applied sciences

MDPI

Article

ICVTest: A Practical Black-Box Penetration Testing Framework for Evaluating Cybersecurity of Intelligent Connected Vehicles

Haichun Zhang [1,2], Jie Wang [2,3,*], Yijie Wang [4], Minfeng Li [2], Jinghan Song [2] and Zhenglin Liu [4]

[1] School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China; zhanghaichun@hust.edu.cn
[2] Shenzhen Kaiyuan Internet Security Technology Co., Ltd., Shenzhen 518000, China; liminfeng@seczone.cn (M.L.); songjinghan@seczone.cn (J.S.)
[3] School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China
[4] School of Integrated Circuits, Huazhong University of Science and Technology, Wuhan 430074, China; wunyje@hust.edu.cn (Y.W.); liuzhenglin@hust.edu.cn (Z.L.)
* Correspondence: wangjie@seczone.cn

Abstract: Intelligent connected vehicles (ICVs) are equipped with extensive electronic control units which offer convenience but also pose significant cybersecurity risks. Penetration testing, recommended in ISO/SAE 21434 "Road vehicles—Cybersecurity engineering", is an effective approach to identify cybersecurity vulnerabilities in ICVs. However, there is limited research on vehicle penetration testing from a black-box perspective due to the complex architecture of ICVs. Additionally, no penetration testing framework has been proposed to guide security testers on conducting penetration testing for the whole vehicle. The lack of framework guidance results in the inexperienced security testers being uncertain about the processes to follow for conducting penetration testing. Moreover, the inexperienced security testers are unsure about which tests to perform in order to systematically evaluate the vehicle's cybersecurity. To enhance the penetration testing efficiency of ICVs, this paper presents a black-box penetration testing framework, ICVTest. ICVTest proposes a standardized penetration testing process to facilitate step-by-step completion of the penetration testing, thereby addressing the issue of inexperienced testers lacking guidance on how to initiate work when confronted with ICV. Also, ICVTest includes 10 sets of test cases covering hardware and software security tests. Testers can select appropriate test cases based on the specific cybersecurity threats faced by the target object, thereby reducing the complexity of penetration testing tasks. Furthermore, we have developed a vehicle cybersecurity testing platform for ICVTest that seamlessly integrates various testing tools. The platform enables even novice testers to conduct vehicle black-box penetration testing in accordance with the given guidance which addresses the current industry's challenge of an overwhelming number of testing tasks coupled with a shortage of skilled professionals. For the first time, we propose a comprehensive black-box penetration testing framework and implement the framework in the form of a cybersecurity testing platform. We apply ICVTest to evaluate an electric vehicle manufactured in 2021 for assessing the framework's availability. With the aid of ICVTest, even testers with limited experience in automotive penetration can effectively evaluate the security risks of ICVs. In our experiments, numerous cybersecurity vulnerabilities were identified involving in-vehicle sensors, remote vehicle control systems, and in-vehicle controller area network (CAN) bus.

Keywords: intelligent connected vehicles; penetration testing; black-box; cybersecurity; test case set

check for updates

Citation: Zhang, H.; Wang, J.; Wang, Y.; Li, M.; Song, J.; Liu, Z. ICVTest: A Practical Black-Box Penetration Testing Framework for Evaluating Cybersecurity of Intelligent Connected Vehicles. *Appl. Sci.* **2024**, *14*, 204. https://doi.org/10.3390/app14010204

Academic Editor: Luis Javier García Villalba

Received: 6 November 2023
Revised: 13 December 2023
Accepted: 20 December 2023
Published: 25 December 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

While intelligent connected vehicles (ICVs) bring convenience to transportation, they also complicate the vehicle's architecture which introduces serious cybersecurity risks [1]. The in-vehicle systems in ICVs, which contain software and hardware, expose numerous cybersecurity attack surfaces [2,3]. The wireless communication interfaces between ICVs and the external internet enables remote attacks [4–6]. Bus transmission protocols without

security mechanisms create favourable conditions for attackers to control vehicles [7–10]. Inadequate security protection regarding in-vehicle sensors pose significant cybersecurity challenges to autonomous driving [11–13]. Cybersecurity threats increase the possibility of malicious attacks, triggering numerous cybersecurity attacks, and endangering the safety of vehicles and people.

To tackle cybersecurity challenges in the automotive industry, the ISO/SAE 21434 "Road vehicles—Cybersecurity engineering" [14] standard provides detailed specifications for cybersecurity activities throughout all stages of the vehicle life cycle to prevent potential cybersecurity risks. As a result, ISO/SAE 21434 has emerged as the industry's prevailing standard for managing cybersecurity. Both automobile manufacturers and suppliers systematically assess product's cybersecurity through penetration testing during development and validation phases to ensure compliance with vehicle type approval (VTA).

Penetration testing is a crucial approach for identifying potential cybersecurity vulnerabilities in ICVs throughout their entire life cycle. Traditional functional testing often overlooks unexpected behaviors, while penetration testing evaluates the system's security performance against attacks. The objective of penetration testing is to ensure consistency between the realization and design of security goals [15]. Investigating the penetration testing of ICVs holds significant importance in discovering cybersecurity vulnerabilities, reducing the likelihood of accidents, and safeguarding lives and property.

Based on information acquired about the target of test (TOT), penetration testing can be categorized into white-box tests, gray-box tests, and black-box tests. In terms of the TOT, penetration testing in the field of ICVs can be divided into vehicle penetration testing (VPT) and in-vehicle system penetration testing (IVSPT). Among these test scenarios, black-box VPT is considered most complex. On one hand, testers have limited access to public information about the TOT during black-box penetration testing scenarios. On the other hand, compared to in-vehicle systems, vehicles have fewer external interfaces available to access. The limited availability of information and interfaces poses significant challenges to conducting comprehensive black-box VPT research as follows.

Firstly, the absence of a standardized cybersecurity testing process in the automotive industry hinders the formalization of black-box VPT. Although the utilization of penetration testing for identifying potential cybersecurity vulnerabilities has emerged as a crucial cybersecurity practice, the execution process of conducting penetration testing varies across different organizations. The absence of a standardized penetration testing process within the automotive industry results in the inexperienced testers lacking guidance on how to initiate such security assessments.

Secondly, due to the intricate structure of ICVs, testers require advanced expertise to conduct VPT effectively. The development of penetration testing schemes may vary among testers with different levels of experience, leading to divergent conclusions in cybersecurity evaluations. In the absence of proper guidance, inexperienced testers lack the knowledge to select and execute appropriate test cases for a systematic evaluation of vehicle cybersecurity.

Additionally, due to the emerging nature of the technology field, there is a dearth of an integrated cybersecurity testing platform for comprehensive penetration testing of entire vehicles in the automotive industry. The automotive architecture is highly intricate, encompassing over 100 cybersecurity test cases. Without an integrated platform to manage and guide numerous penetration testing tasks, it becomes arduous to effectively execute and oversee penetration testing and management endeavors.

Cybersecurity is an emerging technological domain within the automotive industry. Traditional automobile manufacturers are still grappling with the challenges of transitioning towards intelligence and electrification, resulting in limited investment in cybersecurity initiatives. However, as automotive cybersecurity laws, regulations, and technical standards are gradually being implemented, both manufacturers and suppliers find themselves confronted with a significant contradiction between the demand for penetration testing and the shortage of skilled personnel. Addressing key challenges associated with penetration testing and proposing a systematic guidance framework for cybersecurity testing to

facilitate efficient standardized black-box VPT have become crucial strategies to resolve the contradiction between penetration testing requirements and technician shortages. The above motivation behind this paper leads to the proposal of the black-box penetration testing framework (ICVTest). The main contributions of ICVTest are as follows:

At first, ICVTest introduces a standardized six-step process to address the issue of testers lacking guidance when conducting VPT with limited information. Guided by the six-step penetration testing process, testers are equipped with the knowledge to initiate a VPT task and meticulously follow the guidelines to systematically accomplish each step, thereby facilitating comprehensive VPT.

Next, ICVTest presents 10 hierarchical test case sets that draw from extensive experience across various vehicle types and can be applied universally. Testers only need to select appropriate test cases from 10 predefined test case sets based on the target's characteristics, enabling systematic evaluation of its security. Predefined hierarchical test case sets ensure consistency among different testers' results and facilitate rapid development of penetration testing capabilities for relevant organizations.

Finally, we have developed an integrated cybersecurity testing platform for ICVTest, which encompasses testing infrastructure and testing tools. The platform incorporates a comprehensive test guidance file that assists testers in conducting VPT according to the provided instructions, thereby reducing the skill requirements for penetration testing personnel. The platform offers advanced functionalities such as testing tool management, test case management, and project management to ensure scalability.

To the best of our knowledge, our proposed comprehensive framework for black-box VPT based on extensive industry experience is pioneer in this filed. Additionally, we have successfully implement an integrated cybersecurity testing platform for the framework. Compared with the existing security testing frameworks in the automotive field, we not only recommend penetration testing cases that need to be performed for security evaluation, but also standardize the process of carrying out penetration testing. In terms of test cases, we abstract automotive security test scenarios into 10 sets that correspond to different fields of software and hardware architecture in vehicles which allows for the reuse of test cases, thereby enhancing the versatility of ICVTest. The ability of 10 fields of vehicle cybersecurity threat modeling is not available in other frameworks.

A practical case study was conducted to demonstrate the application of ICVTest in physical VPT. Leveraging the capabilities of ICVTest, testers efficiently devised a comprehensive black-box penetration testing scheme tailored to address potential cybersecurity risks associated with the vehicle. As a result of applying appropriate test cases, testers successfully identified multiple vulnerabilities pertaining to the in-vehicle CAN bus, infotainment system, sensors, and remote vehicle control. This case study exemplifies how ICVTest can expedite standardized vehicle black-box penetration testing procedures while significantly mitigating the challenges encountered during such evaluation.

2. Related Work

Due to the importance of penetration testing for ICVs, researchers have carried out a large number of related studies. The vehicle-based penetration testing method primarily focuses on in-vehicle systems. Prathap et al. [16] introduced penetration testing to the in-vehicle ECU, considering it as an embedded system and adapting the existing penetration testing method for embedded systems. However, with the increasing intelligence of vehicles, certain ECUs such as in-vehicle infotainment systems have become highly complex. Traditional embedded system penetration testing methods may not be fully applicable to new in-vehicle ECUs. Talebi et al. proposed a penetration testing scheme for the automotive CAN bus [17]. The researchers conducted an analysis of the cybersecurity threats faced by the CAN bus and evaluated its resilience against data monitoring attacks, injection attacks, replay attacks, and tampering attacks. However, it is worth noting that their experimental environment was limited to simulation settings which may not fully reflect the real-world performance of penetration testing on actual vehicles. Ebert et al. focused on

conducting penetration testing, specifically targeting in-vehicle infotainment systems [18]. They employed threat analysis results to guide gray-box penetration testing and divided the process into 10 distinct steps. Nevertheless, this study lacks detailed information regarding both threat analysis and penetration testing methodologies employed by the researchers. Furthermore, no specific procedures or standards were established for performing these analyses and tests.

The threat-based penetration testing method integrates threat analysis with penetration testing, utilizing the results of the threat analysis to formulate a tailored penetration testing scheme for ICVs [19–21]. Dürrwangin et al. demonstrated the effectiveness of this method by applying it to the airbag control ECU [19]. However, limited research has been conducted on formulating a standardized penetration testing scheme. Bayer et al. introduced the fundamental steps of threat-based penetration testing for ICVs [20] yet a standardized process has not been established, which hinders effective guidance for testers in conducting standardized penetration tests. Mahmood et al. employed this method to systematically assess and test the security of an on-board OTA system [21] but failed to consider comprehensive vehicle-wide penetration testing.

The model-based penetration testing method automatically generates test cases by utilizing the TOT [22]. This approach employs a formal language to represent automotive networks and bus systems [23], enabling testers to construct attack models using the same formal language. As a result of leveraging these attack models, researchers can generate cybersecurity test cases for in-vehicle networks and bus systems. However, despite providing a logical description method for constructing attack models and generating test cases, the model-based penetration testing method lacks tools that can automatically generate executable code for the generated test cases based on formal attack models. To address this limitation, Mahmood et al. proposed a penetration testing method specifically designed for in-vehicle over-the-air (OTA) systems [24]. Their approach utilizes an attack tree model to represent OTA system threats and automates the generation of corresponding test cases, resulting in significant time saving. Nevertheless, the test cases generated automatically are not complete and accurate, and it remains unclear how well this penetration testing method performs when applied to other in-vehicle systems.

In addition to the security research on vehicles and subsystems, some researchers have proposed cybersecurity testing and evaluation frameworks for different in-vehicle systems. S. Li et al. proposed a security evaluation framework for in-vehicle infotainment systems based on threat analyses and penetration tests [25]. The framework conducts a comprehensive threat analysis for automotive in-vehicle infotainment (IVI) systems and performs a thorough security assessment of the identified threats. However, it does not primarily focus on penetration testing and lacks practical implementation examples. F. Luo et al. present a cybersecurity testing method which extends the penetration testing execution standard (PTES) from the perspective of testing processes [26]. However, the testing tool developed in this framework only supports in-vehicle CAN bus testing. K. He et al. proposed a comprehensive and implementable test evaluation method based on the professional darkroom [27], but a detailed test process and implementation was not provided.

While previous framework has extensively focused on specific in-vehicle systems like controller area network (CAN) bus and in-vehicle infotainment systems, there remains some comprehensive vehicle-wide penetration testing framework. Q. Li et al. proposed a security evaluation framework for ICVs based on attack chains [28]. The framework focuses on the generation mechanism and steps of attacks, systematically evaluates the current risk level and extent of damage in a targeted manner, and devises appropriate security management measures based on the severity of harm. However, the framework primarily focuses on threat evaluation, with less emphasis on penetration testing. Whether the threat is really harmful needs to be tested and verified. Shirvani et al. proposed security risk assessment framework for electric vehicles (EVs) [29]. In the framework, the evaluation of vehicle cybersecurity is conducted based on five components: charging station security, information privacy, software security, connected vehicle security, and autonomous driving

security. The framework only gives the threat points that need to be concerned about in security evaluation and does not give the process of penetration testing. Although a risk assessment case is given based on the framework, a cybersecurity test platform is not implemented to assist testers in carrying out security assessment. Schönhärl et al. established an automotive penetration testing education platform [30]. It consists of three layers representing different attack points of a vehicle: outer, inner, and core layers. The layer of threats lacks the necessary granularity, and the platform is limited to educational scenarios, rendering it unsuitable for actual penetration testing scenarios.

In summary, due to the intricate nature of vehicle architecture, most penetration testing research focuses on specific in-vehicle subsystems. However, in response to the growing demand for vehicle cybersecurity, some researchers have proposed frameworks for evaluating and testing vehicle cybersecurity. Nevertheless, the current security assessment framework primarily emphasizes threat modeling of the entire vehicle and develops risk assessment models based on factors such as severity of damage and likelihood of attack. Nonetheless, TARA represents a theoretical analysis activity applied in the automotive development process's design phase. The conversion of threats into actual exploitable vulnerabilities needs verification through penetration testing. Therefore, research on vehicle penetration testing and test platforms remains highly significant.

3. Black-Box Penetration Testing Framework for ICVs

3.1. Penetration Testing

Penetration testing is an efficient technique that helps security testers identify cybersecurity vulnerabilities at the end of development. Rather than simply testing, penetration testing involves systematically conducting security tests on target applications, networks, systems, and other entities to verify the presence of security vulnerabilities. Any exploitable security vulnerabilities discovered are then reported to the system owner. Penetration testing can be categorized into white-box, gray-box, and black-box based on the level of knowledge about the target [30]. White-box testing allows testers to access key data such as source code, internal data, and documentation. Black-box testing is a traditional method where testers can only gather information through public channels. Gray-box testing falls in between these two methods, which means testers are able to obtain some information about the architecture and cybersecurity threats of the target before conducting the test.

Typical penetration testing consists of three stages: the information collection stage, the threat analysis stage, and the penetration testing stage [31]. (1) The information gathering stage involves collecting available information within the limits set by authority. For instance, it includes searching open-source technical documentation for implementation details of the tested object. As more information is gathered gradually, testers gain a deeper understanding of the tested object which allows them to iterate on their information acquisition methods. The expansion in ways to gather information lays a foundation for system threat analysis and subsequent penetration testing. (2) The threat analysis stage necessitates the modeling of threats on the target under examination based on the gathered information. Testers deduce the fundamental system architecture of the target by considering factors such as software and hardware composition, functional features, open services, and other relevant details. As a result of analyzing system architecture, testers identify high-value assets that are more enticing to potential attackers. For high-value assets, testers scrutinize their threat scenarios encompassing attack entry points and paths. Building upon this foundation, testers assess the potential harm posed by each threat and evaluate the severity of cybersecurity risks using indicators like attack probability, property loss, and privacy disclosure. Based on the results of threat analysis, different severity levels are assigned to threat scenarios to assist testers in determining test case priorities. (3) The penetration testing stage requires a test plan tailored to address specific threats identified during the threat analysis stage in order to verify if these threats can be exploited into actual cybersecurity vulnerabilities. Executing test cases according to the plan helps uncover potential cybersecurity vulnerabilities. However, it is crucial to verify discovered

vulnerabilities in order to assess their level of harm. All vulnerabilities detected through penetration testing should be reported and managed following vulnerability management regulations while conducting regression testing after technical fixes have been implemented by the organization responsible for maintaining the tested system.

3.2. Penetration Testing Process in ICVTest

Providing a concise summary of the vehicle penetration testing process tailored for ICVs holds significant importance in standardizing test procedures. Based on the NIST three-stage penetration testing specification [31], ICVTest outlines six distinct steps for conducting VPT, as illustrated in Figure 1.

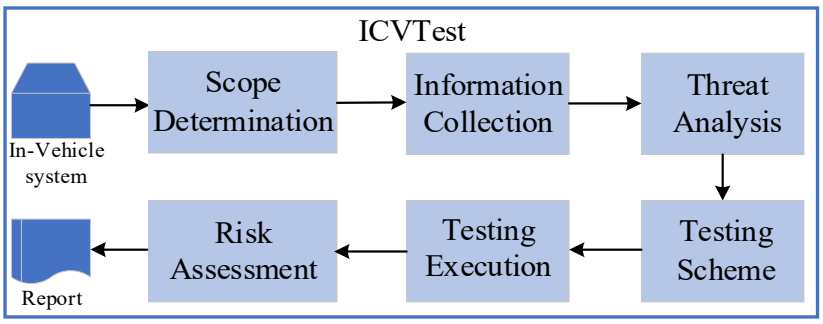

Figure 1. The penetration testing process in ICVTest.

1. Scope Determination

The architecture of ICVs is intricate. Without a precise test scope, testers cannot accurately define the boundaries of the test, which may impact normal vehicle functions and cloud platform services during testing. Clarifying the test scope resolves the issue of unclear test boundaries during the testing process, making it easier for testers to comprehend and establish these boundaries.

2. Information Collection

During the information collection stage, testers gather as much relevant information as possible about the TOT with proper authorization. This includes obtaining network addresses, chip details, communication protocols, interfaces, etc. Valuable public information can be sourced from official websites, promotional pages, and news reports. Testers can also disassemble the test target under authorized conditions to acquire crucial information such as hardware interfaces and chip models. The process of gathering information is iterative throughout penetration testing. As testing progresses further, more abundant and applicable data becomes available.

3. Threat Analysis

The objective of penetration testing is to comprehensively evaluate the TOT's cybersecurity from an attacker's perspective. The primary challenge lies in determining what to test, which can be addressed through threat analysis. Threat analysis serves as a reference for developing a robust penetration testing scheme.

4. Testing Scheme Development

Testers employ techniques such as attack trees to model potential threats faced by the TOT and identify its specific cybersecurity risks. The resulting penetration testing scheme meticulously covers these identified threats, ensuring that they are individually tested to prevent their evolution into actual vulnerabilities and mitigate serious cybersecurity risks.

5. Testing Execution

Once the testing scheme is determined, the tester selects the appropriate testing technology to conduct penetration testing based on the established scheme.

6. Risk Assessment

Risk assessment evaluates the severity of cybersecurity vulnerabilities by considering both the likelihood of successful attacks and their potential impact. The probability of a cybersecurity threat transforming into a vulnerability depends on factors such as attack sophistication, technical expertise of personnel involved, complexity of attack equipment, duration of attack, and economic costs incurred. The harm caused by an attack can manifest in various ways including vehicle function failure, personal injury, property loss, and privacy breaches.

The scope determination is employed to restrict the testing scope and prevent testers from conducting unauthorized tests. Information collection serves as a crucial activity for comprehending the TOT, laying the groundwork for threat analysis. Threat analysis constructs a threat model of the TOT to assist testers in elucidating its cyber security risks. Testing scheme development generates specific test cases based on the cyber security threats encountered by the TOT. Testers strictly adhere to the test scheme during execution. Risk assessment scrutinizes test results to aid designers in devising solutions. These six activities synergistically ensure standardization of VPT processes.

3.3. Penetration Testing Case Set in ICVTest

From the perspective of the in-vehicle bus, cybersecurity attacks on ICVs can be categorized into two groups: accessible bus cybersecurity attacks and inaccessible bus cybersecurity attacks. Accessible bus cybersecurity attacks have the potential to infiltrate the bus network and target other ECUs within it. On the other hand, inaccessible bus cybersecurity attacks focus on targets that are not connected to the bus network. Even if these targets are compromised, they cannot access or pose a threat to other ECUs within the same network. In scenarios where vehicle dismantling is not involved, potential entry points for accessible bus cybersecurity attacks include:

1. OBD-II

The second on-board diagnostics (OBD-II) interface serves as a crucial interface for ICVs to offer diagnostic services and presents an accessible approach for daily maintenance and ECU programming [32]. Various handheld scanning tools have been developed by automobile manufacturers and have been equipped with dedicated software that enables access to the in-vehicle bus network through the OBD-II interface for querying or programming ECUs within the in-vehicle network. However, if these tools fall under malicious control, unauthorized manipulation of ECU configuration becomes possible.

2. Sensors

ICVs are equipped with a diverse range of sensors to perceive both the surrounding environment and their own state, encompassing tire pressure monitoring sensors, global positioning systems, and ultrasonic sensors, among others. These sensors collaborate with numerous ECUs to form an in-vehicle network. Consequently, malevolent attackers can exploit vulnerabilities within these sensors as a means to infiltrate the in-vehicle network.

3. Infotainment system

ICVs offer infotainment functionalities such as navigation and multimedia playback. The ECU responsible for these functions exhibits a complex architecture, encompassing numerous physical and wireless communication interfaces, thereby expanding the potential attack surface. Once an adversary successfully compromises the infotainment system, they can exploit it to further infiltrate the in-vehicle network.

4. Wireless Communication

Wireless communication units, such as Wi-Fi, Bluetooth, radio frequency identification (RFID), tire pressure monitoring system (TPMS), and dedicated short range communication (DSRC), can be directly connected to ECUs like infotainment systems and telematics boxes (T-Box). Attackers may exploit these ECUs through wireless access points to launch further attacks on the in-vehicle bus network.

The objective of cybersecurity attacks on ICVs is to manipulate the vehicular behaviors by attacking the ECU. Cybersecurity attacks on ICVs typically consist of two stages. In the first stage, when disassembling the vehicle is not feasible, attackers can only gain access to externally exposed interfaces provided by the vehicle. This includes wireless communication interfaces such as Wi-Fi, Bluetooth, and cellular networks, as well as physical interfaces like OBD-II and universal serial bus (USB). If any of these in-vehicle communication modules possess a cybersecurity vulnerability, an attacker can exploit it to target the specific ECU through these mentioned interfaces. Subsequently, in the second stage, once an ECU has been compromised successfully, it serves as a gateway for attackers to infiltrate and control other ECUs within the in-vehicle network.

The attack path in ICVs encompasses various nodes, including cloud platforms, mobile phone applications (APPs), sensors, key ECUs, as well as physical and wireless communication channels facilitating node-to-node communication. The nodes involved in the attack path of ICVs are abstracted into 10 fields within ICVTest: hardware boards, ECU firmware, ECU operating systems, in-vehicle buses, network communications, cloud platforms, mobile devices, sensors, and private data. ICVs may face cybersecurity threats across 10 fields depicted in Figure 2. Within the vehicle, the on-board ECU consists of a hardware board with firmware or an operating system running on it. Different ECUs are interconnected through the vehicle bus to form an in-vehicle network. Outside the vehicle, sensors enable the automobile to perceive its surrounding environment and internal state. Sensor data is transmitted via the bus system to corresponding ECUs for analysis purposes. The radio communication module serves as a gateway between off-vehicle networks and the in-vehicle network. In addition to connecting vehicles with cloud service platforms, communication technology also facilitates connectivity with mobile terminal devices such as smartphones, introducing additional security risks for ICVs. Furthermore, the advent of ICVs brings forth significant concerns regarding data privacy while providing services.

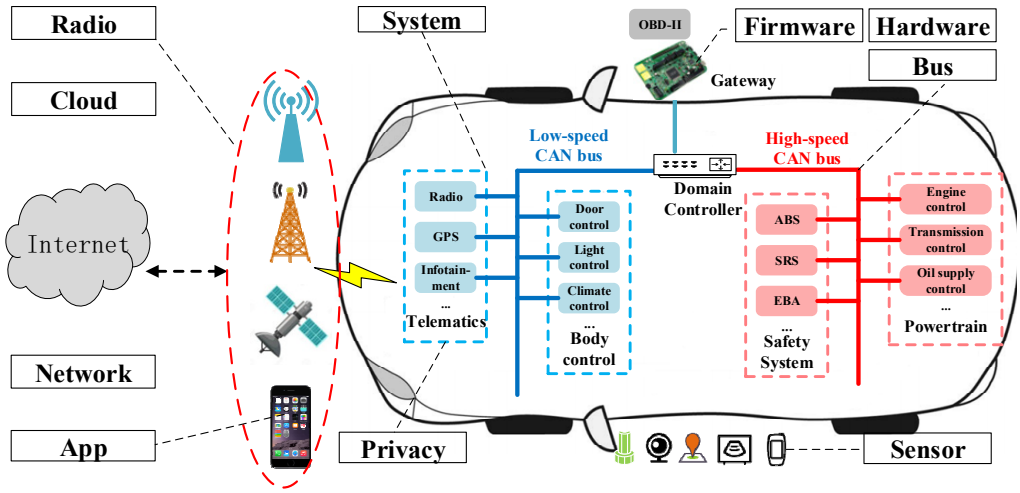

Figure 2. Ten Security Fields of ICVTest.

Based on the threats identified in Figure 2, ICVTest abstracts the vehicle cybersecurity test case set into 10 fields. The common cybersecurity threats that need to be verified by security tests are summarized in each test case set.

1. Hardware Security Test Case Set

Hardware security threats encompass printed circuit board (PCB) vulnerabilities, processor chip weaknesses, memory chip susceptibilities, hardware debugging interface risks, and on-board bus vulnerabilities. In the case of PCB in ECU, there is potential for information leakage regarding integrated circuit (IC) chip models, interface details, and bus protocol. The processor chip may inadvertently disclose side channel information such as electromagnetic signals, timing data, and power consumption patterns [33] thereby

facilitating attackers to launch timing attacks [34], power consumption attacks [35], and electromagnetic attacks [36]. Additionally, fault injection techniques [37] pose significant security risks to processor chips. Attacks like voltage fault injection [38], electromagnetic fault injection [39], and laser fault injection [40] can alter the operational logic of the processor. Data storage chips are also susceptible to data residue threats. For instance, flash memory can be read by a programmer. Furthermore, hardware debugging interfaces like joint test action group (JTAG), serial wire debug (SWD), and USB provide attackers with opportunities to access internal storage data of the system on chip (SoC). Attackers may also monitor buses such as serial peripheral interface (SPI) or inter-integrated circuit (I2C), leading to potential data leakage risks.

2. Firmware Security Test Case Set

Private information, such as passwords, keys, network resource addresses, usernames, and email addresses, may be stored in plain text within the firmware. Malicious attackers can decipher the operational logic of the TOT through reverse engineering techniques. Furthermore, these attackers can even assess whether the target firmware possesses cybersecurity vulnerabilities within real-world operating environments using dynamic analysis.

3. In-Vehicle Bus Security Test Case Set

ICVs employ various in-vehicle bus protocols, including CAN bus, FlexRay bus, local interconnection network (LIN) bus, media oriented system transport (MOST) bus, and in-vehicle ethernet bus. On one hand, certain in-vehicle bus protocols lack essential security mechanisms such as encryption, authentication, and integrity to meet the specific requirements of low-latency communication within vehicles. On the other hand, some application layer protocols for in-vehicle buses allow external individuals to access and control the vehicle's functionalities. Examples include unified diagnostic services (UDS) and scalable service-oriented middleware over IP (SOME/IP). The absence of adequate security measures renders these in-vehicle buses vulnerable to potential attackers who can gain control over them, resulting in severe consequences.

4. System Security Test Case Set

A secure ICV operating system necessitates stringent control over external entities' access to resources within the system. Ensuring the security of the operating system entails not only safeguarding data through mechanisms like authority access control, encryption, and integrity verification during design but also mitigating cybersecurity risks arising from design and implementation defects via a series of configurations.

5. Radio Security Test Case Set

ICVs are equipped with various wireless communication modules that operate on different radio frequencies to cater to diverse communication needs in scenarios such as Wi-Fi, Bluetooth, near field communication (NFC), RFID, cellular networks, and dedicated short-range communications (DSRC). However, the inherent openness of the wireless communication channel introduces vulnerabilities for ICVs. If the traffic transmitted over the radio channel is not encrypted, malicious attackers can intercept and manipulate user secrets by collecting and analyzing radio signals.

6. Network Security Test Case Set

In contrast to radio security, network security primarily focuses on ensuring the security of the TCP/IP protocol stack at the network level, while the former concentrates on physical layer and link layer security. Malicious attackers have the potential to intercept sensitive data transmitted over a network, and they can also manipulate legitimate data obtained through replay attacks in order to deceive recipients. Furthermore, attackers may even exploit communication between entities to carry out man-in-the-middle attacks. Once successfully hijacked, these attackers can eavesdrop and tamper with communication information exchanged among authorized users.

7. Web Security Test Case Set

The interaction between ICVs and the cloud platform may introduce cybersecurity risks to the ICVs. In the event of a compromise of the cloud platform by attackers, not only can private data such as user information be leaked, but also the ICVs can be invaded through remote wireless networks. Web applications, being crucial components of the cloud platform, face threats from both clients and servers. Client-side threats primarily encompass browser vulnerabilities, cross-site scripting (XSS) attacks, cross-site request forgery (CSRF) attacks, clickjacking exploits, and hypertext markup language (HTML) vulnerabilities. On the other hand, server-side threats mainly involve injection attacks, file upload vulnerabilities, authentication and session management weaknesses, access control issues, web framework susceptibilities, distributed denial-of-service (DDoS) attacks, and improper server configurations.

8. APP Security Test Case Set

In the context of remote vehicle control scenarios, applications can potentially serve as a gateway for attacks, posing significant cybersecurity risks to ICVs. The assets that require protection within these applications encompass client files, local storage data, application processes, runtime data, interactive interfaces, and network communication. Client files may inadvertently expose system logic or sensitive information or be subject to malicious tampering. Certain applications may store critical information such as bank cards, ID cards, contacts, and account passwords locally. Without robust security mechanisms in place, application data is highly susceptible to compromise. Upon launching the application, it requests resources from the system and initiates a process to execute its main logic. An attacker could exploit vulnerabilities by forcibly terminating or hijacking a process or injecting malicious data into it, thereby impeding normal program execution.

9. Sensor Security Test Case Set

ICVs heavily rely on sensors for achieving autonomous driving, thereby introducing a broader attack surface and potential cybersecurity risks to vehicles. Various types of sensors are embedded in ICVs, including cameras, Lidar, ultrasonic radar, millimeter-wave radar, GPS, etc. The GPS signal tends to weaken after long-distance transmission, making it susceptible to interference from simulated GPS signals that can cause location deception. Cameras can be rendered ineffective by arrays of powerful light sources. Similarly, millimeter-wave radar can be disrupted by signals with similar waveforms.

10. Privacy Security Test Case Set

In intelligent driving scenarios, vehicles need to periodically transmit their state information to enhance driving efficiency. In the absence of encryption, adversaries can exploit wireless monitoring technology to acquire the status of the target vehicle. Furthermore, services such as danger warning and personalized recommendation necessitate access to sensitive data including vehicle identity information, user habits, and web browsing records.

Test cases in ICVTest are not limited to specific modules and systems in ICVs, such as infotainment systems or automatic driving control systems. Instead, ICVTest abstracts the software and hardware components involved in in-vehicle systems into 10 fields. These test cases comprehensively cover all 10 fields of automotive software and hardware, providing sufficient guidance for testers to conduct detailed cybersecurity evaluation for ICVs. Testers are relieved from the need to customize a penetration testing scheme for each TOT. Instead, they can simply reuse relevant test cases based on the software and hardware architecture of the TOT. As a result of leveraging threat analysis results, testers only need to select the appropriate cybersecurity test cases to assess whether exploitable vulnerabilities exist within the TOT. ICVTest significantly standardizes penetration testing content, reduces its complexity, and enhances overall efficiency.

3.4. Cybersecurity Testing Platform for ICVTest

The implementation of a cybersecurity testing platform for ICVTest aims to facilitate penetration testers, particularly inexperienced novices, in effectively carrying out testing tasks. As depicted in Figure 3, the platform is divided into four components: ICVTest infrastructure, ICVTest tool and guide manual, ICVTest agent, and ICVTest web application.

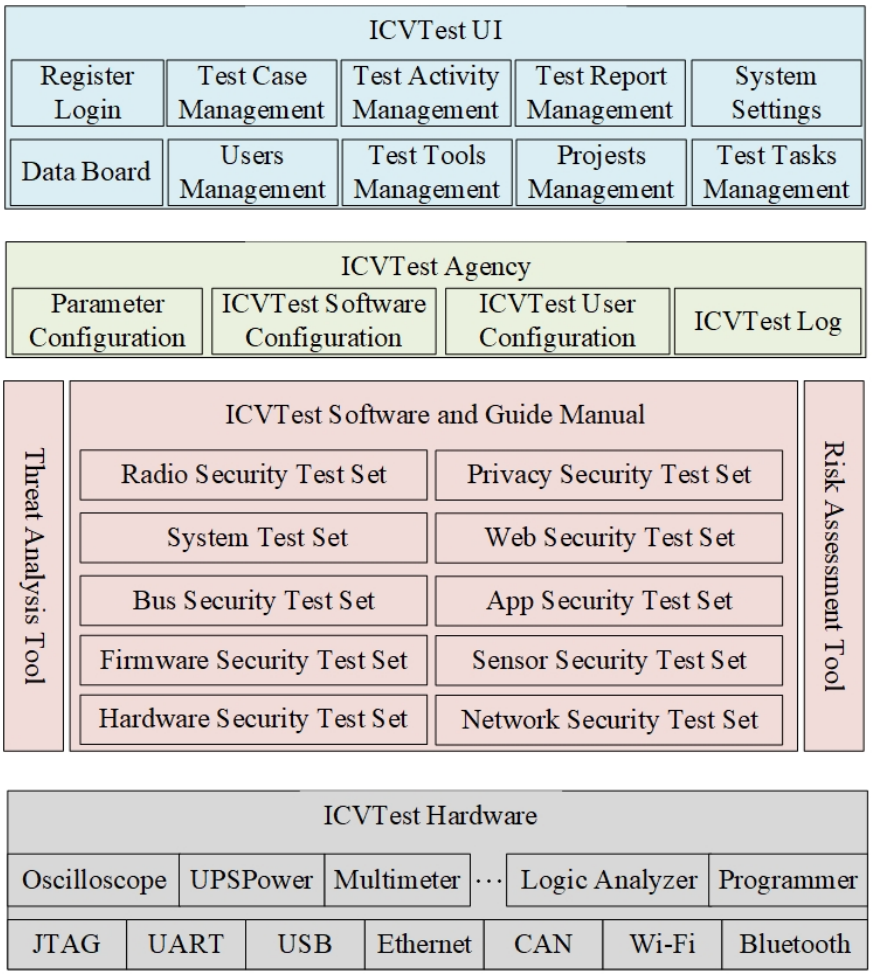

Figure 3. The cybersecurity testing platform for ICVTest.

1. ICVTest Infrastructure

ICVTest infrastructure cabinet is equipped with essential tools for hardware security testing including oscilloscopes, multimeters, logic analyzers, and programmers. It also facilitates communication through various interfaces such as JTAG, USB, in-vehicle ethernet, and CAN bus. Additionally, the cabinet ensures uninterrupted operation during unexpected power loss by utilizing UPS power supply. Furthermore, the integrated server hosts ICVTest tool, ICVTest agents, and ICVTest web applications.

2. ICVTest Tool and Guide Manuals

Due to the complexity of ICVs, there are numerous test cases that cannot all be automated. Therefore, ICVTest offers three distinct types of test case. ICVTest provides automated test scripts for test cases with well-defined criteria and high automation potential, ensuring a low false positive rate. For test cases with ambiguous judgment criteria, ICVTest integrates testing tools and allows testers to make manual judgments based on test data. Additionally, for test cases requiring extensive human participation the ICVTest integrated guide manuals assist testers in conducting tests and provide examples for making judgments.

3. ICVTest Agent

The ICVTest agent serves as a crucial link between the ICVTest web application, ICVTest infrastructure, and ICVTest tool, offering comprehensive support for driving both hardware and software to accomplish automated testing. Upon receiving instructions from the ICVTest web application, the testing agent invokes the local testing tool to execute tests while providing real-time reporting of processed data and results back to the ICVTest web application.

4. ICVTest Web Application

The ICVTest web application facilitates interaction among testers, test supervisors, and project supervisors by integrating features such as data board, user management, tool management, project management, task management, and test case management. The web application primarily focuses on collaborative multi-person test management to effectively handle personnel, equipment, software, projects, tasks, tests, and reports involved in the testing process. The ICVTest scalability is ensured through flexible configuration or online editing of test cases.

In the test scenario, testers log into the ICVTest web application with test terminals and select applicable test cases from the test case database based on the threat analysis results within the scope of authorization to form a penetration test plan. Once physically connected to the vehicle under test, the selected test case is executed. The ICVTest agent transmits the web-based test instructions to the ICVTest infrastructure and displays the test data on the ICVTest web application. Upon completion of the penetration test, an automated report is generated by the platform for review by the design team in order to address any cybersecurity vulnerabilities identified in relation to the vehicle.

4. Experiments

ICVTest is applied for VPT, utilizing the publicly obtained information without any additional knowledge about the specific vehicle, which is an electric ICV manufactured in China, 2021.

4.1. Scope Determination and Information Collection

Testers are authorized to conduct comprehensive penetration testing on the target vehicle without disassembling the vehicle's ECUs or interfering with the normal services of the cloud platform. The scope of penetration testing for the TOT is as follows:

- Testers can access and interact with the vehicle, but they are not permitted to dismantle it in order to obtain the ECU. Due to this limitation, hardware security testing in ICVTest cannot be fully implemented.
- Testers are not allowed to perform penetration tests on the cloud platform so as not to disrupt regular business operations.
- Testers possess physical keys and legal user accounts that enable them to log into the cloud platform. However, since they cannot physically damage or alter the vehicle, all its functions remain black-boxes for testers. With limited information available, testers can only control input parameters of vehicle functions and observe their responses.
- Despite these constraints, testers need to carry out a comprehensive penetration testing for the entire vehicle.

The process of information collection is iterative in nature. As the penetration testing progresses, testers can continuously obtain new information to establish a foundation for subsequent testing phases. Prior to conducting the penetration testing, the publicly available information regarding the vehicle includes:

- Infotainment: the infotainment system features a large touch screen and operates on the Android operating system, offering numerous applications and supporting an extensive range of infotainment experiences. It provides network functionalities such as Bluetooth, 4G, Wi-Fi, and hotspot connectivity. Additionally, it enables communication with cloud platforms and mobile devices.

- Mobile device remote vehicle control: users have the ability to remotely control certain vehicular functions through a dedicated app installed on their mobile devices. Following user identity verification, operations like door unlocking, lighting control, and air conditioning adjustment can be performed remotely.
- Bluetooth digital key: users have the option to activate the Bluetooth digital key functionality which allows their mobile devices to function as virtual keys. However, it should be noted that this feature has not been activated in our experimental vehicle.
- OTA: the target vehicle can remotely upgrade certain functions through OTA.
- When the user approaches the vehicle with the physical key, passive keyless entry allows for automatic unlocking. The engine will only start when the physical key is detected inside the vehicle.
- TPMS: TPMS equipped in the vehicle sends a warning message to the driver when there is a significant deviation from normal tire pressure.
- An ultrasonic sensor installed in the vehicle alerts the driver of obstacles during reversing maneuvers.
- With traffic sign recognition enabled by a high-definition camera, road traffic signs are recognized to assist drivers in making informed decisions while driving.
- Vehicle diagnosis is facilitated through an OBD-II diagnostic interface, allowing for fault detection and localization when connected to diagnostic equipment.

4.2. Threat Analysis

According to the threats depicted in Figure 2 and within the scope of penetration testing, attackers are limited to exploiting the exposed hardware interfaces, wireless interfaces, and sensors of the vehicle for launching attacks. The in-vehicle bus network can be compromised by attackers through the OBD-II interface. As a result of utilizing Wi-Fi, Bluetooth, and 4G network communication, attackers can infiltrate the external network of the vehicle. Malicious attacks targeting GPS, TPMS, ultrasonic sensors, and on-board cameras have potential consequences such as providing incorrect vehicular status or environmental information.

As illustrated in Figure 4 and Table 1, various in-vehicle systems including infotainment systems, cloud platforms, mobile devices, CAN bus networks, and sensors face direct cybersecurity threats from potential attackers. Adversaries can exploit multiple entry points such as Bluetooth, Wi-Fi, 4G networks, mobile applications (APPs), infotainment systems, OBD-II ports, TPMS, passive keyless entry systems (PKESs), GPS, cameras, and ultrasonic sensors.

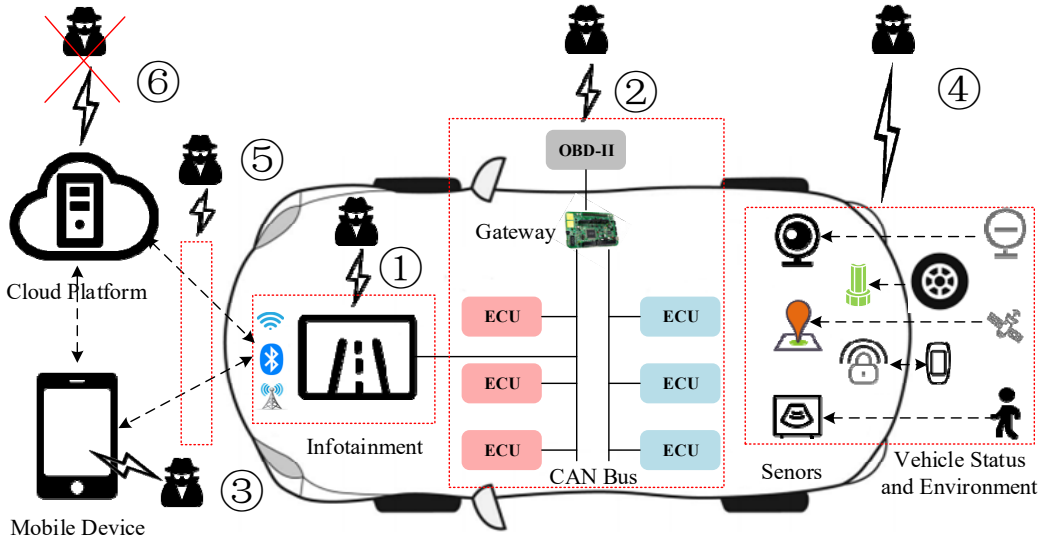

Figure 4. Threats of the target vehicle.

Table 1. Cybersecurity threats.

Threat	In-Vehicle System	Attack Entry	Threat Fields
①	Infotainment	Bluetooth	Radio Threat/ System Threat
		Wi-Fi	
		4G	
		USB	Hardware Threat
②	CAN Bus	OBD-II	In-vehicle Bus Threat
③	Mobile Device	APP	APP Threat
④	Sensors	PKES	Sensor Threat
		GPS	
		Camera	
		Ultrasonic Radar	
⑤	Network	Infotainment	Network Threat
		APP	
⑥	Cloud Platform	APP	Web Threat
		Infotainment	

4.3. Penetration Testing Scheme

The vehicle penetration testing scheme, as presented in Table 2, is based on the results of cybersecurity threat analysis and authorized testing scope to effectively address potential cybersecurity threats. The mobile device primarily serves for remote vehicle control and virtual Bluetooth key functionalities. However, since the Bluetooth digital key function of the target cannot be utilized normally, mobile device testing is considered as a form of remote vehicle control evaluation. Due to hardware limitations preventing access to firmware through disassembling the ECU, testers initially treat the vehicle as a black-box during penetration testing. Consequently, they rely on fuzzing technology to identify potential cybersecurity vulnerabilities by manipulating input parameters related to various vehicle functions and observing corresponding responses. Fuzz testing technology is predominantly employed in assessing communication channels carrying baseband data such as Bluetooth, Wi-Fi, network communication protocols etc. while vision-based sensors like in-vehicle cameras and ultrasonic sensors are excluded from the fuzz testing scope.

Table 2. Penetration testing scheme.

In-Vehicle System	Test Case	Test Case Set
Infotainment	Bluetooth Test	Radio Test Case Set
	Wi-Fi Test	
	4G Test	
	"Vehicle-Cloud" Network Test	Network Test Case Set
	In-vehicle Operating System Test	System Test Case Set
	USB Test	Hardware Test Case Set
CAN Bus	CAN Bus Test	In-vehicle Bus Test Case Set
Remote Vehicle Control	APP Test	APP Test Case Set
	Network Test	Network Test Case Set
Sensors	PKES Test	Sensor Test Case Set
	GPS Test	
	Camera Test	
	Ultrasonic Radar Test	

4.4. Penetration Testing Execution

4.4.1. Infotainment Test

1. Bluetooth Test

After scanning, only one Bluetooth device was detected in the infotainment system, and no unreported Bluetooth devices were found. The identified Bluetooth device utilizes LE Secure Connection mode. When other Bluetooth devices attempt to pair with the car's Bluetooth device, a 6-digit personal identification number (PIN) code is required for authentication. This verification process safeguards against potential security threats posed by malicious devices, however, when the simulated Bluetooth device connects to the in-vehicle Bluetooth system, it bypasses the pairing request verification mechanism, thereby introducing a cybersecurity vulnerability.

2. Wi-Fi Test

The Wi-Fi hotspot utilizes the robust WPA2-Personal protocol, ensuring adequate authentication and encryption strength. The infotainment has no Wi-Fi proxy function. Thereby, malicious attackers cannot conduct man-in-the-middle attacks or hijacking communication traffic between the infotainment system and cloud platform. Disabling Wi-Fi results in network disconnection between the infotainment system and cloud platform, however, the vehicle remote control function remains operational. A conspicuous user prompt is displayed upon enabling Wi-Fi on the infotainment system. Conversely, there is an absence of a clear user prompt when utilizing Wi-Fi to transmit data, potentially leading to the unauthorized transmission of private information.

3. 4G Test

By conducting scanning, the tester can acquire information such as frequency and frequency band of nearby base stations. Utilizing USRP B210 and the obtained base station information, an LTE pseudo base station and LTE interference equipment is constructed. The interference device transmits a significant amount of meaningless data on the communication frequency of the original base station to disrupt communication. Upon detecting deteriorated communication quality, the target vehicle initiates cell reselection mechanisms to ensure stable data services by selecting alternative cells. The vehicle prioritizes connection with carefully configured pseudo-base stations. Despite mutual authentication being required between terminal devices and base stations according to the LTE protocol, there is still necessary data exchange prior to authentication taking place. The terminal device's sent base station attachment request message includes international mobile subscriber identity (IMSI) which facilitates tracking of the vehicle's trajectory. In our experiment, unauthorized road usage by other terminal devices near the target vehicle causes interference, hence multiple IMSIs will be extracted by testers when it is not possible to remove SIM from the vehicle resulting in inability to determine corresponding relationship between vehicle and IMSI.

4. "Vehicle-Cloud" Network Test

Preliminary testing reveals that the communication channels for infotainment and remote vehicle control are independent of each other. Even when 4G communication is disabled, remote vehicle control remains functional, indicating that it operates on a separate network. The infotainment system does not offer Wi-Fi proxy functionality, making it impossible for attackers to hijack its communication traffic based on this feature alone. Testers can monitor the infotainment's communication traffic by utilizing test equipment as a gateway. Analysis of the captured messages confirms that while the infotainment authenticates the cloud platform, there is no reciprocal authentication from the cloud platform to the infotainment. All transmission messages are encrypted within the communication channel and protected by an integrity check mechanism. Although channel encryption cannot completely prevent man-in-the-middle attacks, external certificates cannot be imported or installed in the infotainment system, rendering attackers unable to execute such attacks against its secure communication protocol or crack channel encryption. Consequently,

it can be concluded that cybersecurity protection measures implemented in the network communication of this particular infotainment system possess sufficient strength.

5. In-vehicle Operating System Test

Network scanning reveals that the vehicle exposes DoIP-data service to external connections, which is utilized for remote vehicle diagnostics. If an attacker infiltrates the internal network of the vehicle, they can exploit the diagnostic service to access ECU internal data such as voltage and mileage and even manipulate ECU internal data like fault codes. Regarding account permissions, user login authentication in the vehicle relies on a QR code-based identification method and a mobile phone verification code. The system settings do not provide an option to enable USB debugging as observed during testing. With respect to application permission testing, system applications clearly prompt users with explicit indications when making calls, sending text messages, recording audio or video content, and accessing location information. However, users are not explicitly notified when utilizing location services.

6. USB Test

The on-board USB interface does not restrict the connectivity of external USB devices, thereby enabling potential security risks for users as attackers can exploit BadUSB to emulate an external keyboard and mouse, gaining control over the vehicular infotainment system. This vulnerability could be exploited to manipulate in-vehicle Bluetooth functionality, such as initiating unauthorized phone calls.

4.4.2. CAN Bus Test

1. CAN Frame Reverse Test

CAN frames are capable of controlling vehicle functions and status through specific segments in the data field, exhibiting certain characteristic features. As the state of the vehicle changes so do these characteristics, which can be analyzed to obtain a communication matrix. Through testing, we have identified the control segment in the data field of CAN frames (as shown in Table 3) which attackers can exploit to manipulate vehicular behaviors.

Table 3. The control segment in CAN frame data field.

CAN ID	CAN Segment	Function
0 × 375	[28, 29]	left turn signal
0 × 37d	[51, 51]	high beam
0 × 3a5	[20, 20]	air conditioner
0 × 1a3	[19, 21]	player volume
0 × 379	[21, 22]	right window

2. Replay Attack Test

In the CAN frames replay test, the testers replicated the CAN frames generated by various vehicle actions, including activating the left turn signal, engaging high beam headlights, adjusting air conditioner settings, modifying multimedia volume levels, and opening windows. As depicted in Figure 5, when these replicated CAN frames were injected into the in-vehicle CAN bus system, corresponding expected vehicle behaviors were observed.

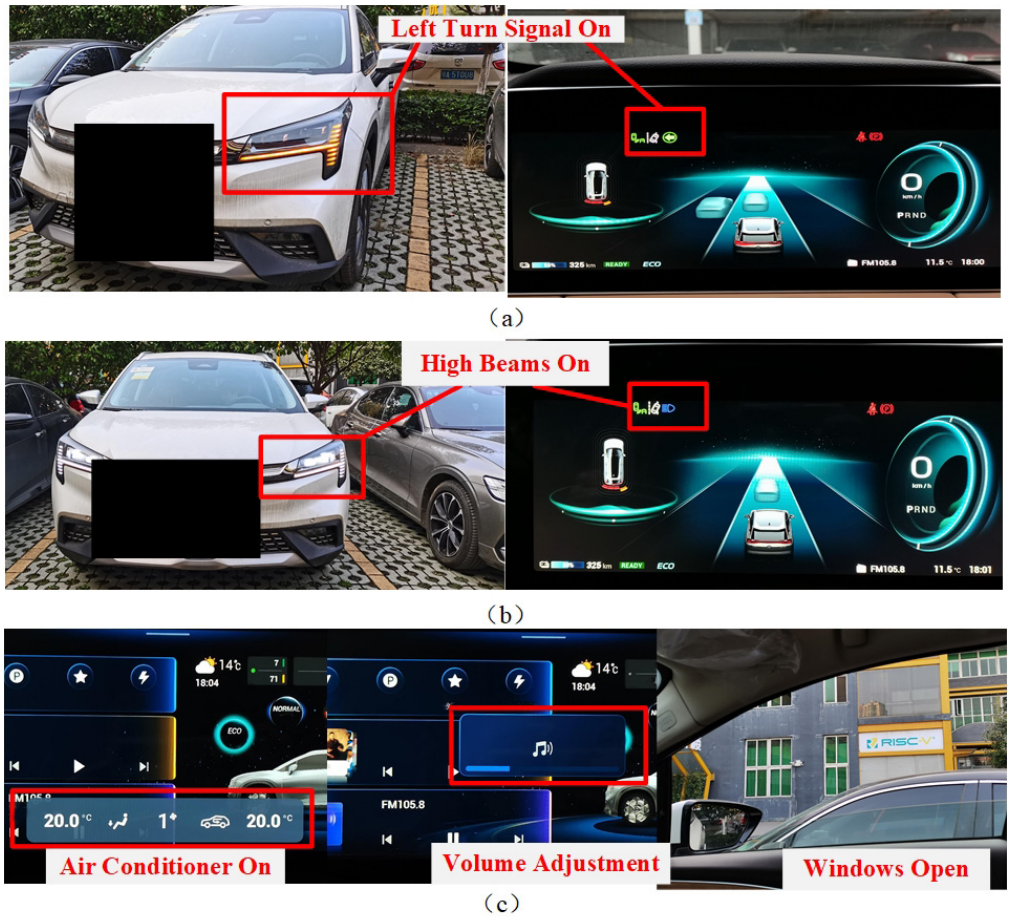

Figure 5. Replay attack scenario: (**a**) left turn signal on; (**b**) high beams on; (**c**) air conditioner on, volume adjustment, and windows open.

3. Drop-Off Attack Test

The CAN bus protocol employs a priority-based arbitration mechanism, whereby the transmission of high-priority messages can lead to the blocking of low-priority messages resulting in denial of service for ECUs. According to the CAN Frame Reverse Test, the tester constructed CAN frames for a drop-off attack as presented in Table 4. The testers continuously transmitted the constructed data frames to the vehicle's CAN bus at a specific rate, rendering both the high beam and left turn signal non-functional as shown in Figure 6. By injecting CAN frames for an air-conditioning drop-off attack into the CAN bus, the testers observed a significant weakening of engine sound when running the air-conditioning system. Simultaneously, in-vehicle air conditioning ceased cooling, although its status displayed on the infotainment system screen remained unchanged. While volume adjustment was still possible during offline attacks, it could not be accurately reflected on the display. Additionally, window control commands were ineffective under offline attacks.

4. Fuzzing Test

A fuzzing test is an effective approach for identifying potential unknown cybersecurity vulnerabilities. As a result of collecting CAN frames and subjecting them to mutation before sending them to the in-vehicle CAN bus, abnormal behavior of the vehicle can be induced during unmanned operation. These anomalous CAN frames may be exploited by malicious attackers. Table 5 presents the specific abnormal behaviors exhibited by vehicles in response to corresponding CAN frames.

Table 4. The CAN frames for Drop-Off attacks.

CAN ID	CAN Data	Function
0 × 375	[‘0 × 80’, ‘0 × 02’, ‘0 × 10’, ‘0 × 00’, ‘0 × 01’, ‘0 × 02’, ‘0 × 02’, ‘0 × 68’]	left turn signal
0 × 37d	[‘0 × 00’, ‘0 × 00’, ‘0 × 00’, ‘0 × 00’, ‘0 × 00’, ‘0 × 01’, ‘0 × a6’, ‘0 × 87’]	high beam
0 × 3a5	[‘0 × 00’, ‘0 × 00’, ‘0 × 00’, ‘0 × 00’, ‘0 × 00’, ‘0 × 00’, ‘0 × 00’, ‘0 × 00’]	air conditioner
0 × 1a3	[‘0 × 1a’, ‘0 × ed’, ‘0 × 40’, ‘0 × d9’, ‘0 × 7b’, ‘0 × 00’, ‘0 × 06’, ‘0 × 72’]	player volume
0 × 379	[‘0 × 59’, ‘0 × d0’, ‘0 × 00’, ‘0 × 11’, ‘0 × 45’, ‘0 × 68’, ‘0 × 95’, ‘0 × 14’]	right window

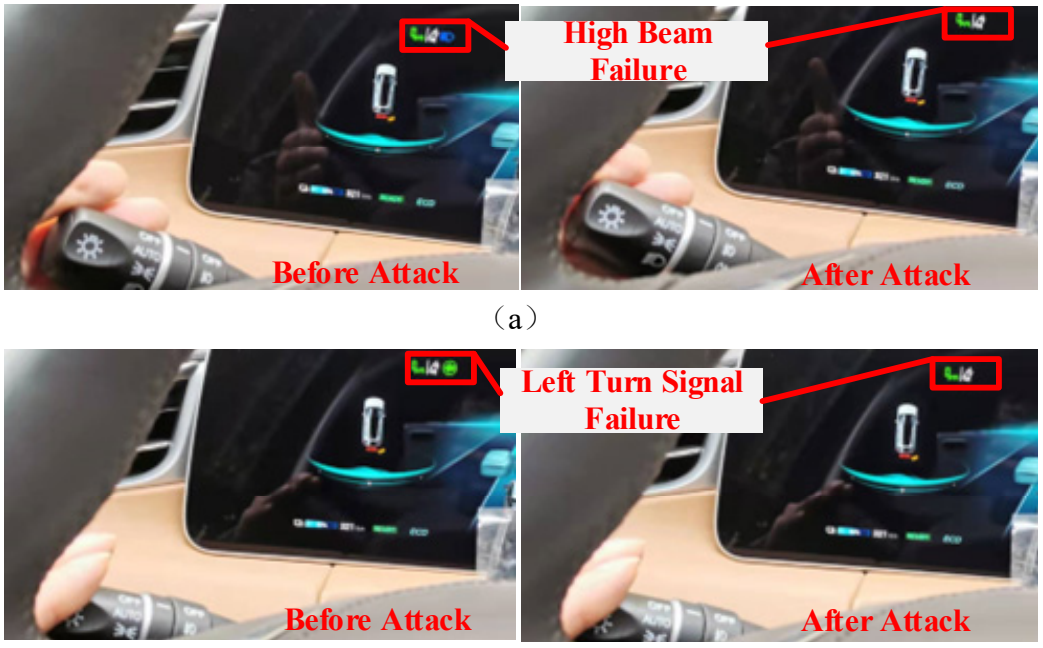

Figure 6. Drop-off attack scenario: (**a**) high beams failure; (**b**) left turn signal failure.

Table 5. The CAN frames for Fuzzing attacks.

CAN ID	CAN Data	Function
0 × 375	[‘0 × 41’, ‘0 × 02’, ‘0 × 10’, ‘0 × 1d’, ‘0 × 00’, ‘0 × 02’, ‘0 × 12’, ‘0 × 3a’]	left turn signal
0 × 375	[‘0 × aa’, ‘0 × 41’, ‘0 × 00’, ‘0 × 24’, ‘0 × 0c’, ‘0 × 19’, ‘0 × a3’, ‘0 × b5’]	left turn signal
0 × 42c	[‘0 × b1’, ‘0 × 0a’, ‘0 × ba’, ‘0 × c6’, ‘0 × de’, ‘0 × f4’, ‘0 × 03’, ‘0 × bc’]	wiper
0 × 37b	[‘0 × 29’, ‘0 × ee’, ‘0 × a3’, ‘0 × 9b’, ‘0 × 04’, ‘0 × 1a’, ‘0 × 7e’, ‘0 × 72’]	high beams
0 × 283	[‘0 × 3a’, ‘0 × dc’, ‘0 × 54’, ‘0 × 9b’, ‘0 × ac’, ‘0 × c8’, ‘0 × 24’, ‘0 × 1a’]	seat belt warning light
0 × 44c	[‘0 × b9’, ‘0 × 07’, ‘0 × ec’, ‘0 × 5a’, ‘0 × 30’, ‘0 × 80’, ‘0 × 7c’, ‘0 × 59’]	sunroof

5. ECU Access Test

The diagnostic CAN ID of a specific ECU can be determined by analyzing the diagnostic message. For instance, the hybrid power ECU of the vehicle has a diagnostic CAN ID of $0 \times 79a$. An attacker could exploit the 0×22 read data diagnosis service to retrieve internal storage data from this ECU. Figure 7 illustrates selected data from the hybrid power ECU during closed park driving conditions. Notably, as the electric pedal depth increases, there is a corresponding rise in motor load rate and vehicle power output, accompanied by significant changes in speed and power battery voltage.

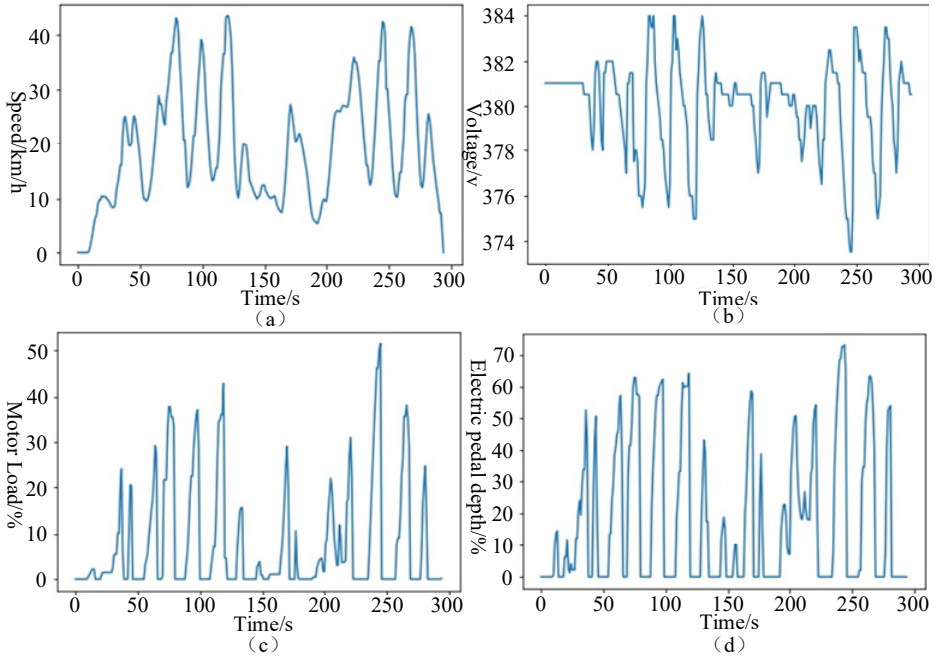

Figure 7. Power ECU access: (**a**) speed; (**b**) voltage; (**c**) motor load; (**d**) electric pedal depth.

4.4.3. Remote Vehicle Control Test

1. APP Test

When the APP resets the PIN code, it securely transmits the new PIN code along with the verification code to the cloud service platform. The cloud service platform receives both codes simultaneously and performs a successful verification of the received verification code before resetting the user's PIN code. In case of an unsuccessful verification, the request to reset the PIN code is rejected. The conducted tests demonstrate that no security vulnerability related to bypassing verification codes exists in the APP.

2. Authentication Test

The APP initiates a vehicle control request in the remote vehicle control scenario. In response to the request, the cloud server transmits remote vehicle control instructions to the vehicle. It is necessary to assess whether the communication between the APP and the cloud server possesses adequate security measures to prevent unauthorized forging of a malicious vehicle control request. Packet capturing tools like Fiddler can be utilized for intercepting communication packets exchanged between these two entities. The certificate contained within packets serves as crucial identity authentication data, enabling us to determine whether both entities have successfully authenticated each other's identities. By analyzing the communication packets, the cloud service platform sends a certificate to the APP as proof of its identity. The APP then verifies the identity of the cloud service platform and initiates a session key negotiation for encrypting communication data. However, it is noteworthy that during the handshake phase, no certificate is requested from the APP to verify its identity.

3. Encryption Test

The captured communication packets between apps and cloud servers can be analyzed using tools such as Wireshark. The packets reveal that the communication between the APP and the cloud service platform utilizes the TLS1.2 protocol, ensuring secure transmission of data. While the communication channel encrypts the data to maintain its confidentiality, it is important to note that complete security of transmitted data cannot be guaranteed solely through channel encryption. Potential attackers may exploit man-in-the-middle attacks to decrypt transmitted data and gain access to communication content. Further evaluation is required to determine if data payload undergoes encryption prior to being encapsulated into the communication channel. In our test scenario, we successfully decrypted the transmitted data payload that was encrypted in the communication channel, which contains multiple plaintext fields. However, crucial data payload has been encrypted before entering and being encapsulated within the channel, thereby enhancing security protection strength.

4. Integrity Test

Following the transmission of the request message to the cloud service platform, a response is generated by the platform, providing a verification outcome for the PIN code. If any unauthorized modification occurs in the encrypted PIN code within the request message, and this tampered request message is forwarded to the cloud service platform, an error in signature validation will be returned by the platform. Consequently, it can be inferred that the data integrity check was not successfully passed by this manipulated request message. While ensuring APP data integrity through its examination there exists a lack of integrity verification on cloud server data from APP's end. This vulnerability enables potential attackers to manipulate response data from the cloud service platform.

5. Replay Test

Following the testing phase, it has been determined that in order to successfully execute the remote control command of a vehicle, the tester needs to replay both the PIN code verification request message and the control command request message. All vehicle remote control commands such as air-conditioning activation/deactivation, seat heating adjustment, lighting toggling, and door opening/closing operations are susceptible to potential replay attacks. Subsequent testing revealed a time window of 30 s within which replay attacks can be executed successfully. If the time interval between consecutive replay control commands exceeds this 30 s threshold, the replay attack will fail.

6. PIN code Test

The testers efficiently generated 50 PIN code verification requests within a limited timeframe, all of which received accurate responses from the cloud service platform. Notably, the cloud service platform does not impose any restrictions on the number of attempts for entering the PIN code, thereby facilitating potential brute force attacks.

4.4.4. Sensor Test

1. PKES Test

PKES are extensively utilized in ICVs due to their convenience. However, the utilization of both low-frequency and very high-frequency channels for communication exposes PKES to potential relay attacks. These attacks involve a relay device that extends the communication range between the key and the vehicle, enabling an attacker to gain unauthorized access without detection by the owner. In our experiment, we positioned a low-frequency receiving antenna on the vehicular door handle as part of the vehicle-side relay device. Simultaneously, another tester carried a key-end relay device equipped with a high-power, low-frequency transmitting coil while following the vehicle owner. As depicted in Figure 8a,b, when appropriately positioned, the low-frequency receiving antenna successfully captures broadcast beacons from the vehicle and relays communications between the key and vehicle, resulting in successful door opening by testers. Furthermore,

as shown in Figure 8c, placing the low-frequency receiving antenna inside the vehicle simulates scenarios where keys are located within it, allowing the testers to start the vehicle and drive off with ease.

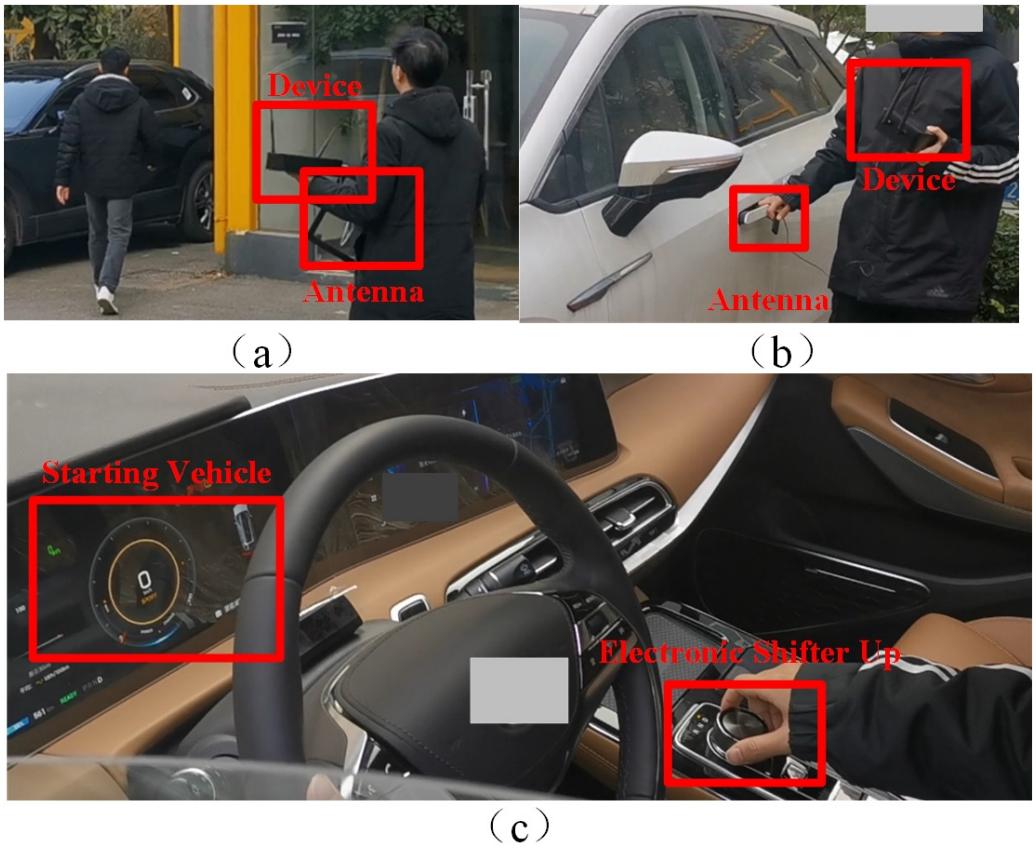

Figure 8. PKES relay test scenario: (**a**) tester follows vehicle owner; (**b**) tester is close to the vehicle; (**c**) tester opens the door and starts the vehicle.

2. GPS Test

When designed, GPS did not take into account the complexities of electromagnetic environment interference and cybersecurity attacks. It lacks encryption, integrity verification, and identity authentication mechanisms, making it convenient for attackers to forge GPS signals. In a GPS spoofing attack, the attacker can manipulate location-specific navigation messages based on ephemeris data. Due to the absence of an identity authentication mechanism, the attacker can generate stronger GPS signals to deceive receivers.

The GPS test primarily assesses whether the vehicle has implemented security measures to mitigate the risk of spoofing by counterfeit GPS signals. In the experiment, the attacker used HackRF to generate false GPS signals, and sent them with strong power to observe whether the target vehicle could be tricked into the set position. As depicted in Figure 9a, prior to the attack, the vehicle was situated in a park near our laboratory with relatively strong satellite signal strength. However, during the attack process shown in Figure 9b, there is a noticeable decline in satellite signal strength due to significant interference caused by GPS spoofing. Consequently, as illustrated in Figure 9c, although it is shown that the vehicle is located in Taipei City, the vehicle's surroundings still indicate its presence within our laboratory's park area. This successful spoofing attack effectively deceived the vehicle's GPS system.

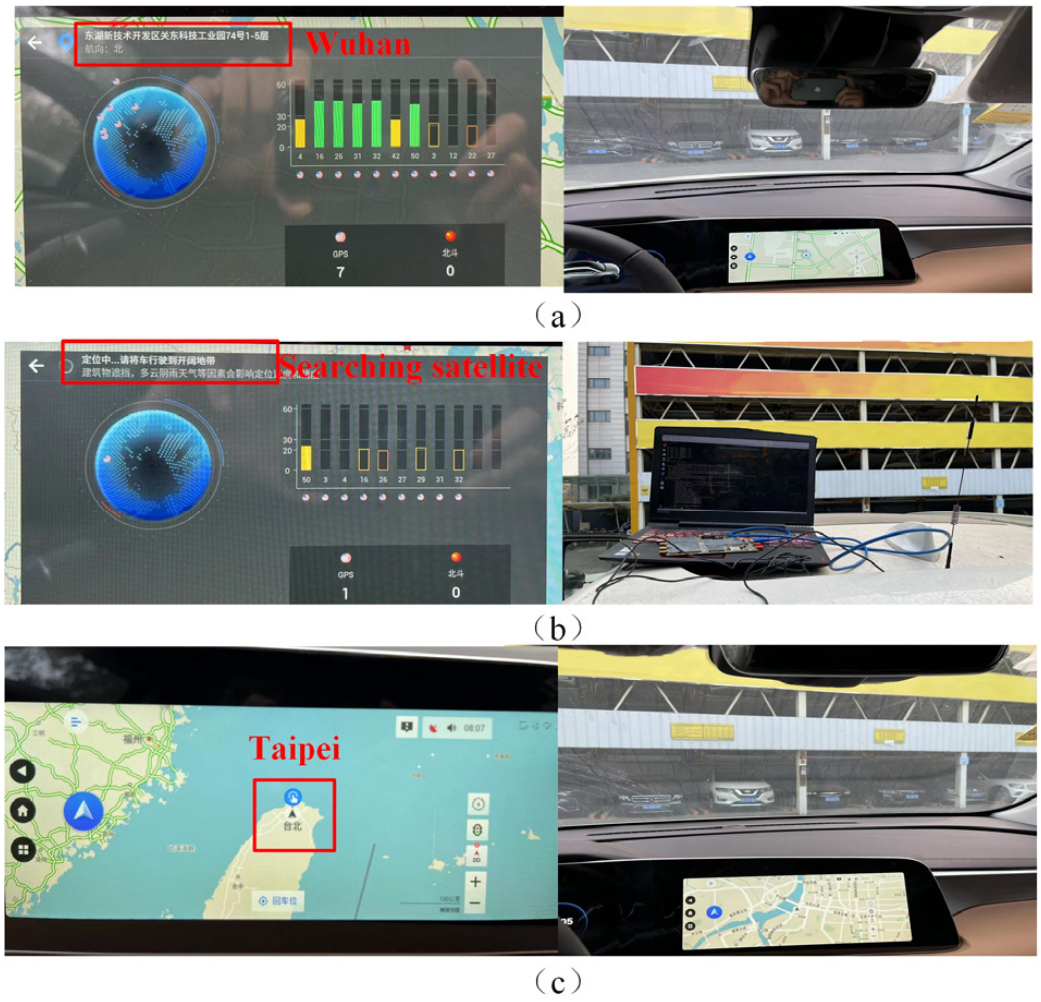

Figure 9. GPS spoofing attack scenario: (**a**) no attack; (**b**) jamming attack; (**c**) spoofing attack.

3. Camera Test

Attacks based on adversarial examples have emerged as a potent means to compromise the integrity of deep neural networks (DNNs). This technique introduces carefully crafted perturbations into images, which can disrupt the functioning of ICV image classifiers and lead to erroneous decisions in traffic sign recognition systems. As depicted in Figure 10a, under normal circumstances the vehicle accurately identifies the traffic signal displayed on the signboard. Conversely, in Figure 10b, during non-targeted attacks, the recognized traffic signal deviates from its actual representation according to the traffic sign recognition system. Furthermore, Figure 10c illustrates an instance of a hidden-target attack scenario where the system fails to identify any traffic signals within its environment successfully. Lastly, Figure 10d demonstrates how target attacks alter vehicle's traffic sign recognition results as they vary with respect to distance from the signage.

Figure 10. Camera attack based on adversarial examples: (**a**) no attack; (**b**) no target attack; (**c**) target hide attack; (**d**) specific target attack.

4. Ultrasonic Radar Test

Ultrasonic radar is extensively employed in the automotive industry due to its cost-effectiveness for measuring the distance between vehicles and obstacles. Adversaries can exploit an ultrasonic signal generator to produce ultrasonic waves of varying frequencies and phases, thereby disrupting the target ultrasonic radar's ranging capabilities. In the ultrasonic test, the tester employs an ultrasonic signal generator to produce ultrasonic signals of identical frequency as the in-vehicle ultrasonic sensor in order to verify the accurate measurement of obstacle distance by the target vehicle under interference conditions. The ultrasonic sensor test equipment is developed based on STM32 platform and CMSIS firmware library, ensuring excellent code portability. Four types of ultrasonic sensors (HC-SR04, US-015, US-100, and SR04T) are employed for cross-validation of experimental results. During the experiment, a 3.3–5 V level conversion module is utilized for voltage conversion between the STM32 platform and the ultrasonic sensor.

As depicted in Figure 11b, when there are no pedestrians behind the vehicle, no warning sign is displayed in the reversing image. However, if a pedestrian approaches from the rear, as illustrated in Figure 11c, a warning sign appears on the reversing image along with simultaneous emission of a high-frequency alarm sound to alert the driver. Furthermore, as shown in Figure 11d, even without any pedestrians present behind the vehicle, an obstacle warning sign accompanied by a high-frequency alarm sound is displayed on the reversing image. The ultrasonic attack device successfully simulates obstacles at the rear of the vehicle, leading to the emission of high-frequency alarm sounds and the display of warning signs.

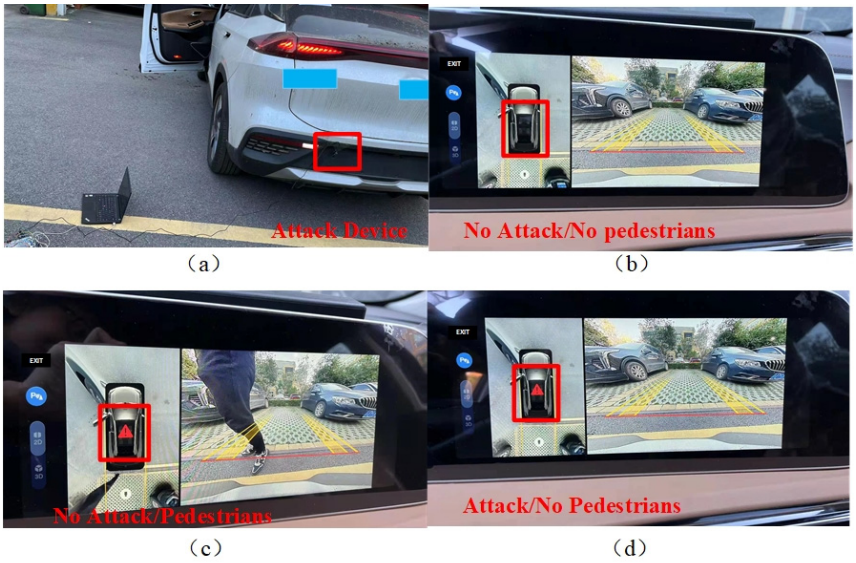

Figure 11. Ultrasonic radar attack scenario: (**a**) attack device; (**b**) no attack and no pedestrians; (**c**) no attack and pedestrians; (**d**) attack and no pedestrians.

4.5. Cybersecurity Risk Assessment

1. Remote diagnosis service information leakage and ECU control vulnerability

The remote diagnosis service exposes information leakage vulnerability and ECU control vulnerability. Attackers can exploit the UDS service whose ID is 22 to remote obtain some data in ECU, such as VIN code, hardware version, and firmware version. With the help of UDS service whose ID is 11, malicious attackers can manipulate the electronic control unit (ECU) into resetting. However, successful exploitation of cybersecurity vulnerability requires prior compromise of the vehicle's internal network. Considering the high strength of Wi-Fi passwords, attempting brute force attacks to guess passwords for invading the vehicle network becomes economically and temporally expensive. Consequently, due to the low probability of converting this cybersecurity threat into an actual attack, its associated risk is correspondingly low.

2. In-vehicle USB unauthorized access vulnerability

The infotainment system of vehicles lacks proper restrictions on accessing USB peripherals, thereby exposing a serious unauthorized access vulnerability. Attackers can employ BadUSB devices with malicious attack script that emulate unauthorized keyboards and mice to take control over various functions within the vehicle, including making phone calls.

3. ADB information leakage vulnerability

By enabling ADB debugging in engineering mode, attackers can acquire shell privileges over the infotainment system. Although the file system remains in read-only mode, attackers can still obtain configuration files, script files, shared library files, as well as various APK files through vulnerability mining techniques. The aforementioned files offer vital information for attackers to methodically analyze the vehicle and subsequently devise sophisticated attack strategies.

4. Unauthorized APP installation vulnerability

The vehicle under test was permitted to install the APP without authorization. Once ADB debugging is enabled, attackers have two options: pulling files from the infotainment system, and uploading APK files to specific directories within the file system. By utilizing specific command-line tools effectively, attackers can install malicious applications onto targeted systems leading to significant cybersecurity risks.

5. PIN brute-force cracking vulnerability

The vehicle does not impose a limit on the number of PIN code entries for remote vehicle control. Although the cloud service platform can only process PIN code verification requests at a limited speed within a specific time window, an attacker can still generate numerous real-time PIN code verification requests to perform brute force attacks and crack the PIN code. Once the PIN code is compromised, the attacker gains access to the remote vehicle control password, resulting in significant harm. However, this attack requires logging into the user's account with SMS assistance. It is challenging for an attacker to traverse all SMS verification codes within 5 min, thereby presenting a moderate cybersecurity risk. Additionally, remote vehicle control commands can be replayed within a 30 s timeframe.

6. Insecure in-vehicle CAN bus communication vulnerability

The in-vehicle CAN bus lacks encryption, authentication, and integrity verification mechanisms to safeguard communication data. Attackers can monitor bus data, reverse engineer communication matrices, construct unauthorized vehicle control commands, and launch offline attacks and diagnostic attacks against vehicles' systems through physical access to the CAN bus network. Nevertheless, due to requiring physical proximity for exploitation purposes, the direct exploitation of this vulnerability poses significant challenges. The vulnerability is usually used as an attack point in an attack chain initiated from outside to achieve vehicle control.

7. GPS spoofing vulnerability

An adversary has the capability to fabricate GPS signals with stronger signal strength in order to deceive the GPS system of autonomous driving vehicles that heavily rely on positioning information, posing substantial cybersecurity hazards. GPS attacks based on Software-Defined Radio tool are comparatively straightforward and have extensive coverage areas where they can be executed effectively. Consequently, the cybersecurity risks associated with such attacks are considerably high.

8. Ultrasonic Radar jamming vulnerability

When there is no obstacle behind the vehicle, the attacker successfully fabricates a false obstacle to deceive the sensor of the vehicle, leading it to mistakenly perceive an obstacle behind it. Although this attack is relatively simple to execute, it typically targets the parking assistance function and poses a minimal cybersecurity risk.

9. Traffic sign recognition system spoofing vulnerability

Under adversarial sample attacks, the vehicle fails to accurately recognize traffic signs on the road or even detect their presence. This attack significantly impacts functions such as automatic cruise control and advanced assisted driving in high-speed scenarios, posing substantial risks to personal safety and property. While constructing sophisticated adversarial samples is highly challenging, once these samples targeting specific vehicles are leaked, any attacker can launch an attack without physical access to the vehicle. Therefore, this vulnerability presents a high cybersecurity risk.

10. PKE relay vulnerability

Attackers can exploit relay attack devices to gain unauthorized access to vehicle doors and start the engine, facilitating vehicle theft. Despite the challenging development of such attack equipment, if assailants manage to acquire it through illicit means, they can effectively execute attacks and pilfer target vehicles without requiring specialized knowledge. This poses significant risks in terms of substantial property losses and heightened cybersecurity vulnerabilities.

5. Discussions

5.1. Scalability

ICVTest follows a principle of inside-to-outside, bottom-to-top, and hardware-to-software when abstracting the software and hardware involved in the attack path of intelligent connected vehicles into 10 fields: hardware board, ECU firmware, ECU operating system, in-vehicle bus, sensor, network communication, cloud platform, mobile device, and privacy data. The framework decouples penetration test cases from automotive subsystems while tightly binding them to automotive functions and associated software and hardware components. Therefore, no matter the size and complexities of vehicles, any vehicle equipped with corresponding functions and software/hardware components will require penetration tests against each field to assess the relevant cybersecurity threats. Compared to some existing penetration test frameworks which focus on specific automotive subsystems like infotainment systems, our proposed hierarchical penetration testing framework provides standardized test cases for various vehicles with different sizes and complexities. Moreover, our framework could expand with additional security assessment fields if it is required in the future.

5.2. Applicability

The ISO/SAE 21434 "Road vehicles—Cybersecurity engineering" has emerged as the prevailing standard for the automotive cybersecurity activities. The standard mandates that penetration testing be conducted on vehicles during the development and verification phases to evaluate potential cybersecurity risks. The ICVTest establishes a standardized process for penetration testing, providing 10 fields of testing instructions. Additionally, we have implemented a cybersecurity testing platform to facilitate efficient penetration testing by testers guided by ICVTest. The guidance provided by ICVTest enables testers to systematically conduct cybersecurity assessments of vehicles at appropriate stages in the development process, ensuring compliance with VTA.

5.3. Evolution

New cybersecurity threats are constantly emerging as automotive attack technologies evolve. ICVTest abstracts the vehicle cyber security test points into 10 fields. The test case is bound to JTAG interface, Wi-Fi module and other highly granular vehicle functions and software and hardware modules, which are highly decoupled from the modules to be tested. Testers do not need to customize the test plan for each vehicle module to be tested, but only need to reuse the corresponding test cases in the framework based on the module software and hardware architecture. If a new security threat appears, we will develop a specific test case and add the test case to the ICVTest test case database. Furthermore, the test method will also be integrated into the cybersecurity test platform in the form of test tools or guide manuals. The ongoing updating of the test case database guarantees that ICVTest is capable of supporting the evaluation of emerging cybersecurity threats.

5.4. Privacy

ICVTest focuses on the protection of vehicle privacy data. Firstly, the tester will exclusively conduct penetration testing on authorized systems. Unauthorized systems will not undergo any test. Secondly, from the perspective of black-box testing, testers are provided with limited information and lack proper access to the vehicle system, ensuring the security of local application data. The access control mechanism itself is also subject to evaluation. Thirdly, in accordance with ICVTest data management regulations, testers are required to strictly safeguard vehicle data and test reports obtained during testing. Additionally, the main purpose of ICVTest is to identify cybersecurity vulnerabilities and threats in 10 fields of ICVs, including privacy. During the penetration testing process, some vehicle data may be obtained, however, beyond meeting test requirements, no additional review of the data content is performed by the tester.

5.5. Performance

The cybersecurity testing platform for ICVTest was implemented on the hardware platform with the following specifications: Processor-Xeon 2300 series with 8 cores, Memory-64GB, Hard disk-1TB. CentOS7 is used as the operating system. The performance of the platform is presented in Table 6.

Table 6. The performance of cybersecurity testing platform for ICVTest.

Metric	Definition	Parameter
Response Time	The number of services that can be handled by the platform per unit time	<500 ms
Query to View Time	The total time consumed by the Query service	<50 ms
Query per Second	The number of queries the platform can process per unit time	>600
Transaction per Second	The number of integrated services that can be handled by the platform per unit time	>400
Concurrent Users	The number of users who simultaneously log in to the platform and perform business operations	≥30
Concurrent Agents	The number of agents that log in and register at the same time for business processing	≥30
Failure Ratio	Failure Ratio = (failed service)/(total service) * 100%	<0.1%
CPU Usage	Utilization rate of CPU resources (mean/peak) under platform service	12%/80%
RAM Usage	Utilization rate of RAM resources (mean/peak) under platform service	1 GB/8 GB
Disk Throughput	The amount of data read and write from a disk per unit time in the absence of disk failure	400 MB/s
Network Throughput	The number of network data per unit time in the absence of network failures	9 MB/s

5.6. Comparison

Feature pairs with other frameworks are shown in Table 7. Compared with the existing frameworks in the automotive cybersecurity field, we have several different characteristics.

Table 7. The comparison between ICVTest and other frameworks.

Framework	Process	Test Case Set	Platform/Tool	Scenarios	Type
ICVTest	**Yes**	10	**Yes**	Penetration Testing	VPT
Q. Li et al. [28]	**No**	3	**No**	Risk Assessment	VPT
Shirvani et al. [29]	**No**	10	**No**	Risk Assessment	VPT
Schönhärl et al. [30]	**No**	3	**Yes**	Education	VPT
S. Li et al. [25]	**No**	4	**No**	Penetration Testing	IVSTP
F. Luo et al. [26]	**Yes**	1	**Yes**	Penetration Testing	IVSTP
K. He et al. [27]	**No**	2	**Yes**	Penetration Testing	IVSTP

- Testing Process. The majority of frameworks do not prioritize the testing flow. Instead, most existing frameworks focus on theoretical threat analysis and risk assessment. However, our proposed ICVTest offers a standardized penetration testing procedure, which mitigates variances in test conclusions, enabling even novice testers to swiftly initiate penetration testing.

- Comprehensive Test Case Set. ICVTest abstract automotive security test scenarios into 10 sets that correspond to different fields of software and hardware architecture in vehicles, which allows for the reuse of test cases. The ability of 10 fields of vehicle cybersecurity threat modeling is not available in other frameworks.
- Integrated platform. Most frameworks only theoretically point out the threats that should be considered during security assessment, and not actual tools or platforms provided for security testing. In ICVTest, we have developed a cybersecurity testing platform to facilitate the testing task for security testers.
- Industry Scenario. The ICVTest we propose differs from the one used in educational scenario, as it demonstrates superior performance in real-world automobile cybersecurity testing scenario. While some frameworks theoretically address threat analysis and risk assessment during the vehicle design phase, they fail to tackle the issue of cybersecurity testing during the verification phase effectively. Consequently, verifying whether the identified threats indeed manifest as exploitable vulnerabilities becomes unfeasible for these frameworks.
- Vehicle Penetration Testing. ICVTest is intended for comprehensive vehicle cybersecurity testing, which includes in-vehicle subsystem cybersecurity testing as well. Furthermore, the test cases used in ICVTest are compatible with those of the in-vehicle subsystem cybersecurity test framework.

6. Conclusions

In this paper we proposed ICVTest, a novel black-box penetration testing framework, which is the first of its kind to focus on comprehensive vehicle-oriented security evaluation and draws upon extensive experience in automotive penetration testing. The penetration testing process in ICVTest is structured into six steps of specification, enabling even novice testers to be efficiently guided through the initiation phase. Additionally, leveraging extensive testing experience, ICVTest abstracts automotive cybersecurity test cases into 10 sets. Testers only need to conduct threat analysis on the target object and select appropriate test cases from predefined test case sets based on the results of threat analysis to evaluate vehicle security. Furthermore, we have implemented a comprehensive cybersecurity testing platform that seamlessly integrates testing tools and guide manuals to facilitate testers in conducting penetration testing efficiently.

Ten test case sets in ICVTest are highly decoupled from the specific in-vehicle system. Despite variations in electronic and electrical architectures among vehicles, their software and hardware security assessments can be aligned with the 10 test case sets of ICVTest. By leveraging ICVTest, testers only need to reuse appropriate test cases based on the threat analysis results of the object under test. While ICVTest itself does not delve into specific testing methods, it offers a set of systematic guidance methods for evaluating vehicle security based on practical penetration testing experience. Hierarchical test case sets of ICVTest are highly versatile, providing guidance not only to testers with varying levels of experience but also for the security evaluation of diverse vehicles and on-board systems.

We applied the ICVTest to evaluate a physical vehicle manufactured in 2021, which facilitated efficient identification of multiple cybersecurity vulnerabilities encompassing the in-vehicle CAN bus, sensors, infotainment systems, and remote vehicle controls. Our experimental results demonstrate that employing methodology provided by ICVTest is helpful to conduct the cybersecurity assessment of ICV, while it reduces the complexity associated with penetration testing and enhancing overall efficiency.

In future, the ICVTest and the cybersecurity test platform still require further enhancements. (1) Although the framework divides vehicle penetration testing into 10 fields, continuous investigation and quality improvement is needed to determine the specific content that should be tested at each field in order to effectively address the emerging cybersecurity risks and to identify the unknown vulnerabilities. (2) The majority of tests within ICVTest are currently triggered by the real-time participation from testers, resulting in a low degree of automation. To enhance the platform's automated testing capabilities, it

is necessary to develop the automated testing scripts for different test cases in the future. (3) Presently, we only mentioned AI security in sensor security testing, such as adversarial example-based traffic sign recognition robustness testing in camera security. However, as high-level autonomous driving becomes more prevalent in the future, there may be potential for separating AI security testing from sensor security testing.

Author Contributions: Conceptualization, J.W. and Z.L.; Methodology, H.Z. and J.W.; Software, H.Z. and Y.W.; Validation, M.L.; Formal analysis, J.S.; Investigation, Y.W., M.L. and J.S.; Writing—original draft, H.Z.; Writing—review & editing, J.W. and Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a grant of key technologies RD general program of Shenzhen, No. JSGG20201102170601003.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the Testing Confidentiality Agreement.

Acknowledgments: This research is supported by key technologies RD general program of Shenzhen, No. JSGG20201102170601003, which was awarded to Shenzhen Kaiyuan Internet Security Co., Ltd. for the research on cybersecurity testing of intelligent connected vehicles. Researchers from Xiamen University of Technology and Huazhong University of Science and Technology participated in this research project based on similar research interests. Shenzhen Kaiyuan Internet Security Co., Ltd. proposed the framework based on the testing requirements of the industry and implemented some experiments. Huazhong University of Science and Technology has completed theoretical research, manuscripts, and some experiments. Xiamen University of Technology implemented some experiments.

Conflicts of Interest: Authors Haichun Zhang, Jie Wang, Minfeng Li and Jinghan Song were employed by the company Shenzhen Kaiyuan Internet Security Co., Ltd. The remaining authors declare that the re-search was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Chattopadhyay, A.; Lam, K.Y.; Tavva, Y. Autonomous vehicle: Security by design. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 7015–7029. [CrossRef]
2. Hataba, M.; Sherif, A.; Mahmoud, M.; Abdallah, M.; Alasmary, W. Security and Privacy Issues in Autonomous Vehicles: A Layer-Based Survey. *IEEE Open J. Commun. Soc.* **2022**, *3*, 811–829. [CrossRef]
3. Li, J.; Zhang, M.; Lai, Y. A light-weighted machine learning based ECU identification for automative CAN security. In Proceedings of the 2023 International Conference on Networking and Network Applications (NaNA), Qingdao, China, 18–21 August 2023.
4. Rathore, R.S.; Hewage, C.; Kaiwartya, O.; Lloret, J. In-vehicle communication cyber security: Challenges and solutions. *Sensors* **2022**, *22*, 6679. [CrossRef] [PubMed]
5. Ma, B.; Yang, S.; Zuo, Z.; Zou, B.; Cao, Y.; Yan, X.; Zhou, S.; Li, J. An authentication and secure communication scheme for in-vehicle networks based on SOME/IP. *Sensors* **2022**, *22*, 647. [CrossRef] [PubMed]
6. Francia, G.A. Connected vehicle security. In Proceedings of the International Conference on Cyber Warfare and Security (ICCWS 2020), Norfolk, VA, USA, 12–13 March 2020.
7. Anwar, A.; Anwar, A.; Moukahal, L.; Zulkernine, M. Security assessment of in-vehicle communication protocols. *Veh. Commun.* **2023**, *44*, 100639. [CrossRef]
8. Hariharan, S.; Papadopoulos, A.V.; Nolte, T. On in-vehicle network security testing methodologies in construction machinery. In Proceedings of the 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA), Stuttgart, Germany, 6–9 September 2022.
9. Kang, L.; Shen, H. Detection and mitigation of sensor and CAN bus attacks in vehicle anti-lock braking systems. *ACM Trans. Cyber-Phys. Syst. (TCPS)* **2022**, *6*, 1–24. [CrossRef]
10. Rajapaksha, S.; Kalutarage, H.; Al-Kadri, M.O.; Petrovski, A.; Madzudzo, G.; Cheah, M. Ai-based intrusion detection systems for in-vehicle networks: A survey. *ACM Comput. Surv.* **2023**, *55*, 1–40. [CrossRef]

11. Saber, O.; Mazri, T. Security of Autonomous Vehicles: 5g Iov (internet of Vehicles) Environment. The International Archives of the Photogrammetry. *Remote Sens. Spat. Inf. Sci.* **2022**, *48*, 157–163.
12. Mudhivarthi, B.R.; Thakur, P.; Singh, G. Aspects of cyber security in autonomous and connected vehicles. *Appl. Sci.* **2023**, *13*, 3014. [CrossRef]
13. Hallyburton, R.S.; Liu, Y.; Cao, Y.; Mao, Z.M.; Pajic, M. Security Analysis of {Camera-LiDAR} Fusion Against {Black-Box} Attacks on Autonomous Vehicles. In Proceedings of the 31st USENIX Security Symposium (USENIX Security 22), Boston, MA, USA, 10–12 August 2022.
14. *ISO/SAE 21434: 2021*; Road Vehicles: Cybersecurity Engineering. International Organization for Standardization ISO: Geneva, Switzerland, 2021.
15. Potter, B.; McGraw, G. Software security testing. *IEEE Secur. Priv.* **2004**, *2*, 81–85. [CrossRef]
16. Prathap, V.; Rachumallu, A. Penetration Testing of Vehicle ECUs. Master's Thesis, Chalmers University of Technology, Gothenburg, Sweden, 2013.
17. Schönhärl, S.; Fuxen, P.; Graf, J.; Schmidt, J.; Hackenberg, R.; Mottok, J. An Automotive Penetration Testing Framework for IT-Security Education. *Cloud Comput.* **2022**, *2022*, 10.
18. Ebert, C.; Ray, R. Penetration Testing for Automotive Cybersecurity. *ATZelectron. Worldw.* **2021**, *16*, 16–22. [CrossRef]
19. Dürrwang, J.; Braun, M.; Kriesten, R. Pretschner. Enhancement of automotive penetration testing with threat analyses results. *SAE Int. J. Transp. Cybersecur. Priv.* **2018**, *1*, 91–112. [CrossRef]
20. Bayer, S.; Enderle, T.; Oka, D.K.; Wolf, M. Security Crash Test-Practical Security Evaluations of Automotive Onboard It Components. In Proceedings of the Automotive—Safety & Security 2014, Stuttgart, Germany, 21–22 April 2015; pp. 125–139.
21. Mahmood, S.; Nguyen, H.N.; Shaikh, S.A. Systematic threat assessment and security testing of automotive over-the-air (OTA) updates. *Veh. Commun.* **2022**, *35*, 100468. [CrossRef]
22. Utting, M.; Pretschner, A.; Legeard, B. A taxonomy of model-based testing approaches. *Softw. Test. Verif. Reliab.* **2012**, *22*, 297–312. [CrossRef]
23. Santos, E.D.; Simpson, A.; Schoop, D. A formal model to facilitate security testing in modern automotive systems. *Electron. Proc. Theor. Comput. Sci.* **2018**, *271*, 95–104. [CrossRef]
24. Mahmood, S.; Fouillade, A.; Nguyen, H.N.; Shaikh, S.A. A Model-Based Security Testing Approach for Automotive Over-The-Air Updates. In Proceedings of the 2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Porto, Portugal, 24–28 October 2020.
25. Li, S.; Zhang, X.; Zhou, Y.; Zhang, M. SP-E: Security Evaluation Framework of In-vehicle Infotainment System based on Threat Analyses and Penetration Tests. *J. Phys. Conf. Ser.* **2023**, *2517*, 012012. [CrossRef]
26. Luo, F.; Zhang, X.; Hou, S. Research on Cybersecurity Testing for In-vehicle Network. In Proceedings of the 2021 International Conference on Intelligent Technology and Embedded Systems (ICITES), Chengdu, China, 23 September 2022.
27. He, K.; Wang, C.; Han, Y.; Fang, X. Research on cyber security Technology and Test Method of OTA for Intelligent Connected Vehicle. In Proceedings of the 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Virtual Conference, China, 16 July 2022.
28. Li, Q.; Zuo, J.; Cao, R.; Chen, J.; Liu, Q.; Wang, J. A Security Evaluation Framework for Intelligent Connected Vehicles Based on Attack Chains. In *IEEE Network*; IEEE: New York, NY, USA, 2023; p. 1. [CrossRef]
29. Shirvani, S.; Baseri, Y.; Ghorbani, A. Evaluation Framework for Electric Vehicle Security Risk Assessment. *IEEE Trans. Intell. Transp. Syst.* **2023**, 1–24. [CrossRef]
30. Arkin, B.; Stender, S.; McGraw, G. Software penetration testing. *IEEE Secur. Priv.* **2005**, *3*, 84–87. [CrossRef]
31. Scarfone, K.; Souppaya, M.; Cody, A.; Orebaugh, A. Technical guide to information security testing and assessment. *NIST Spec. Publ.* **2008**, *800*, 2–25.
32. Checkoway, S.; McCoy, D.; Kantor, B.; Anderson, D.; Shacham, H.; Savage, S.; Koscher, K.; Czeskis, A.; Roesner, F.; Kohno, T. Comprehensive experimental analyses of automotive attack surfaces. In Proceedings of the 20th USENIX Security Symposium (USENIX Security 11), San Francisco, CA, USA, 8–12 August 2011.
33. Persial, J.P.; Prabhu, M.; Shanmugalakshmi, R. Side channel attack-survey. *Int. J. Adva. Sci. Res. Rev.* **2011**, *1*, 54–57.
34. Devi, M.; Majumder, A. *Side-Channel Attack in Internet of Things: A Survey*; Springer: Singapore, 2021; pp. 213–222.
35. Le, T.H.; Canovas, C.; Clédiere, J. An overview of side channel analysis attacks. In Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, Tokyo, Japan, 18–20 March 2008.
36. Agrawal, D.; Archambeault, B.; Rao, J.R.; Rohatgi, P. The EM side—channel(s). In Proceedings of the International workshop on Cryptographic Hardware and Embedded Systems, Redwood Shores, CA, USA, 13–15 August 2002.
37. Asadi, G.; Miremadi, S.G.; Zarandi, H.R.; Ejlali, A. Fault injection into SRAM-based FPGAs for the analysis of SEU effects. In Proceedings of the 2003 IEEE International Conference on Field-Programmable Technology (FPT), Tokyo, Japan, 17 December 2003.
38. Bozzato, C.; Focardi, R.; Palmarini, F. Shaping the glitch: Optimizing voltage fault injection attacks. IACR Transactions on Cryptographic. *Hardw. Embed. Syst.* **2019**, *2019*, 199–224.

39. Moro, N.; Dehbaoui, A.; Heydemann, K.; Robisson, B.; Encrenaz, E. Electromagnetic fault injection: Towards a fault model on a 32-bit microcontroller. In Proceedings of the 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, Alamitos, CA, USA, 20 August 2013.
40. Van Woudenberg, G.J.; Witteman, M.F.; Menarini, F. Practical optical fault injection on secure microcontrollers. In Proceedings of the 2011 Workshop on Fault Diagnosis and Tolerance in Cryptography, Nara, Japan, 28 September 2011.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.