an efficiency gap in automated risk assessment for modern vehicles.

2.2. Regulations on Automotive Cybersecurity

The growing number of automotive cyberattacks in recent years underscores the urgent need for standards and regulations that enforce automotive cybersecurity. The United Nations Economic Commission for Europe (UNECE) introduced WP29 R155e [7] as a compulsory regulation that Original Equipment Manufacturers (OEMs) and Tier suppliers in UNECE countries must adhere to. This regulation mandates OEMs to establish a CyberSecurity Management System (CSMS) for managing security risks throughout a vehicle's lifecycle. Although R155e enumerates potential automotive cyberattacks and corresponding defenses as references for CSMS, it does not offer specific guidance on configuring a CSMS to meet the requirements. The International Organization for Standardization (ISO) proposed ISO 21434 [6] as a non-mandatory standard that supplies general guidelines for managing security risks across the automotive lifecycle. Contrasting WP29 R155e, which is obligatory, ISO 21434 provides suggestions on how to construct a CSMS. GB/T 40861-2021 [9], published in China, is a standard that stipulates general requirements for ensuring automotive security. This standard outlines cybersecurity threats faced by modern vehicles across six dimensions, encompassing software and hardware systems, in-vehicle and long-distance communication, and in-vehicle data.

3. Interview Methodology

3.1. Study Setup

We present the methodology of our interview in this section, including the design of the interview protocol, the recruitment, the interview procedure, the data analysis

TABLE 1: Interviewee demographics

ID	Sex	Exp[1]	Company [2]	Position[3]	Duration
P1	M	10	C1: 1st Party	TARA	2:52:47
P2	M	3	C2: 3rd Party	TARA, Manag	1:18:16
P3	M	5	C3: 1st Party	TARA, Manag, Reg	1:05:21
P4	M	4	C4: 3rd Party	Test	0:59:55
P5	M	3	C4: 3rd Party	Test	0:43:31
P6	M	3	C5: 3rd Party	Test	1:05:36
P7	M	3	C4: 3rd Party	Test, TARA	0:55:44
P8	M	7	C6: 1st & 3rd	Test	0:38:50
P9	M	5	C7: 3rd Party	Test, TARA, Manag	0:56:32
P10	F	3	C8: 1st Party	Test, TARA, Manag	0:53:15
P11	M	3	C8: 1st Party	TARA, Mang	1:27:38
P12	M	5	C8: 1st Party	Test, TARA	1:33:18
P13	M	6	C6: 1st & 3rd	TARA, Manag	1:37:45
P14	M	20	C9: 3rd Party	TARA, Manag, Reg	2:11:10
P15	M	3	C3: 1st Party	TARA, Manag	1:05:34

[1] Years of working experience in security;
[2] From 1st party vehicle manufacturer or 3rd party supplier;
[3] TARA: Threat Analysis and Risk Assessment; Manag: Project manager; Reg: Regulation-related study; Test: Security testing.

process, and the detailed interview structure. Other details such as ethical considerations are presented in Appendix.A.

Design of the Interview Protocol. The preliminary interview protocol was developed in accordance with the three exploratory motivations: 1) identifying challenges and gaps in the implementation of security activities within the industry; 2) evaluating the effectiveness and relevance of current regulations in addressing specific threats; and 3) exploring the limitations and providing recommendations for enhancing existing regulations. In particular, a qualitative analysis of current regulations was conducted to: 1) establish metrics for assessing existing threats, and 2) create an initial threat database by integrating knowledge from multiple regulations. Specifically, we first collected the threat descriptions from current regulations [6, 7, 9], and two authors performed iterative coding on them to derive (a). a list of initial threats that are expected to be expanded during the interview process, and (b). the evaluation criteria on assessing these threats. This qualitative analysis contributes to the design of the interview protocol. This qualitative analysis extracted 38 threats distributed in 6 codes, and our interview study finally expanded this database to 119 threats in 28 codes. The protocol can be accessed in [2]. After 10 rounds of interviews, the protocol was finalized and remained consistent for all subsequent interviews.

Recruitment. We invited experts working in the field of automotive cybersecurity from both first-party automotive manufacturers and third-party suppliers to participate in the interview (two employer companies play the role as 1st-party OEM and 3rd-party provider at the same time). The information of the 15 interviewees is presented in Table 1. On average, they had about 6 years of experience in the security field, and there are senior experts with experience over 10 years (P1 and P14). 8 out of 15 are from 1st-party OEMs, and 9 are from 3rd-party suppliers, with two overlaps. Their roles included TARA, security testing, project management, and regulation study, ensuring that all participants were experienced experts from diverse companies who could provide convincing opinions in the field. Particularly, the 1st party manufacturers include companies from multiple countries (e.g., China and Germany), and the 3rd party suppliers also offer security services (e.g, security testing and security consulting) for automotive companies from all over the world (e.g., including car brands from Germany, Japan, America, China, and others). After the 15th round of interview, we identified a saturation of new opinions and the threat cases that the experts can offer, and stopped recruiting more participants.

3.2. Procedure and Data Analysis

Interview Procedure. The interviews were conducted through online meetings. During the interviews, the interviewer (i.e., author of this paper) shared the screen to display the interview protocol and related materials (e.g., content of the regulations under discussion) to the interviewees. Both audio and video of the interviewer's screen were recorded for further analysis. We began the interview by collecting basic information about the interviewees, and then proceeded to discuss the specific topics in the protocol. The interviews were conducted in a semi-structured manner, allowing the interviewees to freely express their thoughts. After the interview study, we derived an automotive threat database with 119 threats under 28 codes, and sent back this database to all participants for suggestions on final modifications. The interview process was started in November 2022 and finished in March 2023.

Data Analysis. We first transcribed the recorded audio to text for further analysis. We then carried out an iterative open-coding process on the collected data [15, 23]. First, an initial codebook was established by all authors based on the interview protocol. Then, two authors separately performed multiple rounds of iterative open coding on all interview data. After that, the two authors verified each other's coding results and resolved the conflicts, and meanwhile updated the codebook. We continued with the iterative coding process until no new code emerged [15]. The final codebook is available in [2].

3.3. Interview Structure

The interview, as depicted in Fig.1, starts with an introduction where objectives are outlined and interviewees are encouraged to share personal opinions on security activity implementations and regulations (**Sec.1 Introduction**). We then collect basic information about the interviewees' work experience and role within their organizations (**Sec.2 Collection of Basic Information**). In the next stage, the interview delves into the specifics of how security activities are carried out per existing regulations (**Sec.3 Investigation of Security Activity Implementation**). Following Sec.3, the effectiveness of TARA approach proposed by ISO 21434 is assessed by scrutinizing its implementation within the interviewee's group (**Sec.4 Assessment of Existing TARA Approaches**). The quality of threat databases provided by existing regulations is then evaluated, alongside the showcasing of an integrated threat database derived from a preliminary study (**Sec.5 Evaluation of Current Threat Databases**).

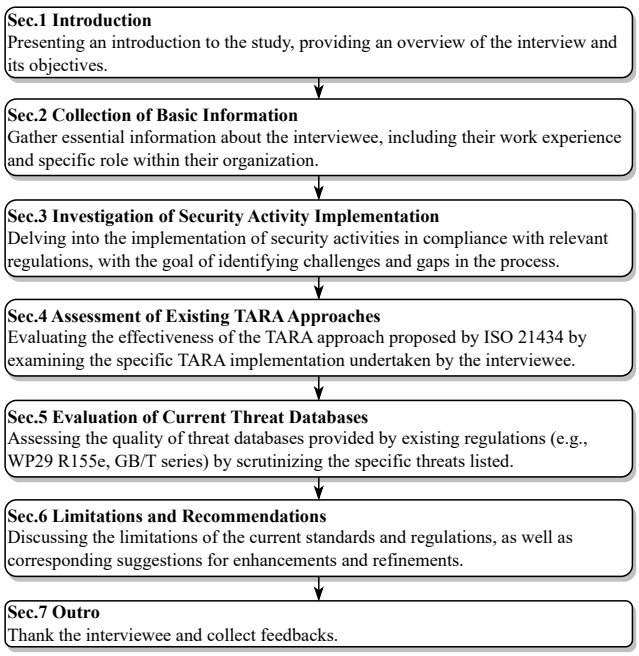


Figure 1: The flow of our semi-structured interview. Each section unfolds with particular question, in the meantime interviewees can freely express their thoughts that might discover insights beyond the current section.

The interview proceeds to discuss the limitations of current standards and regulations and collects suggestions for improvements (**Sec.6 Limitations and Recommendations**), before concluding with an expression of gratitude and feedback collection for improving the interview process (**Sec.7 Outro**).

4. Interview Results

In this section, we present our findings based on 15 semi-structured interviews conducted with experts. The structure of this section follows the interview flow presented in Fig. 1. Each subsection reports the detailed findings of the corresponding interview section, with specific Key Points (KPs) identified and summarized at the end of each part. Notably, we highlight direct quotes from the interviewees by italicizing them and using quotation marks. Particularly, we mainly report KPs discovered from Sec.4, 5 and 6 in Fig.1, and more insights are available in [2].

4.1. Assessing TARA

We identified the following key points revealing the weakness of current TARA, which corresponds to the Sec.4 in Fig.1.

• **KP.1: Asset identification is difficult.** We identified that the very first step of TARA: asset identification, is a challenge stage due to the often-missing information, and the complexity of the the automotive system. Specifically, as reported by P14: "*The asset identification often costs more than half of the time of the whole TARA process. This is because the materials we rely on are often insufficient to list all assets, and we need to consistently contact the provider for the necessary information and improve the comprehensiveness of the listed assets.*" (P14).

• **KP.2: Lack of objective definitions and criteria.** Another major limitation we identified from ALL interviewees (11) working on TARA is that the current TARA is a high-level methodology, and there is a lack of specific definitions and criteria to ensure the effectiveness and consistency of the TARA results (e.g., when TARA is performed by different groups). E.g., as reported by P12: "*The evaluation of certain criteria in current TARA can vary a lot between different persons or groups, and there is a lack of more specific criteria. For example, we will do a TARA on the specific product, and our suppliers will also do a TARA on it, but the result of our TARA can be very different as the analysis is based on the subjective expertise instead of objective metrics.*" (P12).

• **KP.3: Low level of automation and low efficiency.** We also identified from ALL interviewees working on TARA that the TARA process is often in a low level of automation, and a huge manual effort is still required to finish TARA. E.g., as reported by P11: "*A lot of effort is needed to analyze the attack path in our TARA process. Currently, this process still heavily relies on our own experience and expertise. *" (P11). Additionally, as reported by P12: "*It is still difficult to craft an automated TARA process because this process is complex and require certain expertise. At least, we currently still rely on our expertise to do the very specific TARA on the products. *" (P12).

Summary on TARA: We identified that currently the TARA applied by practitioners suffers from limitations including requiring heavy manual effort, low efficiency, and the lack of objective definitions and criteria. Although ISO 21434 has presented the high-level TARA methodology, it still remains a challenge on how to conduct TARA efficiently.

4.2. Evaluating Threat Database

This section reveals the key points related to the specific threats listed by existing regulations, which corresponds to the Sec.5 in Fig.1.

• **KP.4. More common and automotive-specific threats are needed, rather than copying existing threats from other areas.** The automotive system consists of multiple sub-components (e.g., the cloud, the app side, the IoT-related modules). However, the majority of the interviewees (14/15) agree that currently listed threats are largely copied from other domains but not the practical or commonly-seen threats in automotive systems. E.g., as reported by P2: "*Many existing threats are just copied from other areas, rather than describing the truly common threats for automotive systems. I do not think it necessary to detail these already known threats. *" (P2). As also reported by P10: "*We are expecting the regulations to give more common and detailed threats that are really related to current*

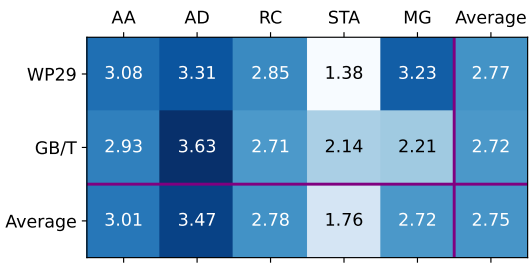


Figure 2: Average score from 5 evaluation criteria for WP29 R155e [7] and GB/T [9].

automotives. For example, some manufactures are adding some fancy functions to their products, such as remotely heating the seat. It is OK for the regulations to not mention these unique functions. However, I think it necessary for the regulations to give very detailed guidelines on the very common functions, such as remotely opening the door, which I believe is a function that the majority of vehicles have already applied. " (P10).

• **KP.5: Low scores are given to existing threats by practitioners.** During the interview, we asked the experts to evaluate the quality of threats in current regulations from five metrics, including the Attack Description (AD), the Root Cause (RC) of the threat, the Security Testing Approach (STA) to identify the threat, and the MitiGation (MG). Specifically, they were asked to choose a score from 1 to 5 to present how satisfy they were about the existing threats from the above 5 aspects, and the overall scores are shown in Fig.2. Note that the average scores for WP29 R155e and GB/T are 2.77 and 2.72, respectively, representing that experts are overall unsatisfied with the quality of current threats. Moreover, extremely low scores are identified from the aspect of STA.

Summary on threats in regulations: From the practitioners' perspectives, the specific threats listed by existing regulations are far from being satisfying. Particularly, there is a lack of specific threats for automotive systems, and currently listed threats are short of a comprehensive description from various dimensions (Fig.2).

4.3. Limitations and Recommendations for Existing Regulations

This section reveals the key points related to the open-ended discussion of the limitations on current regulations, which corresponds to the Sec.6 in Fig.1.

• **KP.6: More detailed information would certainly help the security groups.** Multiple previous key points have revealed that the missing of particular details brings challenges to security groups. Particularly, we identified that ALL interviewees agree that a more detailed regulation would certainly help their work, including being more specific on provided threat, giving clear threshold and objective criteria, etc. E.g., as reported by P10: "*Our group mainly relies on our TARA results to express the specific threats to other groups. However, this process would be much more efficient if more details could be found in current regulations.*" (P10).

• **KP.7: Gaps exist between traditional IT threats and automotive threats.** We identified that current regulations failed to give guideline on how to define the severity of the specific threats, especially when the threat exists in the automotive system instead of traditional IT networks, which could result in an incomplete understanding of the threat. This KP also corresponds to the previous KP.4. Particularly, as reported by P10: "*I think there is a significant gap when we switch our concepts from the traditional IT threats to the automotive threats, because we are unsure about how to define and analyze the threats when they are connected to the automotive system with the very specific hardware. For example, an engineer with only software security background would find it challenging to precisely define the threat in automotive system. For example, our group would think a vulnerability allowing attacker to remotely open the door is very critical, but other groups would tell me that this would not be a critical case according to the functional safety regulation. I would expect the regulations to give more details on how we should understand the severity of the specific threats.* " (P10).

Summary on other limitations: Besides the inefficiency on TARA and threat database, we also offer various insights on how current regulations could be improved. More detailed results are available in [2].

4.4. Summary on Interview Study

Compared with the previous studies revealing the "business decision" impact of security parties [34, 54, 65], we emphasize the following key points specific to automotive security, which are not covered by the previous works. First, we have identified a series of limitations of the specific regulations on automotive cybersecurity, including the insufficiency of threat database, TARA guidance, security testing approach, and many others. Second, we also identified the challenges in conducting security activities in automotive system, including the lack of automation in TARA process, the lack of quantifiable criteria for risks assessment, conflicts with other groups, and many others. Specifically, these challenges can be attributed to the following two aspects:

Lack of high-quality threat database. The threats offered by existing regulations are insufficient from various dimensions (KP.2, KP.4, and KP.5). Due to the above gap, practitioners have to rely on the experience and expertise of the group to perform security activities, which could be incomplete or inefficient.

Lack of efficient tool for TARA. ISO 21434 [6] presents the high-level methodology of TARA, but the specific implementation is still facing various challenges. Particularly, the lack of criteria can lead to the inconsistent TARA results (KP.2), and the lack of automatic tools can make TARA very inefficient (KP.3).

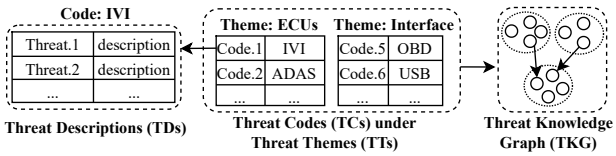


Figure 3: The proposed hierarchical framework to describe automotive cybersecurity threats.

5. Improved Threat Database

5.1. Hierarchical Framework for Automotive Threats

In response to the lack of high-quality automotive threat database, we construct a new threat database that is improved by the collected threats from the interview. Particularly, we use a hierarchical framework to present the automotive-specific threats (in Fig.3), in which the involved concepts are explained as follows:

Threat Description (TD). A threat description (TD) is the smallest element in the framework. It is a set of natural language sentences to describe the details of one particular threat, including the specific Attack Description (AD), the Root Cause (RC) of the threat, the Security Testing Approach (STA) to identify the threat, and the MitiGation (MG) to prevent the threat.

Threat Code (TC). A threat code (TC) is a group of TDs under a particular category. Here the word *"code"* comes from the qualitative analysis methodologies [23], in which the process of *coding* is to give labels to the qualitative data (e.g., interview texts). For example, in Fig.3, *Code.1 IVI* is the code containing the threat descriptions under the in-vehicle infotainment (IVI) ECU.

Threat Theme (TT). A threat theme (TT) is a group of threat codes following a particular high-level classification logic. For example, in Fig.3, the *Threat Theme: ECUs* includes the threat codes representing the in-vehicle *ECUs* (e.g., IVI, ADAS), while *Threat Theme: Interface* includes threats related to vehicular interfaces (e.g., OBD, USB).

Threat Knowledge Graph (TKG). We derive the concept of knowledge graph (KG) [17, 38, 70] to further represent the relations between the threat codes. Specifically, a knowledge graph can be represented by a set of triplets: *(head entity, relation, tail entity)*, meaning that the *head entity* and the *tail entity* has the particular *relation*. In our scenario, the entities are the threat codes, and the triplet *(TC.1, relation, TC.2)* represents the logical relation between the two codes. For example, the triplet *(Code.1 IVI, vulnerable to threats in , Code.6 USB)* connects the code IVI and code USB because the USB interface is a common interface on IVI.

5.2. Detailed Threats

The final result of our threat database is shown in Fig.4, with the following specific threat theme and codes:

T1: General Requirements. The various ECUs can share a set of theats that are general to various implementations, and this T1 describes these common threats from five threat codes: *C1.Hardware, C2.Software, C3.RTOS, C4.Complex OS, and C5.Data*. The advantage of setting up this theme is that *we do not need to repeat these common threats in the specific ECU categories*. For example, secure boot is the de facto mitigation that should be deployed on various types of ECUs. There are 24 threat descriptions under T1.

T2: In-Vehicle Components. T2 describes the threats to specific components in the vehicle, including the threats on various ECUs and on the In-Vechile Network (IVN). T2 contains the following 8 codes: *C6.IVI, C7.Telematics, C8.Sensor, C9.Gateway and Zone Controller, C10.ADAS, C11.IVN, C12.BMS, and C13.Other ECUs*. These codes focus on the threats that are particular to the function of the ECU. For example, the *C6-10: browser threat*, is the very specific threat that exists in the IVI but not on other ECUs, because the browser module has been widely used in the IVI system to support rich infotainment functions. There are 36 threat descriptions under T2.

T3: Outside-vehicle Components. T3 describes the threats for specific components outside the vehicle, but can communicate with the vehicle and affect automotive cybersecurity. Specifically, T3 contains the following 3 codes: *C14.Mobile APP, C15.Backend Server, C16.Charging Pile*. The vulnerabilities in these external components can pose a threat to the vehicle itself. For example, the private data can be leaked through the charging pile. There are 14 threat descriptions under T3.

T4: Communication Protocols. T4 describes the threats to the communication protocols implemented in the automotive context. Specifically, T4 contains the following 4 codes: *C17.UWB, NFC and BLE, C18.V2X, C19.CAN, and C20.Ethernet*. The unsafe implementation of these protocols can introduce risks. For example, lack of encryption on the data transmitted via the protocol can lead to information leak. There are 16 threat descriptions under T4.

T5: Communication Channels/Interfaces. T5 describes the threats on the communication channels and interfaces on the vehicle. Specifically, T5 contains the following 4 codes: *C21.Wi-Fi, Bluetooth and Cellular, C22.Charging Port, C23.USB and SD card, C24.OBD*. Unsafe implementation of these interfaces leads to threats when these interfaces are exposed to the attacker. For example, the attacker can modify vehicular parameters through the OBD port due to the lack of proper authentication. There are 15 threat descriptions under T5.

T6: Vehicular Functions/Services. T6 describes threats to vehicular function and services, with the following 3 codes: *C25.OTA, C26.Diagnostic, C27.Remote monitor and control*. The implementation of these "trendy" functions can vary for different manufacturers and car models, and can introduce risks when the design is insecure. For example, the unsafe implementation of the secret keys for remote control can be exploited to launch attacks. There are 12 threat descriptions under T6.

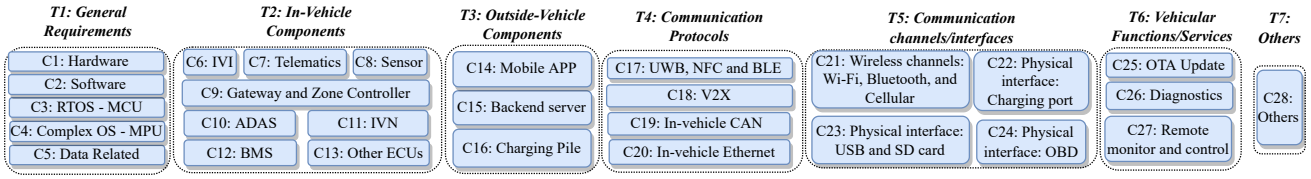


Figure 4: An improved hierarchical threat database derived from the interview study, containing 28 threat codes (TCs) under 7 threat themes (TTs). This database serves as an improvement to existing regulations both qualitatively and quantitatively, and is available in [2].

T7: Others. T7 includes other threats (e.g., insider attack) that do not fit into other themes. There are 3 threat descriptions under T7.

Several authors have gathered together and performed various rounds of revision on this threat database. The complete database is available in [2]. Moreover, we analyzed the relations among the threat codes and presented a TKG which presents the connections of the threats (in Appendix.B).

6. CarVal: Approach

In response to the lack of efficient tool for TARA, we introduce CarVal: the first Datalog-based approach designed to automatically generate attack paths in IVNs, calculate corresponding risk values, and thus make TARA in automotive systems more efficient. We first present the challenges encountered during the design of CarVal in §6.1. Then, in §6.2, we describe the workflow of CarVal in detail, including how attack paths are inferred and how risk values are calculated. An example is given in §6.3 to provide a clear demonstration of the approach. Finally, we provide the implementation details of CarVal in §6.4.

6.1. Challenges and Solutions

Challenges. Previous research has proposed datalog-based approaches for automatic attack path generation in enterprise networks (e.g., MulVAL [51, 52, 53]). However, these approaches cannot be directly applied to the automotive domain due to the particular challenges. Firstly, traditional attack path reasoning engine [51, 52, 53] relies on manually crafted reasoning rules in IT networks. However, there are no existing rules that could be applied to reasoning attack paths in IVN. Secondly, unlike enterprise networks, where each node is treated as a host with an identical set of rules, the IVN consists of electronic control units (ECUs) with various hardware and software settings. Unfortunately, previous approaches [51, 52, 53] do not account for these new features in IVN and cannot represent the up-to-date IVN model, and thus it is unclear how to transform the IVN network into Datalog representation. Thirdly, previous works only discussed how to calculate the feasibility (i.e., likelihood) of specific attacks in the attack path [32, 68, 69], and failed to consider the attack impact indicated by ISO 21434 TARA [6], which leads to incomplete output in the specific automotive system.

Solutions. Firstly, we construct the reasoning ruleset and define cybersecurity attacks based on the threat database collected from industry experts through an extensive interview study. Secondly, we introduce a hybrid model combining the *bus model* that represents the broadcasting nature of in-vehicle bus (e.g., CAN bus) with the *star model* that represents the up-to-date gateway design (further shown in Fig.7). This model can precisely present the IVN model and contribute to correct reasoning of attack paths. Thirdly, we enhance the reasoning engine by calculating the *attack impact* of each node on the path, in addition to the *attack feasibility* indicated by ISO 21434. This is a improvement specifically for the automotive systems.

6.2. Workflow

We propose CarVal, an automatic approach for attack path reasoning and risk assessment in IVN, which is shown in Fig.5.

6.2.1. Input. There are the following four parts of input to the CarVal reasoning engine:

Attack Goal. This component specifies the particular attack that serves as the objective for datalog reasoning. For instance, the following clause represents the attack goal for performing a cross-domain attack on the Body Control Module (BCM) ECU, which involves sending malicious commands like unlocking the doors or opening the car windows:

```
crossDomainAttack(bcm).
```

The attack goal describes a specific attack in the IVN and is derived from a set of primitive nodes, which include the IVN information and possible vulnerabilities. These nodes are referred to as derived Attack Nodes (AN).

Attack Entry. This component describes how the attacker can gain access to the automotive system, which serves as the starting point of the attack path. For instance, the following clause assumes that the attacker can access the In-Vehicle Infotainment (IVI) ECU via the Wi-Fi channel:

```
attackerCanAccess(ivi, wifi).
```

In the attack path, such an attack entry is referred to as the Entry Node (EN).

Vulnerability Set. This component consists of the possible vulnerabilities that can lead to specific attacks. For instance, the following clauses describe the vulnerabilities that exist in two ECUs: IVI and BCM, and these vulnerabilities will serve as the prerequisites to the Attack Node (AN):

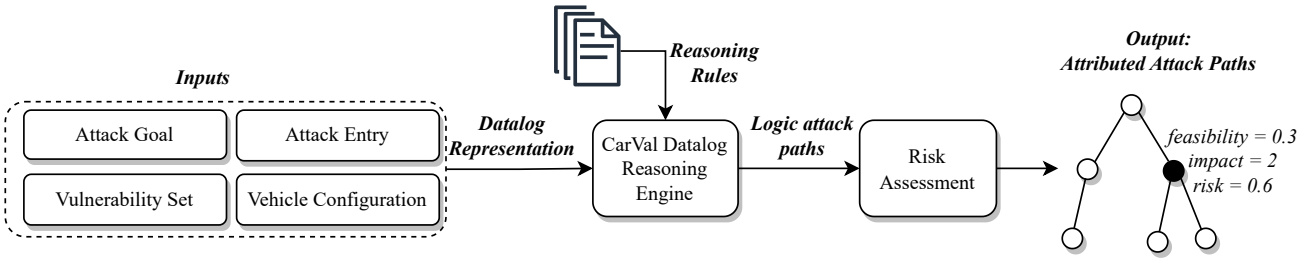


Figure 5: CarVal workflow: Automatic attack path reasoning and risk assessment in automotive system.

```
vulExists(ivi, 'lowPrivCodeExec').
vulExists(ivi, 'unauthorizedBroadcast').
vulExists(bcm, 'lackMessageAuth').
```

Such vulnerabilities are referred to as Vulnerability Nodes (VNs) on the attack path.

Vehicle Configuration. This component includes vehicle-specific information required for attack path reasoning. Specifically, such information comprises the IVN topology and the attributes of the Electronic Control Units (ECUs) and buses in the IVN. For instance, the following clauses indicate that the IVI ECU and Gateway (GTW) ECU are both located on the *infoCAN* bus, and the broadcasting nature of this Controller Area Network (CAN) bus may lead to specific attacks:

```
ecuOnBus(ivi, infoCAN).
ecuOnBus(gtw, infoCAN).
busTypeBroadcast(infoCAN).
```

Such information is referred to as Fact Nodes (FN) on the attack path.

6.2.2. Attack Path Reasoning. CarVal initiates datalog reasoning upon receiving the aforementioned inputs to determine the feasible attack path from the attack entry to the attack goal. The effectiveness of this reasoning process depends on the carefully-designed reasoning rules. For instance, the following rule explains how an attacker can enhance the attack impact after executing malicious code in the ECU:

```
attackerBroadcastOnBus(ECU, Bus) :- // AN
    execCode(ECU, Priv), // AN
    ecuOnBus(ECU, Bus), // FN
    busTypeBroadcast(Bus), // FN
    vulExists(ECU, 'unauthorizedBroadcast'). // VN
```

In this reasoning rule, the attacker can broadcast the attack message on a particular bus (e.g., CAN bus) bus after achieving code execution in the particular ECU (e.g., IVI). It is worth noting that this rule is derived from one Vulnerability Node (VN), two Fact Nodes (FN), and one Attack Node (AN).

6.2.3. Risk Assessment of Generated Attack Paths. The datalog reasoning module generates a logical attack path, representing how an attacker can achieve the attack goal from the attack entry. Subsequently, CarVal conducts automatic risk assessment of the derived attack nodes along the attack path. As per the ISO 21434 regulation, the risk value of a threat in an automotive system is not solely determined by its feasibility, but also by its impact. Initially, starting from the attack entry, CarVal calculates the attack feasibility and attack impact of all ANs (representing specific threats) along the path. Finally, it evaluates the risk values by taking into account both the feasibility and impact.

TABLE 2: Explanations for the symbols used for risk assessment.

Symbol	Range	Explanation
f_{EN}	(0, 1]	How likely the attacker can access this particular attack surface of EN
f_{VN}	(0, 1]	How likely such a vulnerability in VN can exist in automotive system
i_{VN}	>1	How severe the potential impact brought by VN can be
f_{AN}	(0, 1]	How likely the AN can happen when all prerequisite nodes are satisfied
i_{AN}	>1	The intrinsic severity of the AN
F_{AN}	(0, 1]	Cumulative feasibility of this AN on the specific attack path
I_{AN}	>1	Cumulative impact of this AN on the specific attack path
R_{base}	>0	Baseline risk value of the inferred AN
R_{AN}	>1	Cumulative risk of this AN on the specific attack path
$N_1 \rightarrow N_2$	N/A	On the attack path, N_1 is one of the prerequisite to infer N_2

Definitions. The risk assessment module uses two metrics: *feasibility* and *impact*. The specific definitions of these metrics are presented in Table 2. The intrinsic *feasibility* and *impact* of a particular node are represented by lower case *f* and *i*, respectively. These values are fixed for all generated attack path. The cumulative metrics F_{AN}, I_{AN}, and R_{AN} represent the *cumulative feasibility*, *cumulative impact*, and *risk value*, respectively, which can vary for different attack nodes in different attack paths.

Attack Feasibility Calculation. The cumulative on-path attack feasibility of an attack node is calculated as the multiplication of the feasibility value in all its premise nodes. Since all feasibility values are within the range (0,1], the cumulative attack feasibility decreases as the attack path becomes deeper. The calculation is expressed in the following equation:

$$F_{AN} = f_{AN} \times \prod_{AN_i \rightarrow AN} F_{AN_i} \times \prod_{EN_i \rightarrow AN} f_{EN_i} \times \prod_{VN_i \rightarrow AN} f_{VN_i} \quad (1)$$

Attack Impact Propagation. Similarly, the cumulative on-path attack impact of an AN is the multiplication of the impact value in all its premise nodes. As all impact values are greater than 1, the cumulative attack impact will increase as the attack path gets deeper. This calculation can be expressed using the following equation:

$$I_{AN} = i_{AN} \times \prod_{AN_i \to AN} I_{AN_i} \times \prod_{VN_i \to AN} i_{VN_i} \quad (2)$$

Risk Value Calculation. The determination of the risk value associated with a particular threat in automotive systems is based on two factors, namely, its *feasibility* and *impact*. A higher feasibility and impact imply a greater risk value. According to ISO 21434 regulation [6], the final risk value of the node is a baseline value added by the multiplication of the feasibility and impact:

$$R_{AN} = R_{base} + F_{AN} \times I_{AN} \quad (3)$$

R_{base} can be assigned with any fixed value (ISO 21434 [6] assigned this value to be 1 for demonstration). After the above calculation, each derived attack node is assigned with the cumulative attack feasibility, impact, and risk value on the specific attack path.

6.3. A Demonstrating Example

The output of CarVal is the Attributed Attack Path (AAP), which provides both the logical flow of how an attack can be carried out in the IVN and the quantitative metrics, such as attack feasibility, impact, and risk value, associated with each attack node on the path. Figure 6 illustrates an example attack path generated by CarVal, in which the attack goal is Node.1: *attackerBroadcastOnBus(ivi, infoCAN)*, indicating that the attacker can broadcast malicious messages on the *infoCAN* bus in the IVN, by exploiting a compromised *IVI* ECU. Node.2 represents the reasoning rule connecting four prerequisite nodes: Nodes 3, 4, 5, and 6, with Node.6 being the prerequisite attack node (AN), indicating that the attacker needs to first achieve code execution in IVI. Node.3 is a vulnerability node (VN) indicating that a vulnerability exists in IVI that allows the attacker to send crafted messages on the internal bus. Nodes 4 and 5 are fact nodes (FN) representing supplementary conditions, including that the target bus (*infoCAN*) has a broadcast nature (i.e., CAN bus) and IVI is connected to this particular bus. Moreover, Node.6 *execCode(ivi, noRoot)* is inferred from two additional nodes: Node.8 indicating a vulnerability in IVI that allows the attacker to execute malicious code at a low privilege level, and Node.9 indicating the attack surface – the attack starts from the Wi-Fi channel on IVI.

The risk assessment module automatically computes the cumulative feasibility, impact, and risk value of all derived attack nodes on the path, as calculated by Equations 1 to 3.The attack node is represented by two separate nodes: the *RULE* node, indicating the intrinsic feasibility and impact (Nodes 2 and 7 in Figure 6), and the derived attack node, indicating the cumulative metrics (Nodes 1 and 6). The cumulative metrics of Node.6 are derived from Nodes 7, 8, and 9, while the final cumulative metrics of the attack goal Node.1 are derived from Nodes 2, 3, and 6. Note that

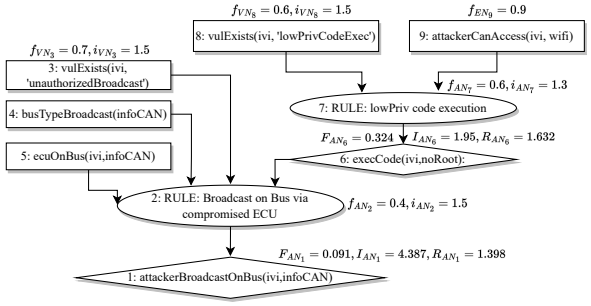


Figure 6: Example: an attributed attack path generated by CarVal.

Fig.6 is only for helping to understand how CarVal works, rather than being comprehensive. We use more examples to demonstrate how the risk assessment works for different attack paths in Appendix.C, and will detail more sophisticated attack paths that we exploited on real cars in §7.

6.4. Implementation

The datalog reasoning engine of CarVal is implemented based on the MulVAL reasoning framework [53] and XSB database system [5]. Particularly, the calculation of the attack impact, attack feasibility, and overall risk value is implemented by Python with treelib library [4]. The code of CarVal is open-sourced in [2].

7. Experimental Analysis on Real Vehicles

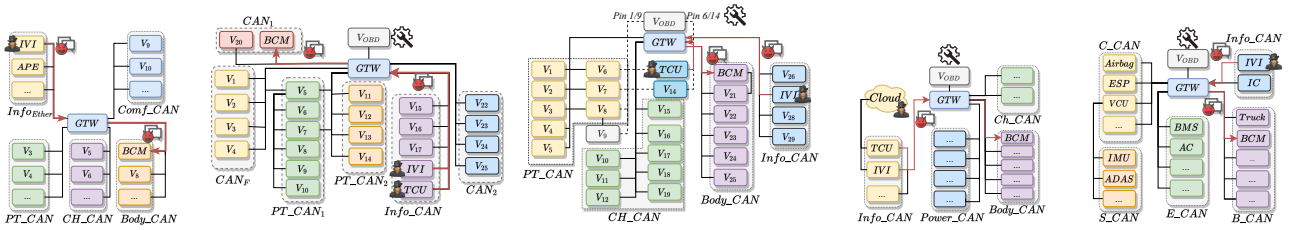
In this section, we demonstrate how CarVal can be applied to real cars to assist the security analysis to exploit realistic threats in automotive systems.

7.1. Vehicles Under Examination

We apply CarVal to five modern vehicles: a Tesla Model S, a BMW i3, a Roewe Marvel X, a Mercedez Benz E-Class, and another anonymous vehicle, which we will refer to as *Car A*, *Car B*, *Car C*, *Car D*, *Car E*. Particularly, *Car E* is anonymized because the corresponding vulnerabilities are still under the process of being fixed. Meanwhile, for *Car A*, *Car B*, *Car C*, and *Car D*, all vulnerabilities have been reported and fixed by the responsible party. All five vehicles offer sophisticated IVI systems, supporting entertainment activities through audio/video players and Web browsers, and their manufacturers provide mobile applications for remote control, which are the new attack surfaces that previous works failed to consider [21, 43, 49]. In addition, all five vehicles' In-vehicle network have adopted the up-to-date domain E/E architecture instead of the old two-bus in-vehicle network. Overall, the above new features distinguish our works from previous ones that exploited the *"old"* vehicles [21, 43, 49].

7.2. IVN Topology Discovery

We obtain the IVN topology of target vehicles from professional diagnostic tools ([8, 45, 57]), which are the de-



(a) *Car A*: Tesla Model S (b) *Car B*: BMW i3 (c) *Car C*: Roewe Marvel X (d) *Car D*: Mercedez Benz E-class (e) *Car E*: Anonymous vehicle

Figure 7: The IVN topologies and POC attack paths of five investigated vehicles.

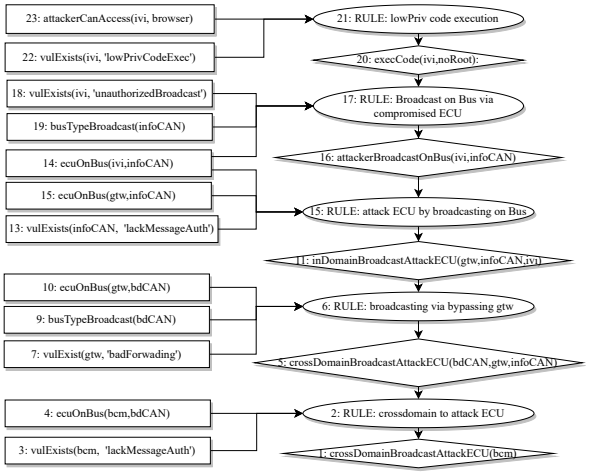


Figure 8: Attack Path 1: the attacker obtain code execution in IVI via IVI browser, and finally controls the BCM ECU via sending crafted bypass messages to gateway. This attack path is exploited on *Car A*, *Car B*, and *Car C*.

vices programmed to communicate with the vehicle through diagnostic protocols to diagnose the possible problems of the ECUs. Particularly, these diagnostic tools are embedded with the *IVN topology* of the target vehicles, in which how the ECUs are connected is presented to help the experts quickly gain an overview of the vehicular architecture. Fig.7 shows the IVN topologies of the five target vehicles we derived from the diagnostic tool.

The derived topologies are then parsed to corresponding datalog clauses, which will serve as the *Vehicle Configuration*, as stated in §6.2. Particularly, we set various *Attack Goals* and *Attack Entry* as the input to CarVal, and the specific topologies serve as the *Vehicle Configuration*, to output specific attack paths. Based on the attack paths generated by CarVal, we perform security analysis accordingly and finally exploited the practical attack chains and launched PoC attacks on real cars. In the following sub-sections, we will detail these attack paths and our corresponding security analysis.

7.3. Path 1: Bypassing gateway: from IVI browser to BCM

The IVI ECU, which is responsible for entertainment and communication functionalities within the vehicle, often contains a wide range of attack surfaces, and is often equipped with functionalities to control the BCM ECU. However, as shown by the specific topologies, the IVI and BCM are segmented by the gateway ECU, making previous injection attacks [21, 43] infeasible in this new IVN topology. To demonstrate the capability of CarVal to generate *multi-stage* attack paths in the increasing complex IVN topology, we set the attack entry on the IVI ECU, and the attack goal on the BCM ECU, to generate the corresponding attack path that shed light on the subsequent analysis including TARA and security testing. The logical attack path is shown in Fig.8, in which five attack nodes (Node.20, 16, 11, 5 and 1) are involved to reach the final attack goal. First, the attacker exploits IVI vulnerabilities via the IVI browser to obtain code execution (Node.20). By compromising the interface between IVI and in-vehicle bus *infoCAN*, the attacker broadcasts messages on infoCAN (Node.16), affecting the gateway ECU (Node.11). Then, the attacker crafts malicious messages to bypass the gateway and reach the *bdCAN* (Node.5), and finally transmits the attack message to the BCM ECU (Node.1).

PoC attack. The attack path in Fig.8 is validated on three vehicles: *Car A*, *Car B* and *Car C*, and we have conducted PoC attacks on these real vehicles. This attack path is demonstrated as the red bold line in Fig.7.(a), (b), (c) (from IVI to GTW and then to BCM). We first send malicious web pages to IVI browsers, obtain code executions on IVI, and compromise the interface from IVI to in-vehicle networks. By crafting bypass messages, we make the gateway transmit our malicious messages to the BCM ECU, allowing remote vehicle control.

Insights. By exploiting the attack path in Fig.8, we have demonstrated the significance potential threats in IVI software, with the representation of a new attack surface - browser, which is a general user interface that allows remote access and can be especially vulnerable if developed without caution [27, 29]. Additionally, we demonstrated the capability of CarVal to infer the multi-stage attack in complex IVN architecture.

7.4. Path 2: From Official APP to Car Control

The second attack path we demonstrate involves a replay attack, initiated from the official mobile app, to gain control over the vehicle's BCM. We selected the official mobile apps as the investigated attack surface due to their remote access functionality and the severe consequences of a potential

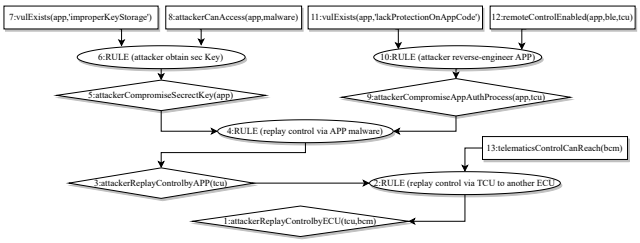


Figure 9: Attack Path 2: Control BCM by compromising the wireless communication between mobile app and telematic (i.e., TCU). This attack path is exploited on *Car C*.

compromise. Compared to dongle apps [72], which require a third-party device attached to the OBD port, the official app functions without external devices and is typically installed on the vehicle owner's phone. Consequently, if this official app is compromised, many on-road vehicles could be affected, causing significant financial damage to the manufacturer. The associated logical attack path is illustrated in Fig.9, involving four attack nodes (Nodes 5, 9, 3, and 1) to achieve the final attack goal. First, due to insufficient application code protection (e.g., lack of code obfuscation or encryption), an attacker can conduct extensive security analyses on the application code (e.g., reverse-engineering) to recover the authentication process between the app and the Telematics Control Unit (TCU) (Node.9). Subsequently, with malware installed on the victim's mobile phone, the attacker can access the secret key used for the authentication process (Node.5). Upon obtaining both the secret key (Node.5) and the authentication algorithm (Node.9), the attacker can impersonate the official app using malware and launch a replay attack on the TCU (Node.3). Finally, since the TCU can invoke control functions on the BCM, the attacker can initiate these controls by launching replay attacks from the malware (Node.1).

PoC Attack. The attack path in Fig.9 was validated on *Car C*. This path is represented as the red bold line of *TCU to Gateway (GTW) to BCM* in Fig.7.(b). Specifically, by conducting extensive security analyses on the corresponding mobile app, we recovered the authentication process, as shown in Fig.10. Our analysis revealed that: 1) the secret key used for authentication is set to update *every three months*, which is too long and allows the attacker the opportunity to crack this key; 2) the code contained in the APK file is not protected by obfuscation or encryption, enabling the attacker to directly access essential data through static analysis (e.g., the UUID used for BLE communication); 3) the code to generate the challenge response for the control request is called by the Java Native Interface (JNI) [10], and the critical authentication algorithm is stored in a *.so* file without obfuscation or encryption. Consequently, we launched a replay attack based on these vulnerabilities to remotely control the vehicle's door using an unrooted malware that sends crafted BLE messages, ultimately performing car control actions such as unlocking the door and opening the trunk.

Insights. While previous research focused on the security of OBD-dongle apps [72], we present the first practical attack that exploits vulnerabilities in the *official mobile app* to control a vehicle. We demonstrated that the official mobile app is a critical attack surface for modern vehicles and should be stringently protected.

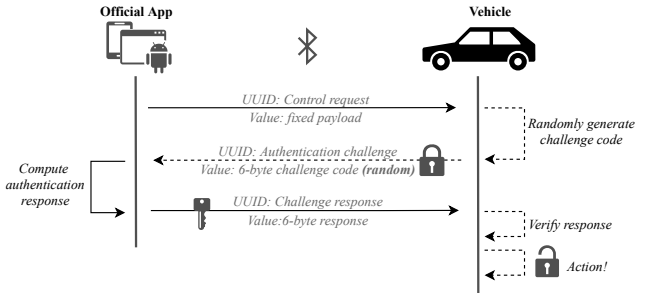


Figure 10: *Car C*: Remote control process of the mobile app via BlueTooth Low Energy (BLE).

7.5. Path 3: Multi-stage root via in-vehicle Ethernet

The third attack path we demonstrate illustrates a practical case of how an attacker can expand the scope of attack impact within an IVN. The corresponding logical attack path, based on the topology of *Car A*, is shown in Fig.11. In this attack path, the attacker first obtains root code execution in the In-Vehicle Infotainment (IVI) system by exploiting specific vulnerabilities (Nodes 10 to 7). Next, by leveraging a vulnerability between the IVI and Autopilot ECU (APE), the attacker expands code execution privileges from the IVI to the APE (Node.4). Finally, by exploiting a vulnerability in the APE, the attacker gains root execution in the APE (Node.1).

PoC attack. The attack path in Fig.9 is validated on *Car A*. Specifically, by conducting extensive security analysis on the IVI, we successfully obtain code execution via the browser's attack surface (Node.10) and further gain *root* code execution by exploiting a vulnerability in the outdated OS implementation (Node.7). Afterward, we identify that the IVI and APE communicate using an unencrypted UDP [12] protocol. By exploiting a vulnerability in the APE's update process, we successfully execute our code in the APE (Node.4). Finally, by compromising the authentication within the APE, we obtain *root* code execution in the APE (Node.1).

Insights. The exploitation of Path 3 demonstrates how an attacker can expand the impact range in the IVN. Specifically, we show that an attacker can exploit in-vehicle vulnerabilities to gain root access to another ECU (APE) that has no direct communication channel with external clients. As IVN topologies become increasingly complex and information exchange between ECUs intensifies, it is crucial to address such threats to ensure IVN security.

7.6. Path 4: From Cloud to Car Control

We also exploit a practical attack chain from a new attack surface, the *backend server*, to control the car. The corresponding logical attack path is shown in Fig.12. Specifically, the Telematics Control Unit (TCU) is responsible

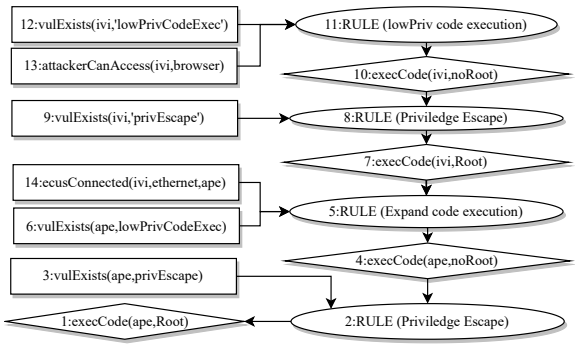


Figure 11: Attack Path 3: Multi-stage rooting via in-vehicle Ethernet. This attack path is exploited on *Car A*.

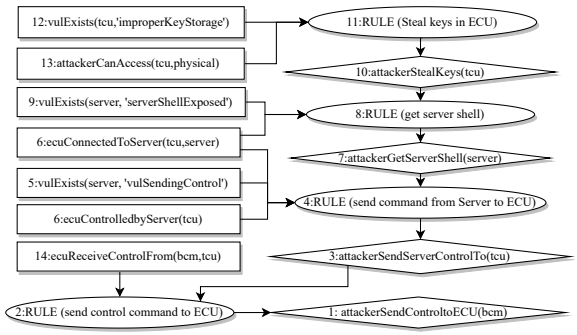


Figure 12: Attack Path 4: Compromising the backend server and sending control command back to vehicle. This attack path is exploited on *Car D*.

for remote communication, including transmitting messages with the mobile app (as shown in Path.3) and the backend server. Due to improper secret key storage in the TCU, an attacker can steal these keys by analyzing the TCU (i.e., dumping the firmware and performing reverse engineering), as shown in Node.10 of Fig.12. After obtaining the keys, the attacker accesses the server and achieves code execution (e.g., obtaining a shell) on the server by exploiting corresponding vulnerabilities (Node.7). With code execution, the attacker further analyzes how the server sends control commands to the TCU and replays these control commands (Node.3). Finally, upon receiving the malicious control messages sent by the attacker from the server side, the TCU triggers the control of the BCM (Node.1).

PoC attack. This attack path is validated and exploited on *Car D*. We first perform reverse engineering on the firmware dumped from the TCU and identify that the certificate used to establish authenticated connections with the intranet server is hard-coded in the firmware, which can be directly accessed (Node.10). Once the intranet server is accessible, we obtain the server shell by exploiting a Server-Side Request Forgery (SSRF) vulnerability (Node.7). With shell access on the server, we further analyze how the server sends commands to the vehicular TCU and can successfully send control commands to *any* vehicle by its Vehicle Identification Number (VIN) (Node.3), ultimately invoking control on the BCM (Node.1). In Fig.7.(d), this attack path originates from the Cloud, proceeds to the TCU, and bypasses the Gateway (GTW) to reach the BCM.

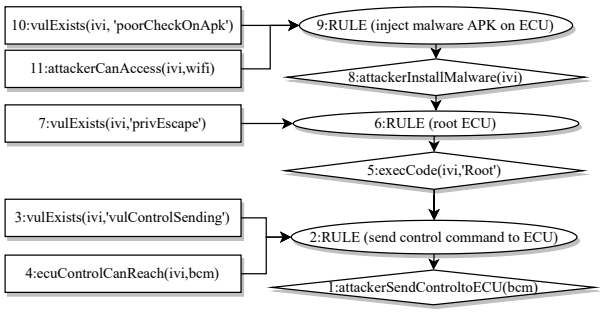


Figure 13: Attack Path 5: Invoking vehicular controls of BCM from the malicious application in IVI. This attack path is exploited on *Car E*.

Insights. This attack path demonstrates the feasibility of vehicle control from the backend server, a new attack surface in modern vehicles. These servers are responsible for handling sensitive information, such as personal and financial data, as well as controlling critical systems within the vehicle.

7.7. Path 5: From IVI malware to Car Control

The last attack path we demonstrate involves invoking the control of the BCM from a malicious application installed in the IVI. The corresponding logical attack path is shown in Fig.13. In the previous Path 2, we demonstrated that an attacker can launch remote vehicular control from malware installed on a victim's mobile device. However, as IVIs become more complex, many IVIs are now equipped with intelligent operating systems (e.g., Android or Android Automotive) that allow users to install various applications, introducing corresponding risks. Specifically, as shown in Fig.13, by exploiting a vulnerability in the installation process (e.g., lack of authentication on the APK file), an attacker can inject a malicious application into the IVI (Node.8). By further exploiting a vulnerability in the IVI, the attacker can escalate from low privilege to root code execution (Node.5), and finally send control commands to the BCM (Node.1).

PoC attack. The attack path is validated and exploited on *Car E*. Specifically, we identify that the IVI of *Car E* is implemented with the Android OS, allowing users to install Android applications. By conducting extensive security analysis on the IVI and sniffing the network, we identify a vulnerability that allows us to launch a Man-In-The-Middle (MITM) attack and inject a malicious app into the IVI (Node.8). The malicious app contains code to root the IVI using a corresponding vulnerability (Node.5) and finally invoke vehicle controls on the BCM, including unlocking doors and opening trunks (Node.1).

Insights. As modern vehicles become increasingly intelligent and connected, the IVI system grows more complex, making it crucial to ensure the security of in-vehicle applications. This attack path highlights the importance of securing the IVI and related applications, as vulnerabilities in these systems can lead to attackers gaining control over critical vehicle functions.

7.8. Responsible Disclosure

We have reported all vulnerabilities identified in *Car A*, *Car B*, *Car C*, and *Car D* to the respective manufacturers, and they have been promptly addressed. For *Car E*, we have recently reported the corresponding vulnerabilities and are awaiting a response. Consequently, we have anonymized the car brand and specific model of *Car E* for the time being. The details of our security analysis on these vehicles, including the security testing process, disclosure timeline, and implemented fixes, are available in [2].

7.9. Summary on Our Attacks

We would like to emphasize our contributions from the following aspects:

Attack path guidance. As identified in our interviews, conducting security activities (e.g., TARA and security testing) can be labor-intensive due to the absence of automated tools. In our experimental analysis, we demonstrated how attack paths generated by CarVal can aid security testing, showing that our automated tool is a valuable complement to the TARA methodology [6].

New attack surfaces. We explored a range of new attack surfaces that previous studies did not address, emphasizing the importance of protecting emerging attack surfaces as the automotive industry rapidly evolves. A detailed comparison with previous attack surfaces is provided in Appendix.D.

Multi-stage attack in complex IVNs. Our attack path reasoning and corresponding security analysis is based on the gateway-segmented IVN [36, 42] in five real vehicles, which is more complex than traditional architectures without gateways [21, 43]. Furthermore, increasingly complex and advanced IVNs are under development [1]. As such, CarVal can automatically reason attack paths in the context of these increasingly complex IVNs.

8. Discussion

Baseline risk values. Currently, the baseline "attack impact" and "attack feasibility" values (f_{EN}, f_{VN}, i_{VN}, f_{AN}, i_{AN} in Table.2) were manually assigned based on the specific context. For example, the feasibility to access the physical OBD-II port should be lower than the feasibility to access the wireless channel, and the impact brought by root execution is higher than that brought by low-privilege execution. Note that it is challenging to derive a set of universal or common baseline values that can be applied to all situations. This is because different groups may assign different baseline values to better suit their demand, and these impact and feasibility values can vary in different car models. Overall, these baseline values are flexible for users to set. Additionally, some research is focusing on scoring the individual automotive threat [14, 71], which can give guidelines about how to set up these baseline risk values.

Comparison with existing threat modeling tools. While previous threat modeling tools [3, 11, 28, 61] provide instructions on performing each sub-task of TARA (e.g., the 7 TARA tasks indicated by ISO 21434), they mainly focus on presenting manual workflows without offering automatic solutions. In comparison, CarVal leverages Datalog to automatically infer attack paths and assess corresponding risks, making the entire TARA process more efficient. Additionally, existing threat modeling tools [3, 11, 28, 61] often offer high-level guidance with limited use cases, leaving gaps in their practical application to real vehicles. In contrast, CarVal provides a specific solution to model the IVN using Datalog clauses, enabling automatic attack path reasoning and risk assessment. We also demonstrated how CarVal aids analysis using real vehicles as examples. Finally, note that CarVal is not intended to fully replace current threat modeling tools. Instead, it can be used in conjunction with other TARA approaches, including existing threat modeling tools. For instance, a team can first perform manual threat modeling of the automotive system (e.g., identifying assets and threat scenarios as indicated by ISO 21434 [6]), and then utilize CarVal to generate attack paths and assess risk values.

Limitation. It is important to acknowledge that not all identified limitations could be resolved within the scope of this work. The complexity of modern vehicles and the ever-evolving landscape of cybersecurity threats present a persistent challenge to the industry. For example, it is out of scope to propose a very clear threshold for how to mitigate the threats that all manufacturers must follow. It is important to recognize that fostering a strong cybersecurity culture and refining existing standards and regulations will require continued efforts from the automotive industry, regulatory bodies, and researchers. We believe that our improved threat database and CarVal approach have contributed to making the regulations more complete, and making the TARA process more efficient. Additionally, it is crucial to continuously update the threat database, and refine the automatic tool to stay ahead of emerging risks.

9. Conclusion

We conducted the first in-depth interview study with 15 experts working in automotive cybersecurity, revealing the specific challenges when security activities are being conducted, and the limitations of existing regulations. Particularly, we found that the threat cases given by current regulations are insufficient, and conducting TARA is often labor-intensive due to the lack of automatic tools. To address these challenges, we constructed a hierarchical threat database for automotive systems based on the interview data, improving the existing database both quantitatively and qualitatively. Moreover, we propose CarVal, a datalog-based approach that could generate multi-stage attack paths in IVN and calculate risk values. By applying CarVal to five real cars, we conducted extensive security analysis based on the generated attack paths, and successfully exploited corresponding attack chains in the new gateway-segmented IVN. In conclusion, our experimental analysis on real cars demonstrated the significant potential risks on new attack

surfaces emerging in modern vehicles. Moreover, the proposed database and methodology will shed light on how security activities (e.g., TARA and security testing) can be conducted more efficiently, as a supplement to existing regulations.

10. Acknowledgment

We thank the anonymous reviewers for their constructive comments on improving this research. We thank all experts who participated in our interview study, including Muchen Su, Zhiyu Zhang, Jianghui Guan from the security group of Avatr, and all other anonymous interviewees. This work is partly supported by HKPolyU Grant (No. ZVG0). Chenxiong Qian was partly supported by the NSFC for Young Scientists of China (No. 62202400).

References

[1] Automotive Zonal Architecture - Guardknox. https://www.guardknox.com/automotive-zonal-architecture/.

[2] Carval code and supplementary materials. https://github.com/VehicleCyberSec/CarVal.

[3] OWASP Threat Dragon. https://owasp.org/www-project-threat-dragon/.

[4] treelib: a tree data structure for Python. https://treelib.readthedocs.io/en/latest/.

[5] XSB: A Logic Programming and Deductive Database system. https://xsb.sourceforge.net/.

[6] Iso/sae 21434:2021: Road vehicles — cybersecurity engineering. https://www.iso.org/standard/70918.html, 2021.

[7] Un regulation no.155 - cyber security and cyber security management system. https://unece.org/sites/default/files/2021-03/R155e.pdf, 2021.

[8] AUTEL 919 professional diagnostic tools. https://item.jd.com/70636576685.html, 2023.

[9] Chinese Standard: GB/T 40861—2021 General technical requirements for vehicle cybersecurity. https://openstd.samr.gov.cn/bzgk/gb/newGbInfo?hcno=2977F0AC1719BBEFB9649C0146B0FC55, 2023.

[10] Jni: Java native interface. https://developer.android.com/training/articles/perf-jni, 2023.

[11] Threatget - threat analysis and risk management. https://www.threatget.com/, 2023.

[12] Udp: User datagram protocol. https://en.wikipedia.org/wiki/User_Datagram_Protocol, 2023.

[13] A. I. Alrabady and S. M. Mahmud. Analysis of attacks against the security of keyless-entry systems for vehicles and suggestions for improved designs. *IEEE transactions on vehicular technology*, 54(1):41–50, 2005.

[14] P. Bajpai and R. Enbody. Towards effective identification and rating of automotive vulnerabilities. In *Proceedings of the Second ACM Workshop on Automotive and Aerial Vehicle Security*, pages 37–44, 2020.

[15] M. Birks and J. Mills. *Grounded theory: A practical guide*. Sage, 2015.

[16] S. Bono, M. Green, A. Stubblefield, A. Juels, A. D. Rubin, and M. Szydlo. Security analysis of a cryptographically-enabled rfid device. In *USENIX Security Symposium*, volume 31, pages 1–16, 2005.

[17] H. Cai, V. W. Zheng, and K. C.-C. Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, 2018.

[18] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. A. Chen, M. Liu, and B. Li. Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 176–194. IEEE, 2021.

[19] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao. Adversarial sensor attack on lidar-based perception in autonomous driving. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 2267–2281, 2019.

[20] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou. Hidden voice commands. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 513–530, 2016.

[21] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno, et al. Comprehensive experimental analyses of automotive attack surfaces. In *USENIX Security Symposium*, volume 4, pages 447–462. San Francisco, 2011.

[22] K.-T. Cho and K. G. Shin. Error handling of in-vehicle networks makes them vulnerable. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1044–1055, 2016.

[23] J. Corbin and A. Strauss. *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage publications, 2014.

[24] G. Costantino, M. De Vincenzi, and I. Matteucci. A comparative analysis of unece wp. 29 r155 and iso/sae 21434. In *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 340–347. IEEE, 2022.

[25] J. Cui, L. S. Liew, G. Sabaliauskaite, and F. Zhou. A review on safety failures, security attacks, and available countermeasures for autonomous vehicles. *Ad Hoc Networks*, 90:101823, 2019.

[26] M. Dibaei, X. Zheng, K. Jiang, R. Abbas, S. Liu, Y. Zhang, Y. Xiang, and S. Yu. Attacks and defences on intelligent connected vehicles: a survey. *Digital Communications and Networks*, 6(4):399–421, 2020.

[27] T. Dougan and K. Curran. Man in the browser attacks. *International Journal of Ambient Computing and Intelligence (IJACI)*, 4(1):29–39, 2012.

[28] M. Ebrahimi, S. Marksteiner, D. Ničković, R. Bloem, D. Schögler, P. Eisner, S. Sprung, T. Schober, S. Chlup, C. Schmittner, et al. A systematic approach to automotive security. In *International Symposium on Formal Methods*, pages 598–609. Springer, 2023.

[29] D. A. Fernandes, L. F. Soares, J. V. Gomes, M. M. Freire, and P. R. Inácio. Security issues in cloud environments: a survey. *International Journal of Information Security*, 13(2):113–170, 2014.

[30] I. Foster, A. Prudhomme, K. Koscher, and S. Savage. Fast and vulnerable: A story of telematic failures. In *9th {USENIX} Workshop on Offensive Technologies ({WOOT} 15)*, 2015.

[31] A. Francillon, B. Danev, and S. Capkun. Relay attacks on passive keyless entry and start systems in modern cars. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. Eidgenössische Technische Hochschule Zürich, Department of Computer Science, 2011.

[32] M. Frigault and L. Wang. Measuring network security using bayesian network-based attack graphs. In *2008 32nd Annual IEEE International Computer Software and Applications Conference*, pages 698–703. IEEE, 2008.

[33] F. D. Garcia, D. Oswald, T. Kasper, and P. Pavlidès. Lock it and still lose it—on the (in) security of automotive remote keyless entry systems. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016.

[34] J. M. Haney and W. G. Lutters. " it's {Scary... It's}{Confusing... It's} dull": How cybersecurity advocates overcome negative perceptions of security. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*, pages 411–425, 2018.

[35] T. Hoppe, S. Kiltz, and J. Dittmann. Security threats to automotive can networks—practical examples and selected short-term countermeasures. *Reliability Engineering & System Safety*, 96(1):11–25, 2011.

[36] S. Hu, Q. Zhang, A. Weimerskirch, and Z. M. Mao. Gatekeeper: A gateway-based broadcast authentication protocol for the in-vehicle ethernet. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pages 494–507, 2022.

[37] A. Humayed, J. Lin, F. Li, and B. Luo. Cyber-physical systems security—a survey. *IEEE Internet of Things Journal*, 4(6):1802–1831, 2017.

[38] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514, 2021.

[39] P. Jing, Q. Tang, Y. Du, L. Xue, X. Luo, T. Wang, S. Nie, and S. Wu. Too good to be safe: Tricking lane detection in autonomous driving with crafted perturbations. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[40] H. J. Jo, W. Choi, S. Y. Na, S. Woo, and D. H. Lee. Vulnerabilities of android os-based telematics system. *Wireless Personal Communications*, 92(4):1511–1530, 2017.

[41] E. Kenneally and D. Dittrich. The menlo report: Ethical principles guiding information and communication technology research. *Available at SSRN 2445102*, 2012.

[42] J. H. Kim, S.-H. Seo, N.-T. Hai, B. M. Cheon, Y. S. Lee, and J. W. Jeon. Gateway framework for in-vehicle networks based on can, flexray, and ethernet. *IEEE Transactions on Vehicular Technology*, 64(10):4472–4486, 2014.

[43] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, et al. Experimental security analysis of a modern automobile. In *2010 IEEE Symposium on Security and Privacy*, pages 447–462. IEEE, 2010.

[44] S. Kulandaivel, S. Jain, J. Guajardo, and V. Sekar. Cannon: Reliable and stealthy remote shutdown attacks via unaltered automotive microcontrollers. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 195–210. IEEE, 2021.

[45] Y. L. Le Yu, P. Jing, X. Luo, L. Xue, K. Zhao, Y. Zhou, T. Wang, G. Gu, S. Nie, and S. Wu. Towards automatically reverse engineering vehicle diagnostic protocols. In *Proc. USENIX Security*, 2022.

[46] D. Lyu, L. Xue, and X. L. Le Yu. Remote attacks on vehicles by exploiting vulnerable telematics, 2016.

[47] C. Miller and C. Valasek. Adventures in automotive networks and control units. *Def Con*, 21(260-264):15–31, 2013.

[48] C. Miller and C. Valasek. A survey of remote automotive attack surfaces. *black hat USA*, 2014:94, 2014.

[49] C. Miller and C. Valasek. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA*, 2015(S 91), 2015.

[50] B. Nassi, Y. Mirsky, D. Nassi, R. Ben-Netanel, O. Drokin, and Y. Elovici. Phantom of the adas: Securing advanced driver-assistance systems from split-second phantom attacks. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 293–308, 2020.

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

[52] X. Ou, S. Govindavajhala, A. W. Appel, et al. Mulval: A logic-based network security analyzer. In *USENIX security symposium*, volume 8, pages 113–128. Baltimore, MD, 2005.

[53] X. Ou and A. Singhal. Attack graph techniques. In *Quantitative Security Risk Assessment of Enterprise Networks*, pages 5–8. Springer, 2012.

[54] H. Palombo, A. Z. Tabari, D. Lende, J. Ligatti, and X. Ou. An ethnographic understanding of software ({In) Security} and a {Co-Creation} model to improve secure software development. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*, pages 205–220, 2020.

[55] L. Pan, X. Zheng, H. Chen, T. Luan, H. Bootwala, and L. Batten. Cyber security attacks to modern vehicular systems. *Journal of information security and applications*, 36:90–100, 2017.

[56] J. Petit and S. E. Shladover. Potential cyberattacks on automated vehicles. *IEEE Transactions on Intelligent transportation systems*, 16(2):546–556, 2014.

[57] L. T. professional diagnostic tools. X-431 Pad III. https://launchtechusa.com/new-product-x431-pad3/, 2023.

[58] K. Ren, Q. Wang, C. Wang, Z. Qin, and X. Lin. The security of autonomous driving: Threats, defenses, and future directions. *Proceedings of the IEEE*, 108(2):357–372, 2019.

[59] I. Rouf, R. D. Miller, H. A. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar. Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study. In *USENIX Security Symposium*, volume 10, 2010.

[60] T. Sato, J. Shen, N. Wang, Y. Jia, X. Lin, and Q. A. Chen. Dirty road can attack: Security of deep learning based automated lane centering under {Physical-World} attack. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 3309–3326, 2021.

[61] C. Schmittner, S. Chlup, A. Fellner, G. Macher, and E. Brenner. Threatget: Threat modeling based approach for automated and connected vehicle systems. In *AmE 2020-Automotive meets Electronics; 11th GMM-Symposium*, pages 1–3. VDE, 2020.

[62] K. Serag, R. Bhatia, V. Kumar, Z. B. Celik, and D. Xu. Exposing new vulnerabilities of error handling mechanism in {CAN}. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[63] J. Sun, Y. Cao, Q. A. Chen, and Z. M. Mao. Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 877–894, 2020.

[64] J. Takahashi and T. Fukunaga. Implementation attacks on an immobilizer protocol stack. In *11th Embedded Security in Cars Conference Europe, escar Europe*, 2013.

[65] A. Tuladhar, D. Lende, J. Ligatti, and X. Ou. An analysis of the role of situated learning in starting a security culture in a software company. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*, pages 617–632, 2021.

[66] R. Verdult, F. D. Garcia, and J. Balasch. Gone in 360 seconds: Hijacking with hitag2. In *21st {USENIX} Security Symposium ({USENIX} Security 12)*, pages 237–252, 2012.

[67] R. Verdult, F. D. Garcia, and B. Ege. Dismantling megamos crypto: Wirelessly lockpicking a vehicle immobilizer. In *Supplement to the Proceedings of 22nd {USENIX} Security Symposium (Supplement to {USENIX} Security 15)*, pages 703–718, 2015.

[68] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia. An attack graph-based probabilistic security metric. In *Data and Applications Security XXII: 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security London, UK, July 13-16, 2008 Proceedings 22*, pages 283–296. Springer, 2008.

[69] L. Wang, S. Jajodia, A. Singhal, A. Singhal, and X. Ou. *Security risk analysis of enterprise networks using probabilistic attack graphs*. Springer, 2017.

[70] Q. Wang, Z. Mao, B. Wang, and L. Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[71] Y. Wang, B. Yu, H. Yu, L. Xiao, H. Ji, and Y. Zhao. Automotive cybersecurity vulnerability assessment using the common vulnerability scoring system and bayesian network model. *IEEE Systems Journal*, 2022.

[72] H. Wen, Q. A. Chen, and Z. Lin. Plug-n-pwned: Comprehensive vulnerability analysis of obd-ii dongles as a new over-the-air attack surface in automotive iot. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 949–965, 2020.

[73] S. Woo, H. J. Jo, and D. H. Lee. A practical wireless attack on the connected car and security protocol for in-vehicle can. *IEEE Transactions on intelligent transportation systems*, 16(2):993–1006, 2014.

[74] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 103–117, 2017.

Appendix A. Interview Details

Ethical considerations. Before each interview, we explained our data collection goals, the anonymization process for the collected data, and began recording audio only after obtaining permission. All interviewees had approval from their respective organizations to participate in the interviews. They were fully aware of our data collection process and purpose, and consented to our data collection. We specifically encouraged interviewees not to mention information

that could lead to potential ethical issues. We also meticulously examined the transcribed texts to ensure they did not contain sensitive information, such as company names or vehicle model names. Moreover, all collected data were securely stored on encrypted cloud storage, in accordance with the General Data Protection Regulation (GDPR) and previous studies [41].

Recruitment Details. We first started the recruitment by both personal contact and sending invitation Emails. For each contacted expert, we consulted whether they were willing to participate in our study, and provided a shopping card as a token of appreciation. Particularly, we also use snow sampling in our recruitment: at the end of each interview, we asked whether the participant could introduce other experts to join our interview. Additionally, we employed snowball sampling in our recruitment strategy. At the end of each interview, we asked participants if they could introduce other experts who might be interested in joining our study. This snowball sampling technique helped us expand our pool of interviewees in two ways: (1) by reaching experts within the same company (e.g., P11 and P12 were introduced by P10) and (2) by including participants from different companies (e.g., P13 was introduced by P8).

Diversity of the participants. As shown in Table.1, 15 experts were from 9 different companies. However, we emphasize that, even though some participants were from the same company, they have different job duties and also provided diverse data to our study. For example, both P4 and P5 were from Company.4 and worked on security testing, but their specific testing tasks (e.g., testing different ECUs) varied, allowing them to provide different insights and amendments to our threat database. Furthermore, participants P10, P11, and P12 all belonged to Company.8, but they had different job roles (e.g., P10 focused more on team management, while P11 and P12 focused on specific testing and TARA), providing valuable insights from different perspectives.

Consistency between earlier and later interviews. Although the interview protocol wasn't finalized until round 10, the main content and interview flow remain consistent. Specifically, the structure of the interview protocol remained unchanged throughout the whole interview process, and the content in Sec.4 and Sec.5 remain unchanged after the 3rd round of interview. From the 4th to the 10th round, we made minor revisions in Sections 3 and 6, which involved refining the texts and adding more questions to facilitate open-ended discussions. Overall, the improvements on the interview protocol did not break the consistency between earlier and later interviews.

Appendix B. Relation Analysis among Threat Codes

To better illustrate the relations among the threat codes, the 28 codes in Fig.4 are classified into the following two types:

Entity code. An entity code represents a specific automotive *component* carrying functions that can introduce

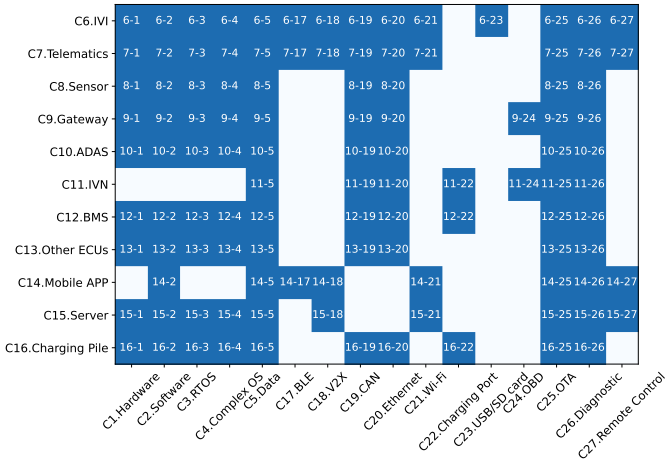


Figure 14: Relation analysis: Adjacent matrix showing the relation between the entity codes (on Y axis) and property codes (on X axis). Marked cell represents that the corresponding triplet is identified between the entity code and property code.

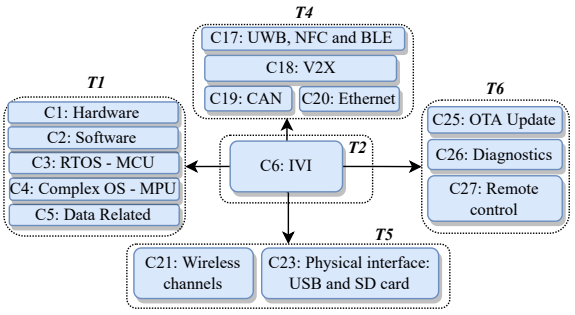


Figure 15: Part of the TKG (14 triplets) showing the property codes related to the entity code: *IVI*. This representation is equal to the first row in Fig.14.

security threats. The entity codes are often the object component in TARA or security testing. Specifically, all 11 codes under *T2: In-Vehicle Components*, and *T3: Outside-Vehicle Components* are the entity codes.

Property code. A property code represents one specific security property residing in one entity code. Specifically, all 17 codes except the 11 entity codes in T2 and T3 are the property codes.

With the above classification, the triplet to build the knowledge graph [17, 38, 70] is further presented as *(entity code, is vulnerable to threats in, property code)*. Finally, we constructed 109 triplets between the 11 entity codes and 17 property codes, and the result is shown in the form of an adjacent matrix in Fig.14. Specifically, in Fig.14, the X axis represents the property codes and the Y axis represents the entity codes. Each marked cell represents that the corresponding triplet is identified. For example, the cell *(C6, C4)* is marked, which means that the threats in *C4: Complex OS* and reside in the entity *C6: IVI*.

Specifically, Fig.15 shows part of the knowledge graph, originating from the entity code *C6.IVI*. This figure represents the threat codes that should be considered when

evaluating IVI security. For example, it is very likely that an IVI is equipped with a complex OS (e.g., Linux, Android) to perform various infotainment functions, and thus there is a triplet connecting *C6.IVI* and *C4.Complex OS*. As a result, such a graphical representation can make the TARA and security testing more systematic and comprehensive. Taking Fig.15 as an example, when evaluating the security of the IVI ECU, the threats under other property codes (i.e., the threat codes in T1, T4, T5 and T6 in Fig.15) should also be considered to build a reliable security baseline for this ECU.

These connections between threat codes enable users to understand the interdependencies among automotive threats, aiding them in building CarVal Datalog rules specific to their car models. Furthermore, users have the flexibility to modify and create their own relationships based on different car implementations.

Appendix C.
CarVal Risk Assessment

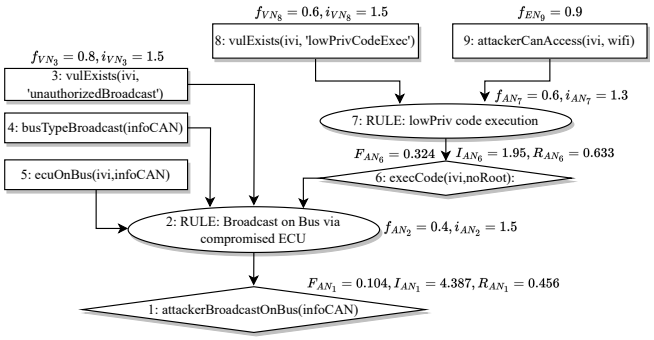


Figure 16: Example 1: Compromise the infoCAN via the browser attack surface on IVI. This attack path has the medium attack impact and comparatively high feasibility, leading to a high risk value.

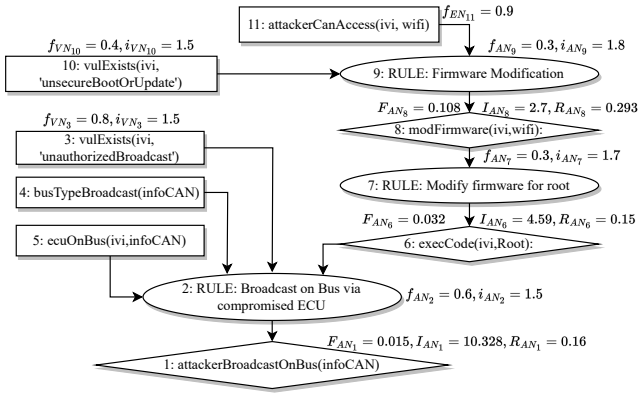


Figure 17: Example 2: Compromise the infoCAN via re-writing the firmware on IVI, though the vulnerable OTA or booting process. Compared with Path-1, this attack path has higher attack impact but comparatively lower feasibility, finally leading to a lower risk value.

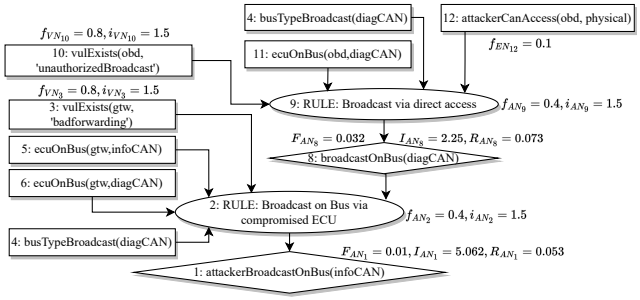


Figure 18: Example 3: Compromise the infoCAN via physically accessing OBD-II port, and compromise the forwarding process of gateway ECU. Comparatively, this attack path has medium attack impact but the lowest feasibility, and finally leading to the lowest risk value among three.

In this section, we demonstrate how CarVal can perform automatic risk assessment based on the generated attack paths. Specifically, we have generated three attack paths based on the IVN of a real vehicle. For all generated attack paths, the baseline value of the calculated risk (R_{base} in Equation.6.2.3) has been set to 0.001. There are the following three attack paths:

Path-1: Compromise the infoCAN via the browser attack surface on IVI (Fig.16). This attack path has the medium attack impact (4.387) and comparatively high feasibility (0.104), leading to a high risk value (0.456).

Path-2: Compromise the infoCAN via re-writing the firmware on IVI, though the vulnerable OTA or booting process (Fig.17). Compared with Path-1, this attack path has higher attack impact (10.328) but comparatively lower feasibility (0.015), finally leading to a lower risk value (0.16).

Path-3: Compromise the infoCAN via physically accessing OBD-II port, and compromise the forwarding process of gateway ECU (Fig.18). Comparatively, this attack path has medium attack impact (5.062) but the lowest feasibility (0.01), and finally leading to the lowest risk value (0.053) among three.

Overall, the three attack paths share the same attack goal, but this goal was achieved via different means. As a result, CarVal assigns different values for attack impact, feasibility, and risk to each path. Users can conduct further analysis based on these values, such as focusing more on impact or feasibility, depending on their specific requirements.

Appendix D.
Revisiting Previous Attack Surfaces

Research studies related to the cybersecurity of modern vehicles are prospering these years, and there are a series of related surveys as the milestones [25, 26, 37, 48, 55, 56, 58]. After going through the surveys and investigating the related works, we compare our research with previously discovered attack surfaces in Tab.3. Note that there are also many studies focusing on attacking the sensors to affect the behaviors of the autonomous driving systems [18, 19, 39, 50, 60, 63].

TABLE 3: Comparison with previously discovered cyberattacks on modern vehicles. - *Attack capability*: ○: Affect trivial functions; ◐: Perform limited car controls; ●: Perform safety-critical car controls. - *Real car?*: ○: Simulation; ◐: Testbed; ●: Real cars.

Ref	Attack Surface	Attack capability	Real car?	Bypass gateway?
[35]	OBD-II	●	●	✗
[43]	OBD-II	●	●	✗
[47]	OBD-II	●	●	✗
[22]	OBD-II	◐	●	✗
[62]	OBD-II	◐	●	✗
[21]	CD, Bluetooth, Cellular	●	●	✗
[49]	USB, Wi-Fi, Cellular	●	●	✗
[59]	TPMS	◐	●	✗
[16, 31, 64, 66, 67]	Immobilizer	◐	●	✗
[13, 33]	PKES	◐	●	✗
[20, 74]	Speech recognition system	○	●	✗
[46]	Mobile APP (OBD-II dongle)	◐	◐	✗
[73]	Mobile APP (OBD-II dongle)	◐	●	✗
[72]	Mobile APP (OBD-II dongle)	◐	○	✗
[40]	Telematics	◐	●	✗
[30]	OBD-II dongle	●	○	✗
[44]	A compromised ECU	◐	●	✗
Ours	Mobile APP (Official), IVI browser, backend server, IVI Malware	●	●	✓

Limitations. As shown in Table.3, various attack surfaces have been explored in previous research. However, they suffer from the following limitations. First, they failed to consider emerging attack surfaces due to automotive user interfaces (e.g., mobile app, in-vehicle browser, server and IVI malware as we exploited). In addition. *none* of them have taken into account the in-vehicle network (IVN) topology, leading to two drawbacks: 1). the old attack may not function on the new IVN (e.g., when there is a gateway protection), and 2). potential attack paths could be neglected, due to the lack of a comprehensive understanding of the IVN topology.

Appendix E. Meta-Review

E.1. Summary

This paper discusses that existing standards and regulations are insufficient to enable security in modern vehicles. The authors interviewed experts from automotive cybersecurity teams to understand the limitations of regulations and research gaps. The paper provides an improved threat database and an approach CarVal to infer multi-stage attack paths. The proposed approach uncovers new attack surfaces. An extensive study is done using five different car models.

E.2. Scientific Contributions

- Provides a Valuable Step Forward in an Established Field.
- Creates a New Tool to Enable Future Science.
- Identifies an Impactful Vulnerability.

E.3. Reasons for Acceptance

1) The paper provides a valuable step forward in an established field. The paper identifies key challenges in the security process for automotive systems and produces a novel solution to the automation of the threat analysis and risk assessment process for this domain.
2) The paper creates a new tool to enable future science. The authors make their tool, database, and resulting codes available to other researchers, allowing for both independent confirmation and as a platform for future research. The tool can also be used directly by car manufacturers to improve security broadly in the community.
3) This paper identifies an impactful vulnerability. Using CarVal, the paper identifies five attack paths not diagnosed in previous research and developed PoC attacks for five real cars.

E.4. Noteworthy Concerns

The interviews are conducted with 15 experts, which is a sufficient number for exploratory work, however, many experts work for the same company (N=6, three participants work for C4 and three work for C8). This overlap introduces some concerns that responses might be more likely to converge, reducing the true saturation of themes in the interviews. This is mitigated to some extent by the fact that almost all participants from the same company worked in different roles, therefore, they still provided unique insights during interviews.