Automotive Cybersecurity Testing: Survey of Testbeds and Methods

Shahid Mahmood, Hoang Nga Nguyen and Siraj A. Shaikh

Abstract Computing and connectivity capabilities in modern cars have introduced new cybersecurity challenges that can potentially affect the safety of an automobile and its occupants. Effective cybersecurity testing of vehicles can play a crucial role in discovering and addressing security flaws; however testing a real vehicle (involving cyber-physical components) carries safety and economic risks. Therefore, many researchers and practitioners rely on testing environments (commonly known as testbeds) for uncovering cybersecurity vulnerabilities. Effective and efficient security testing needs the application of appropriate and systematic testing methods. This study presents a survey of seven different automotive cybersecurity testbeds proposed over the last ten years (between 2012 and 2019) as well as four different types of cybersecurity testing methods employed by cybersecurity researchers. This survey will help students, researchers, and professionals with designing and building their testbeds and refining their testing techniques and methodologies.

1 Introduction

Contemporary vehicles are increasingly vulnerable to attacks due to the embedded computing and internet connectivity capabilities they are equipped with. As cyber-

Shahid Mahmood
e-mail: mahmo136@coventry.ac.uk

Hoang Nga Nguyen
e-mail: hoang.nguyen@coventry.ac.uk

Siraj A. Shaikh
e-mail: siraj.shaikh@coventry.ac.uk
Systems Security Group,
Institute for Future Transport and Cities,
Coventry University,
Coventry CV1 5FB, UK

attacks have the potential to seriously undermine the safety of an automobile and its occupants, effective testing for detecting software flaws and weaknesses is crucial. However, cybersecurity testing of automobiles is not always feasible due to their technical complexity, physical size, safety risks, and high financial costs. In order to overcome these challenges, cybersecurity researchers and professionals often rely on testing environments (commonly referred to as testbeds) by mainly using virtual devices and sometimes real components as well.

Although, using real devices/components in the cybersecurity testing can provide very high degree of fidelity; there are safety and financial ramifications to consider carefully. While testbeds provide a conducive and safe testing environment, appropriate security testing techniques help tremendously in identifying cybersecurity threats in a systematic way. We present a review of seven major automotive cybersecurity testbeds and four security testing approaches that are widely used in the field of automotive cybersecurity.

1.1 Motivation, Objective and Scope of this Study

To the best of our knowledge, there are no prior studies that survey automotive cybersecurity testbeds and testing methods. Therefore, this study aims at closing this gap by providing a comprehensive survey of the recent developments in automotive cybersecurity testbeds and testing approaches. This study does not explore or consider testbeds designed for cybersecurity testing of autonomous vehicles. Moreover, we have deliberately not included works involving real vehicles (e.g., [45]) as a testing platform. The reason for excluding such setups is because they target a very specific make and model of a certain real vehicle. The main objective of this study is to find answers to the following questions: Over the last ten years, What testbeds have been proposed for automotive cybersecurity testing? What are the key characteristics, strengths, and weaknesses of each testbed? What methods have been proposed and widely used for automotive cybersecurity testing?

1.2 Outline

This paper is organized as follows: In section 2, background of the automotive cybersecurity is presented, providing an overview of different cybersecurity threats that modern vehicles are vulnerable to. Section 3 provides an overview of the related work, highlighting similar studies that survey cybersecurity testbeds and methods in other domains. Section 4 discusses seven different testbeds proposed over the last ten years for automotive cybersecurity testing. This is followed by section 5, which discusses and compares various attributes of the testbeds including adaptability, portability, fidelity, safety and cost. Four different types of cybersecurity testing

approaches for automotive security evaluation are presented in section 6 followed by the conclusion in section 7.

2 Automotive Cybersecurity

Software flaws or vulnerabilities in the vehicle can lead to serious consequences, ranging from incidents of information theft to life-threatening situations. Numerous previous studies show how a vehicle can be maliciously controlled by exploiting one or more weaknesses in its software systems. Koscher et al. [45] demonstrate that it is possible for an adversary to maliciously influence a car's behaviour (e.g., engaging or disengaging its brakes) if they are able to access the car's internal network.

A connected vehicle may have several internal and external connections for accomplishing various important tasks. While these connections support correct functioning of different applications in the car, they can be exploited by cybercriminals to launch cyberattacks targeting various digital systems in the vehicle. Some of the external connections to the vehicle include cellular network, WiFi, Bluetooth, Keyless Entry System, KES, and Tyre Pressure Monitoring System (TPMS). Whereas, Onboard Diagnostic II (OBD II) port, USB, and in-vehicle infotainment are some of the internal connections [54]. This section provides an overview of some of the common cybersecurity threats and risks faced by modern automobiles.

2.1 CAN Bus

Most cars today come equipped with a variety of computing devices, known as Electronic Control Units (ECUs). A typical modern vehicle may contain a number of different ECUs, each of which has unique responsibilities for performing one or more functions of the vehicle. For example, one ECU may be responsible for detecting whether there is a passenger present in the vehicle, whereas another one may be monitoring the tyre pressure. In order to perform their duties correctly, these ECUs often need to communicate with each other as well as external world [35].

For local communication, ECUs rely on various automotive networking technologies including Controller Area Network (CAN), Local Interconnect Network (LIN), Media Oriented Systems Transport (MOST), and FlexRay. Each of these technologies has been designed to meet specific needs of a particular automotive application. For instance, MOST is a high-speed network technology to support audio, video, and voice data communications. Whereas, LIN is used in automotive applications requiring low network bandwidth and speed, such as mirror control and door lock/unlock features [26].

Although CAN is a robust and fault-tolerant network technology, it lacks any security mechanisms because it has not been designed with security in mind [7]. Therefore, it is vulnerable to numerous cybersecurity threats which have been re-

ported in many existing studies, such as [47, 28, 48]. For example, an adversary could gain access to the internal network of the vehicle remotely followed by injecting messages onto it in order to compromise and control a target ECU. Once an ECU is compromised, the attacker can potentially control the safety-critical functions of the car, such as braking, acceleration, and steering. It is however important to note that safety-critical components usually reside on a network that is separate from other non-critical components. Nevertheless, hackers may still be able to break into these networks by leveraging a gateway ECU [37].

Other examples of CAN bus exploitation include installation of a malicious diagnostic device to send packets to the CAN bus, using CAN bus to start a vehicle without a key, leveraging the CAN bus to upload malware, installing a malicious diagnostic device in order to track the vehicle and enable remote communications directly to the CAN bus [31].

By compromising one of the connections listed above, cybercriminals can potentially attack in-vehicle systems in order to take over a vehicle remotely, shut down it, unlock it, track it, thwart its safety systems, install malware on it, or spy on its occupants. For example, an adversary can access the vehicle's internal network or the remote diagnostic system remotely by means of cellular connection. Similarly, an attacker can exploit the WiFi connection for gaining access to the vehicle network (from up to 300 yards), intercepting data traffic of the WiFi network, breaking the WiFi password and more [54].

2.2 In-vehicle Infotainment

An In-Vehicle Infotainment (IVI) or an automotive infotainment system is an integrated unit, providing information services and entertainment functionality to the driver and other vehicle occupants for an enhanced in-vehicle experience.

Infotainment systems, one of the major attack vectors in connected cars, are growing both in terms of their capabilities and popularity. As more and more features are being added to infotainment systems, this will likely to increase the number of new vulnerabilities, attack vectors, and threats that can undermine the privacy and safety of the vehicle and its occupants. Typically, an IVI is interconnected with the CAN bus for communicating with other devices. From cybersecurity perspective, this connectivity may have serious implications. Prior studies have evidenced that cybercriminals can target automotive infotainment systems for mounting sophisticated attacks on automobiles [44].

An attacker can exploit weaknesses in the infotainment system or can use it as an entry point to gain access to in-vehicle network, thus to safety-critical features of the vehicle. Some possible use cases include utilising a remote connection to the infotainment system for exploiting the application in the IVI responsible for handling incoming calls, accessing the subscriber identity module (SIM) through the IVI, installing malicious code on the infotainment system, putting the infotainment console into debug mode, using a malicious application to access the internal CAN

bus network, using a malicious application to eavesdrop on actions taken by vehicle occupants.

2.3 Onboard Diagnostic (OBD) Port

Modern vehicles have Onboard Diagnostic (OBD) ports inside them that are used for ECU firmware updates, vehicle repairing and inspections. Implementation of these ports is obligatory since 1998 in the USA and since 2001 for gasoline-powered vehicles and since 2003 for diesel-powered vehicles in the EU respectively [55]. Onboard Diagnostic is mainly used for reporting the data gathered by various sensors in the car to the outside world, providing information on the health status of the vehicle. This information is often used by service providers for fixing any reported problems [16]. Since inexpensive OBD dongles are readily available in the market, attackers can leverage them as an entry point for breaking into in-vehicle networks.

Nilsson and Larsan in [41] demonstrate how a virus can be injected on to the CAN bus through the OBD port that issues some messages for controlling some aspects of the vehicle behaviour (e.g., locks, brakes, etc.) if certain conditions are found to be true.

Unlike the attack mentioned above in [41], which requires physical access to the vehicle, many modern automobiles allow remote access to these dongles via WiFi connections from a computer, allowing adversaries to launch cyberattacks remotely. As reported in a survey [58], more than 50% of the surveyed dongles, were found to be containing vulnerabilities (e.g., exposed keys, weak encryption), which can be exploited by cybercriminals to compromise the security of a vehicle.

3 Related Work

To the best of our knowledge, there is no prior published study that surveys automotive cybersecurity testbeds and testing methods. A few previous studies, such as [56, 43] describe exiting testbeds for automotive cybersecurity testing, but they are limited to very brief, high level descriptions only. For example, Toyama et al. [56] compare their proposed testbed with some existing testing environments [20, 36, 39] and briefly outline their strengths and limitations. Likewise, brief descriptions of some existing testbeds have been presented by Oruganti et al. in [43]. Similarly, there is no prior known work that presents a survey of the testing methods in automotive cybersecurity.

A number of studies surveying testbeds, and some others reviewing security testing techniques in other domains do exist. In this section, we briefly describe some of those surveys.

Holm et al. [27] present a survey of 30 Industrial Control System (ICS) testbeds primarily focusing on facilitating vulnerability analysis, test and education of defense

mechanisms. The study aimed at investigating what ICS testbeds exist, what specific ICS objectives they propose, how ICS components are implemented into these, and how they manage testbed requirements. Cintulu et al. [13] provide a survey on cyber-physical smart grid testbeds along with a taxonomy based on smart grid domains as well as a set of guidelines for developing the testbed. Their survey include a detailed discussion and evaluation of existing smart grid testbeds. Furthermore,they also outline future trends and possible developments in cyber-physical smart grid testbeds.

A comprehensive survey of cybersecurity testing approaches in SCADA systems has been presented by Nazir et al. [40] They provide an overview of various common vulnerabilities that may exist in many SCADA systems. Additionally, among other approaches, authors describe and discuss model-based and simulation-based methods and frameworks for cybersecurity testing of SCADA systems. A number of SCADA testbeds have also been discussed in the study. Approaches, such as machine learning and penetration testing have also been discussed in the context of SCADA systems testing. Finally, the authors highlight some recent developments and future trends in the domain, specifically referring to cloud computing, virtualization, software defined networks (SDN), as well as open standards such as OpenSCADA.

4 Overview of Automotive Cybersecurity Testbeds

In this section we present an overview of seven different automotive cybersecurity testbeds that have been proposed in the last ten years.

Testbeds can generally be categorised in three different types: simulation based, hardware based, and hybrid. Simulation-based testbeds rely solely or substantially on software to simulate the behaviour of ECUs and in-vehicle networks. Since they do not include real cyber-physical components, simulation-based testbeds are generally cheaper to build, and provide a safer environment for the testers. Hardware-based testbeds, on the other hand, include real or emulated hardware components. As opposed to software-based testbeds, hardware-based testbeds enable testers to study interactions between components through physical inputs and outputs. Hybrid testbeds include both software and hardware components, offering strengths of simulation-based and hardware-based testbeds.

Table 1 presents an overview of the surveyed testbeds, indicating whether they are simulation-based, hardware-based, or hybrid. Mobile testing platform from [36] is the only testbed that uses real physical components and a vehicle (go-cart) for investigating cybersecurity threats. OCTANE and the Testbed for Security Analysis of Modern Vehicle Systems are hybrid testing environments. All other testbeds rely on virtual/software components only.

Table 1 Types of Automotive Cybersecurity Testbeds

Name of Testbed	Test Platform	Year	Reference
Open Car Testbed and Network Experiments (OCTANE)	Hybrid	2013	[6]
Mobile Testing Platform	Hardware	2015	[36]
Cyber Assurance Testbed for Heavy Vehicle Electronic Controls	Simulator	2016	[14]
Testbed for Automotive Cybersecurity	Simulator	2017	[22]
Testbed for Security Analysis of Modern Vehicle Systems	Hybrid	2017	[59]
Portable Automotive Security Testbed with Adaptability (PASTA)	Simulator	2018	[56]
Hardware-in-loop based Automotive Embedded Systems Cybersecurity Evaluation Testbed	Simulator	2019	[43]

4.1 OCTANE: Open Car Testbed and Network Experiments

Open Car Testbed and Network Experiments (OCTANE) [6] includes a hardware framework and a software package providing capabilities to reverse engineer and test automotive networks. In particular, the tool can be used for fuzz testing various proprietary vehicular network protocols.

The software package allows transmission and monitoring of CAN messages for general purpose network diagnostic and debugging as well as automated replay testing of Electronic Control Units (ECUs), whereas the hardware framework assists in setting up hardware components of the automotive networks for two main different configurations: lab setup and real-world setup. The hardware framework outlines a structured step-by-step approach to set up a particular environment without prescribing any specific type of hardware components.

The testbed has been designed to enable entry into automotive cybersecurity testing research and teaching in a safe and cost-effective way. In order to maintain clear separation of concerns between software and hardware components, the hardware middle layer plays a pivotal role. This makes it easy to add a new hardware adapter to replace the existing one without affecting other layers. In order to enable adaptability and flexibility, the software package has been designed using a layered architecture consisting of a presentation layer, a business layer composed of a processing layer and a thread layer, a hardware middle layer, and a hardware layer.

This testbed allows security testing of various vehicular network protocols including CAN, LIN, MOST and FlexRay. An appropriate adapter needs to be used when working on a specific network technology. One of the main limitations of the testbed is that it only uses the OBD port as an attack surface. The testbed is not capable of testing vulnerabilities related to wireless connectivity. We observe that although the source code is available to download on the Google Code platform, we were unable to find any related documentation.

4.2 A Mobile Testing Platform

Miller and Valasek [36] implemented a mobile testbed by modifying a go-cart to emulate a real vehicle. They equipped it with various ECUs and sensors, which help study the behaviour of actual devices in an economical way. As compared to a real vehicle, there is low financial risk involved, because the go-cart is much cheaper than a real vehicle. However, risk of physical injuries is still present as it is a moving vehicle.

One of the major capabilities of the real vehicle they included was a power steering control module (PSCM) on the go-cart. Additionally, they integrated different sensors including proximity and speed sensors. A real pre-collision system was also Incorporated for actual distance readings while the vehicle is in motion. While using a moving vehicle instead of a bench setup certainly enabled the testers to study the behaviour of the moving vehicle which a testbed set up on a bench is incapable of, there are some shortcomings as well that the original developers of the environment identified.

While real components were used in the go-cart, they may not represent the complete functionality and behaviour of an actual car. For example, the developers note that PSCM does not work properly after some right and left turns, as the it enters its final state. Another limitation reported by authors is steering wheel radius that does not allow steering to be controlled by the CAN bus. This limitation was also hurdle for auto-park capability, which otherwise could have been realized. Finally, remodelling of the go-cart vehicle is another key challenge for the researchers who may be interested in using this setup. To summarize, while this mobile testing platform enables the tester to evaluate the impact of of cyberattacks on the moving vehicle cost-effectively, it has some considerable limitations as well.

4.3 A Cyber Assurance Testbed for Heavy Vehicle Electronic Controls

This testbed [14] has mainly been proposed for cybersecurity testing of heavy vehicles remotely. It primarily supports J1939 networks that are found in heavy vehicles including buses and trucks. Authors used real ECUs, Linux-based, simulated node controllers for their testing setup.

The testbed allows the researcher to study and manipulate the network traffic by providing various features. For example, one of the distinctive characteristics of this testbed is the capability of remote experimentation, which allows researchers to access the data remotely without physically interaction with the vehicle. For this purpose, the authors of the testbed introduced a custom-built five-layer application. The five layers are web interface, experiment processing, experiment logic, CAN data processor and a database for experiment and J1939 data. The web interface layer allows the tester/researcher to interact with ECUs for monitoring and modifying network traffic. Experiment processing layer is responsible for converting the CAN

messages into a human readable format. The database layer is mainly used for storing the experiment data.

4.4 Testbed for Automotive Cybersecurity

Fowler et al. [22] built a testbed for automotive cybersecurity testing consisting of an established industry, real-time CAN simulator from Vector Informatik.

The simulator along with its associated software CANoe is widely used in the automotive industry primarily by automakers for the development and testing of ECUs. The simulator provides CAN data traffic monitoring, capturing, and analysis capabilities, which help in reverse engineering of vehicles. To validate the testbed, CAN message-injection was performed by using a Bluetooth-enabled dongle connected to an OBD port on the simulator. The messages were successfully injected validating the correct functioning of the testbed. The description presented in the paper is limited to some high level information only without going into details about the architecture and other characteristics of the testbed.

4.5 Testbed for Security Analysis of Modern Vehicles

Zheng et al. [59] developed a prototype of their proposed testbed, which was built around a real-time CAN bus simulator using dedicated hardware from National Instruments and a simulated vehicular infotainment system (using LabVIEW software).The testbed is able to capture CAN messages for security analysis and can inject malicious messages through simulated infotainment system.

It is argued by the authors that while use of real vehicles or vehicle components for testing is more effective and produce accurate results, such test environments provide little or no flexibility in terms of their configurations. The proposed testbed by Zheng et al. is reconfigurable, enabling the testbed to replicate many test configurations. Furthermore, the testbed is able to reproduce the complexity of interconnected ECUs in the in-vehicle network.

The authors performed a denial-of-service attack targeting the CAN bus by leveraging the emulated infotainment system as an entry point into the in-vehicle network. A dump containing a large number of previously captured CAN messages was injected causing the CAN bus to fail to operate properly by rejecting legitimate CAN messages. This testbed is reconfigurable, inexpensive to reproduce, and provides a safe environment for automotive cybersecurity testing. However, being largely a simulated environment its obvious limitation is the lack of physical input and output ports which seriously affects security evaluation requiring these ports.

4.6 PASTA: Portable Automotive Security Testbed with Adaptability

Portable Automotive Security Testbed with Adaptability (PASTA) [56] is another automotive security testbed with a special focus on white-box ECUs, high adaptability and portability. Authors explain why white-box ECUs can be more effective when it comes to automotive cybersecurity testing. First of all, white-box ECUs provide the ability to observe their inputs and outputs as well as disassembly of the ECU programs without involving any suppliers or OEMs. Secondly, ECUs can be reprogrammed and rearranged in a number of different configurations in the automotive networks allowing evaluation of the security technology against cyberattacks. Finally, the ability to modify different parameters, such as CAN ID, payload, or transmission cycle enables the reproducibility of a commercial vehicle.

The authors outline the requirements that they considered while designing their proposed testbed. The first factor is the cost of the testbed, which is typically very high when involving a real vehicle containing a variety of ECUs and in-vehicle networks. High financial cost is one of the barriers to automotive cybersecurity research. To minimize the cost of their testbed, they eliminated expensive sensors and other similar components including simulators such as speed, angle of tyres, and status of headlights. Authors argue that such expensive components are not essential for cybersecurity testing.

Portability of the testbed is another key consideration; PASTA has been designed with portability in mind, its compact size allows it to be easily carried to different places for demonstrating research experiments and results. Another aspect of the testbed is the generalizability of the vehicle to ensure the testing is not restricted to a specific make and model.

Safety is also critical aspect, especially when the testing involves physical subsystems, such as actuators, which can behave in an unexpected way causing injuries to the researchers or any other parties involved. Such safety risks can be addressed by using an emulated actuator instead of a real one, for example. Finally, the target testbed must be designed in a way that it supports the learning of all the stakeholders, especially software developers, because software bugs and flaws in the software design due to human error often result in catastrophic consequences.

PASTA, according to its designers, meets all the requirements outlined above. The testbed can be customized to fulfill specific needs of a researcher, as it has been designed using non-proprietary technologies.The testbed allows testers to use custom security technology and provides the flexibility to design the in-vehicle network as per their particular requirements.

While PASTA is safe, flexible, portable and adaptable, it has some shortcomings too. Its software vehicle simulator is not able to replicate a vehicle's behaviour accurately. As the designers of PASTA have noted that speed of the vehicle reaches 199 km/hour in a very short time when the acceleration is applied, which is obviously not reflective of true behaviour of a real vehicle. The software needs tweaking to resolve this issue. Another limitation of PASTA is that it currently supports CAN protocol only. Other protocols such as LIN, FlexRay, and MOST are not supported. OBD-II port and a tapped CAN cable are the only physical intrusion points that

PASTA provides for launching attacks on the CAN bus. Moreover, it currently lacks attack surfaces such as Bluetooth, WiFi, and cellular networks. Finally, the software architecture for the implementation of ECUs environment is not Automotive Open System Architecture compliant.

4.7 Hardware-in-loop based Automotive Embedded Systems Cybersecurity Evaluation Testbed

Oruganti et al. [43] propose a testbed for automotive cybersecurity testing. Authors report their current progress towards the development of their testbed which will include hardware-in-loop components. The current testbed is completely a virtual setup, thus limited to a software simulation only. Using this virtual testbed, the authors demonstrate a GPS location spoofing attack on a virtual vehicle. The authors list essential elements of the testbed that should be present, which include connectivity, vehicular networks, controller modeling and algorithm implementation, hardware-in-loop and telematics. Each of these subsystems allows the cybersecurity evaluation and validation of a connected car for a range of attack surfaces and attack vectors.

5 Evaluation of Testbeds

This section compares the reviewed testbeds based on their various characteristics, such as adaptability, portability, fidelity and cost. An overview of other capabilities (e.g., types of attacks, attack surfaces, attack targets, and communication protocols supported) of the testbeds have also been discussed.

5.1 An Overview of Supported Network Protocols

Modern cars have multiple network types for facilitating various applications. Not all testbeds that have been surveyed offer support for testing all types of communication standards. Table 2 gives an overview of the protocols supported by each testbed. As can be noticed, OCTANE is the only testbed that claims to support testing for all major vehicular network protocols. All other testbeds do not cover any protocol other than CAN. This means they are unable to support study of threats/attacks related to other network standards found in modern automobiles.

Table 2 Overview of what types of in-vehicle network protocols are supported by each testbed for cybersecurity testing (* FLR stands for FlexRay and Ref. for Reference).

Testbed Name	CAN	LIN	FLR*	MOST	Ref.*
Open Car Testbed and Network Experiments (OCTANE)	✓	✓	✓	✓	[6]
Mobile Testing Platform	✓	N/A	N/A	N/A	[36]
Cyber Assurance Testbed for Heavy Vehicle Electronic Controls	✓	N/A	N/A	N/A	[14]
Testbed for Automotive Cybersecurity	✓	N/A	N/A	N/A	[22]
Testbed for Security Analysis of Modern Vehicle Systems	✓	N/A	N/A	N/A	[59]
Portable Automotive Security Testbed with Adaptability (PASTA)	✓	N/A	N/A	N/A	[56]
Hardware-in-loop based Automotive Embedded Systems Cybersecurity Evaluation Testbed	✓	N/A	N/A	N/A	[43]

Table 3 An overview of the types of exposed attack surfaces, types of attacks, target and/or goal of the attacks supported by each testbed.

Testbed Name	Attack Surface	Attack Type	Attack Target/Goal	Reference
Open Car Testbed and Network Experiments (OCTANE)	OBDII Port	Message sniffing Denial of Service (Dos) Replay	N\A	[6]
Mobile Testing Platform	OBDII Port	CAN Message Injection	Take over vehicle control	[36]
Cyber Assurance Testbed for Heavy Vehicle Electronic Controls	ECU, Ethernet, USB	Brute force, DoS	Evaluation of Seed\Key Exchange Strength and Intrusion Detection System	[14]
Testbed for Automotive Cybersecurity	OBDII Port	CAN Message Injection	Comfort Subsystem Manipulation (e.g., headlamp ON/OFF)	[22]
Testbed for Security Analysis ofModern Vehicle Systems	Infotainment Gateway	Can Sniffing, Code Injection, DoS	Control vehicle maneuver	[59]
Portable Automotive Security Testbed with Adaptability (PASTA)	OBDII Port/Clipping Area	Code Injection/Execution	N\A	[56]
Hardware-in-loop based Automotive Embedded Systems Cybersecurity Evaluation Testbed	Navigation System	GPS Spoofing	Spoof GPS Location	[43]

5.2 Testbeds and Supported Attack Surfaces, Types of Attacks, and Attack Goal

Table 3 highlights types of attack surfaces exposed by each testbed, types of attack supported or demonstrated, and attack target or goal. OBDII port is the most popular choice as an entry point into the in-vehicle network. This is probably due to the fact that all cars do have an OBD port, (since it is legal requirement to have one) OBD scanners are cheap and easily available in the market.

Similarly, most popular type of attack is message/code injection. This is obviously because CAN does not have an authentication or other security mechanism capable of identifying and rejecting malicious contents. Since many testbeds lack support for wireless/remote attack surfaces, they are only confined to testing attack scenarios assuming physical access to the vehicle.

Table 4 A comparative overview of the reviewed testbeds based on adaptability, portability, fidelity, safety and cost

(●●● = High, ●● = Medium, ● = Low)

Testbed Name	Adaptability	Portability	Fidelity	Safety	Cost
Open Car Testbed and Network Experiments (OCTANE) (real-world setup)	●	●	●●●	●	●●●
Open Car Testbed and Network Experiments (OCTANE) (lab setup)	●●	●●	●●	●●	●
Mobile Testing Platform	●	●●	●●	●	●●
Cyber Assurance Testbed for Heavy Vehicle Electronic Controls	●	●●	●●	●●●	●
Testbed for Automotive Cybersecurity	●●	●●	●	●	●●
Testbed for Security Analysis of Modern Vehicle Systems	●●	●●●	●	●●●	●
Portable Automotive Security Testbed with Adaptability (PASTA)	●●●	●●●	●●	●●●	●●
Hardware-in-loop based Automotive Embedded Systems Cybersecurity Evaluation Testbed	●●	●●●	●	●●●	●

Each testbed type has its own strengths and limitations. Ideally, a testbed should be able to reproduce the behaviour of a real vehicle as accurately and faithfully as possible (fidelity), adaptable, portable, safe, and inexpensive to construct, as

explained in [56]. In this section, we provide an overview of how and to what extent each of the testbeds meets the requirements of adaptability, portability, fidelity, safety, and cost-effectiveness.

Adaptability is a measure of a testbed's ability to support different testing configurations, i.e. how well a testbed can adapt to different configurations. Portability refers to how easy it is for the testbed to carry around. Fidelity of testbed can be measured by evaluating how accurately it can imitate the behaviour of a real vehicle in response to a an cyberattack. Testing environments involving cyber physical components have safety implications for the testers and any hardware/equipment involved; therefore it is vital to determine how a given testbed ensures the safety of testers and the equipment involved in the testing. Finally, cost of the entire setup can be compared based on whether it contains real physical devices or virtual components. Table 5.2 provides a comparison of how adaptable, portable, accurate, safe and costly each testbed is.

5.3 Adaptability

In-vehicle networks and ECUs are the major targets of cyber attacks, their security testing is important to identify and fix any security issues. Unfortunately, due to copyright restrictions and closed-source proprietary ECU technologies, it is very difficult to perform security testing on commercial ECUs. In addition, testing specific ECUs does not provide insights and results that can be helpful when testing the ECUs from different manufactures. Similarly, it is also important that the testbed is adaptable to a variety of testing configurations (e.g., with different vehicular network types) and not confined to a particular technology.

For instance, PASTA [56] includes white-box or programmable ECUs that allow the researcher to program an ECU to replicate the behaviour of a specific ECU, with the knowledge of internal implementation, which is usually not possible with real proprietary ECUs. Also, because it is largely software-based setup, it can be used for different testing configurations.

The layered-based design of the OCTANE [6] allows it to be adapted to different testing setups by replacing hardware components in the hardware middle layer without affecting other layers. Furthermore, its bespoke software package can be modified to extend its capabilities according to specific testing scenarios.

The prototype testbed proposed by Zheng et al. [59] has a flexible architecture allowing additional ECUs to be added easily. The setup can also be used for testing and investigating attacks launched via remote connections. Since it is entirely software-based, the testbed presented by Oruganti et al. [43] is adaptable to various testing configurations, as virtual components can be easily added or removed in the software environment.

Daily et al. have relied on actual ECUs and sensor simulations primarily focusing on J1939 based networks, which are specifically designed for heavy vehicles, such as trucks and buses. Although, the authors do not explicitly consider or discuss

adaptability of the testbed, based on the information provided, ***it seems probable for the testbed to be adapted to various testing configurations.***

5.4 Portability

A key factor to consider while designing or using a testbed is the portability. Sometimes it may be necessary to carry the testbed to a different location for demonstration purposes (e.g., in a conference or workshop). A testbed with compact or virtual components is obviously easy to carry around as opposed to the ones that include large actual components. Below we describe how each of the testbeds reviewed supports portability.

OCTANE has two types of main setups: lab based and real world. The lab-based testing environment typically relies on small components and does not involve real vehicle. So, it is possible to carry the lab setup as necessary with ease. However, in the case of a real-world testing setup, the portability depends on the actual components involved. The portability will be affected if, for example, a real car or heavy components are used. PASTA has been designed to be portable, so all its components are able to fit in a briefcase allowing high degree of mobility. Instead of using a real vehicle, a simulated or scale model of a real vehicle is a key factor in allowing this testbed to be more portable. The testbed from Zheng et al. contains simulated and emulated components so it should be easy to relocate if required.

Similar to the Zheng et al., the testbed from Oruganti et al. is purely a software-based environment which allows it to be moved around easily. The cyber assurance testbed by Daily et al. does not use a real vehicle, can be accessed remotely and uses simulated components with real ECUs, hence it satisfies the portability requirements.

5.5 Fidelity

Fidelity of a testbed refers to its ability to accurately reproduce the behaviour of a real vehicle or components in response to a specific event. To achieve high degree of fidelity, real vehicle and/or real hardware components must be included in the test. Software-based testing environments cannot faithfully reflect the conditions of a real car. Thus, fidelity of the test results is directly linked to the type of systems/components involved in the testing. Most importantly, complex interactions among various ECUs and other cyber-physical components inside the vehicle cannot be simply reproduce with high accuracy in a virtual environment.

The software vehicle simulator used in PASTA, for example, reaches 199km/h in a very short time which does not mimic the actual behaviour of the vehicle. While virtual, software-based testbeds have their own merits, they do not generally replicate actual behaviour of a real vehicle.

5.6 Cost

A virtual or software-based testbed is generally cheaper than a testbed which includes cyber physical components. OCTANE, and mobile testing platform (involving go-cart) rely on physical components, they are therefore more expensive. On the other hand, Zheng et al. Fowler et al. Oruganti et al. are software-based testbeds their cost is lower. PASTA and Daily et al. both contain ECUs, their cost will be higher than the pure software-based testbeds.

5.7 Safety Implications

Testing real vehicles help study the actual impact and behaviour of a vehicle as a result of a cyber attack. However, this has serious safety implications for the researcher and the vehicle under test. Physical safety of all stakeholders as well as of all the components/equipment involved must be the top priority. We look what safety implications each of the reviewed testbeds may have. In general, safety risks are high when a real vehicle or large cyber-physical components are used in the testing. The risk is even higher when the testing involves a moving vehicle on the road. While designing a new or using an existing testbed, it is a good idea to carefully consider any safety issues that can potentially surface.

OCTANE has two testing environments - laboratory-based and real world. In the lab-based setup, there are virtually no concerns related to human safety as it is a controlled environment with no real vehicle involved. The real-world testing setup potentially can lead to situations that can affect safety of both the vehicle and the testers.

Since [56], [59], [43] are primarily simulation based, these testbeds do not raise any safety concerns for the testers/researchers. Similarly, because the testbed from Daily et al. [14] is remotely accessible, it is safe to use.

It can be noticed that while software-based testbeds are generally more adaptable, portable, inexpensive, and safe, they however lack physical inputs and outputs (I/O) which may not be useful in the scenarios where evaluation of physical I/O is essential. Moreover, software-only testbeds do not provide accurate results and are often unable to reproduce the behaviour of actual systems.

6 Automotive Cybersecurity Testing Methods

While testbeds play a key role in security assessment of in-vehicle computing systems, effective testing methods are equally crucial for successful security evaluation of these systems. Knowledge of different testing approaches can be useful in choosing and applying the best possible technique for optimal results. We present a survey

of four different automotive cybersecurity testing approaches here, as at the time of this writing, there is no existing work presenting such a survey.

Interconnected computing components (i.e. ECUs) in a modern vehicle control various features including safety-critical functions, such as airbags, braking, acceleration etc. Attackers can exploit security loopholes in these systems to take over control, steal information, or cause damage to the vehicle and/or its occupants. Prior studies [12, 35, 45, 25] discuss different attack scenarios that are possible and practical. Therefore, thorough and systematic testing of automotive components is paramount.

There are effective approaches employed by cybersecurity testers, professionals and researchers, which help detect potential security weaknesses in automotive systems. Following subsections discuss some major cybersecurity testing approaches.

6.1 Automotive Penetration Testing

Penetration testing, in general, is a security assessment approach which is usually adopted by security testing professionals to carry out security testing from the perspective of an attacker to discover security weaknesses in a system. While there are different variants of the approach, it generally has the following key stages as outlined in the NIST Guide to Information Security Testing and Assessment [50]:

1. Planning - this phase is concerned with collecting as much information as possible about the target system as well as the boundaries and relevant components involved in the testing.
2. Discovery - in this phase, all the available public external interfaces of the system are systematically discovered and enumerated.
3. Attack - in order to test the identified interfaces, a series of attacks are launched on the system by exploiting the found vulnerabilities.
4. Reporting - the reporting takes place simultaneously with other three steps. Documentation of the findings is done in this phase. Fig. **??** illustrates the four stages of the penetration testing.

When the tester has no or limited knowledge of the system under test, they largely depend on publicly available information of the target system. In this case, the target system is treated as a black box, as such the specification of the system is not accessible. In contrast, when the tester has detailed knowledge of the system, the system can be referred to as white box, as the internal details of the system are known to the tester. Whereas, the system may be considered a grey box when the tester has partial information about it [4]. Black-box approach is the most appropriate choice for automotive cybersecurity assessment due to the unavailability of the functional specifications of in-vehicle systems.

Durrwang et al. [15] propose an improved penetration testing methodology for testing automotive systems by combining safety and threat analysis for deriving test cases in a systematic manner. The authors integrate attack trees [52] as a threat

modelling technique for deriving quality test cases. Their proposed technique is based on the Penetration Testing Execution Standard (PTES)[46], a technical guide for penetration testing, which recommends threat modeling and integrates it as a key step. The authors perform an experiment attack involving an airbag ECU to demonstrate the application of their proposed technique.

PTES [46] defines the key stages or phases of the penetration testing as follows:

1. Pre-engagement Interactions
2. Intelligence Gathering
3. Threat Modelling
4. Vulnerability Analysis
5. Exploitation
6. Post Exploitation
7. Reporting

Cheah et al. [11] propose a similar penetration testing framework for security evaluation of automotive interfaces. Very similar to the work described above, attack trees are an essential part of the approach for threat modeling. The authors apply the technique to the automotive Bluetooth interfaces for uncovering potential vulnerabilities. They also introduce a proof-of-concept tool to perform testing on vehicles using the proposed framework. The proposed proof-of-concept tool follows an attack tree for carrying out testing on the Bluetooth interface of the automobile. The tool is semi-automated i.e., some manual interventions are required for the tool to complete security assessment of the target system.

All the testbeds surveyed in this study support the penetration testing.

6.2 Automotive Fuzz Testing (Fuzzing)

Fuzz testing or fuzzing is used to discover new vulnerabilities by exposing the system to invalid, malformed, or unexpected inputs and the target system is monitored for any unusual behaviour, which may cause the system to crash. Fuzzing involves three main steps [32, 33]:

1. Preparing the input
2. Delivering the input to the target
3. Observing the behaviour of the target

Fuzzer, a software application specifically designed for performing fuzz testing, is used for bombarding the system under test with a huge number of automatically generated data values. The software then observes system's behaviour to see any reactions to the input data. The input values are crafted either from existing valid input dataets or from a prescribed set of values.

While fuzzing has been around since 90s, and widely used as an effective testing technique in other domains for vulnerability discovery, it is not very popular in automotive security testing yet. This is probably due to the presence of specific

challenges that require some adjustments for successful application of the technique to automotive security assessment. For example, monitoring of the system for unusual behaviour is crucial, but since the same interface is usually used for both the fuzz massage-injection and monitoring purposes, this means the internal reactions of the system might not be visible to the observers. A virtual testing environment with adequate support for observing the reaction of the target ECU can be an effective solution to this challenge as Bayer et al. report in [5].

In their study, Fowler et al. [21] describe a basic experiment attack they performed on a virtual vehicle using fuzz testing with a custom-built fuzzer. OBD port was used to interface the fuzzer with the CAN bus. The attack involved locking/unlocking the door lock of the virtual vehicle by injecting messages onto CAN bus. This was achieved by injecting random CAN messages for a short period of time. Based on their experience by executing the attack successfully and influencing the behaviour of the vehicle, the authors conclude that the fuzzing can be useful in reverse engineering of CAN messages as well as causing disruption to the vehicular networks. Most importantly, they note that the fuzzing can be detrimental for the vehicle under test.

In a more recent work [23], Fowler et al. emphasize the importance and usefulness of fuzzing (and other security testing methods), especially, when it is performed prior to production for allowing the discovery and fixing of bugs, which can lead to serious security issues, in the early phases of the system development.

6.3 Model-Based Security Testing

Model-based security testing is concerned with specifying, documenting and generating security test objectives, test cases, and test suites in a systematic and efficient manner [53]. It primarily uses models to verify if the target system meets its security requirements [18].

Santos et al. [49] propose their automotive cybersecurity testing framework, which uses Communication Sequential Processes (CSP) for representing the models of the vehicle's bus systems as well as a set of attacks against these systems. CSP - a language with its own syntax and semantics - is a process-algebraic formalism used to model and analyze concurrent systems. Using CSP, they create architectures of the vehicle's network and bus systems along with the attack models. One of the key challenges that authors claim to address in their work is the scalability of the testing in distributed environments.

Their system model is comprised of networks, bus systems connected to each network, and the gateways. Additionally, network parameters, such as latency can also be modelled. An attack model is also created, defining the attackers' capabilities as channels. An attacker's capabilities may include command spoofing, communication disruption, eavesdropping and influencing behavior of the system. According to the authors, the ability for a detailed definition of the scope of the attack and test cases is a key advantage of using these models for security testing.

Wasicek et al. [57] present aspect-oriented modelling (AOM) as a powerful technique for security evaluation of Cyber-Physical Systems (CPS), especially focusing on safety-critical elements in automotive control systems.AOM is based on the ideas inspired by aspect-oriented programming, which is concerned with crosscutting aspects being expressed as concerns (e.g., security, quality of service, caching etc.) [17]. Aspect-oriented modelling is used to express crosscutting concerns at a higher level of abstraction by means of modelling elements [8].

The technique presented by [57] models attacks as aspects, and aims at discovering and fixing potential security flaws and vulnerabilities at design time, because it becomes highly costly to find and fix the bugs if they are discovered later in the development life-cycle stages for automotive systems. Some of the main benefits that can be achieved by using AOM for security assessment of automotive systems include: separation of functional and attack models into aspects allows domain experts to work on different aspects without any interference; real-world attack scenarios involving high degree of risks can be modelled easily; general models can be reused in other systems.

An automotive case study is presented by the authors, involving adaptive cruise control system as an example. They use a special modeling and simulation framework, called Ptolemy II, for developing their models. The authors intended to explore effects of attacks on the communication between two vehicles. A discussion of four different attacks (i.e., man-in-the-middle, fuzzing, interruption, and replay) is presented.

6.4 Automotive Vulnerability Scanning

Automotive vulnerability scanning focuses on testing the system for existing known weaknesses in the system to ensure that the system is protected against known threats. An automotive system is typically scanned for identifying known weaknesses in the source code, ICT infrastructure and networks by using a regularly updated database of known vulnerabilities.

Vulnerability scanning can be performed in several different ways, depending on the types of target weaknesses for which the system is being examined. For example, in order to verify whether certain software flaws (e.g., buffer/heap overflows) present in the software, static and dynamic analyses can be performed on the source code. Various interfaces including WiFi, cellular network, and Ethernet can be scanned for open ports and running services in automotive systems. In particular, in-vehicle networks, such as CAN and on-board diagnostic port should be scanned. Finally, analysis of the entire system specifically focusing on various configurations to verify if there are any loopholes that can be leveraged by adversaries to compromise the system. [1].

Vulnerability scanning of an automotive infotainment system is presented in a recent study [24] by Josephlal and Adepu. The infotainment system used in the study has various connectivity interfaces including WiFi, Bluetooth, USB port,

CAN and others. The authors used different tools (e.g., Nmap, Nessus) to support their experiment involving a attack vector analysis and vulnerability scanning of the infotainment system. The scan was able to detect various types of vulnerabilities of varying levels of risks. In particular, IP address of the infotainment system, an infotainment service running on a certain port, as well as a number of information leaking vulnerabilities were identified.

In addition to the vulnerability scan described above, they also report different attacks including a denial-of-service attack they conducted using a malicious smartphone app.

7 Conclusion

Modern cars are open to various cyberattacks due to in-vehicle ICT capabilities they are equipped with. Discovery of any security weaknesses that may potentially be present in the automotive systems is a first important step towards strengthening their security. Testing real automotive systems involves safety and economic risks. One effective solution to this issue is using testing environments instead of relying on real vehicles, as it offers several benefits including a safe and cost-effective testing setup. In order to ensure that maximum number of security flaws are revealed and fixed, a systematic and suitable testing approach must be employed. There are no known studies exist providing information on cybersecurity testbeds and security methods, which can be useful for students, researchers, and security professionals in the automotive cybersecurity domain for setting up their own testing environment and use established, systematic testing methods.

This study presents a survey of seven different automotive cybersecurity testbeds and four different types of testing approaches including automotive penetration testing, automotive fuzz testing, model-based security testing and automotive vulnerability scanning. Core features, merits, limitations and various characteristics of all testbeds and testing methods have been highlighted.

References

1. Bayer, S., Enderle, T., Oka, D. K., & Wolf, M. (2016). Automotive security testing—the digital crash test. In Energy Consumption and Autonomous Driving (pp. 13-22). Springer, Cham.
2. Bayer, S., & Ptok, A. Don't Fuss about Fuzzing: Fuzzing Controllers in Vehicular Networks. In 1ESCRYPT GmbH Leopoldstraße 244 80807 München.
3. Bayer, S., Hirata, K., & Oka, D. K. (2016). Towards a Systematic Pentesting Framework for In-Vehicular CAN Networks. 14th escar Europe, 45-52.
4. Bayer, S., Enderle, T., Oka, D. K., & Wolf, M. (2015). Security crash test-practical security evaluations of automotive onboard it components. Automotive-Safety & Security 2014.
5. Bayer, S., Kreuzinger, T., Oka, D., & Wolf, M,. (2016). Successful security tests using fuzzing and HiL test systems, (2016, December) [Online]. Available:

https://www.etas.com/download-center-files/products_LABCAR_Software_Products/Hanser-automotive_Successful-security-tests-hil-system_en.pdf
6. Borazjani, P., Everett, C., McCoy, D. (2014, June). OCTANE: an extensible open source car security testbed. In Proceedings of the Embedded Security in Cars Conference (p. 60).
7. Buttigieg, R., Farrugia, M., & Meli, C. (2017, December). Security issues in controller area networks in automobiles. In 2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA) (pp. 93-98). IEEE.
8. Chavez, C., & Lucena, C. (2002, April). A metamodel for aspect-oriented modeling. In Workshop on Aspect-Oriented Modeling with UML (AOSD-2002).
9. Cheah, M., Nguyen, H. N., Bryans, J., & Shaikh, S. A. (2017, September). Formalising Systematic Security Evaluations Using Attack Trees for Automotive Applications. In IFIP International Conference on Information Security Theory and Practice (pp. 113-129). Springer, Cham.
10. Cheah, M., Shaikh, S. A., Bryans, J., & Wooderson, P. (2018). Building an automotive security assurance case using systematic security evaluations. Computers & Security, 77, 360-379.
11. Cheah, M., Shaikh, S. A., Haas, O., & Ruddle, A. (2017). Towards a systematic security evaluation of the automotive Bluetooth interface. Vehicular Communications, 9, 8-18.
12. Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., ... Kohno, T. (2011, August). Comprehensive experimental analyses of automotive attack surfaces. In USENIX Security Symposium (Vol. 4, pp. 447-462).
13. Cintuglu, M. H., Mohammed, O. A., Akkaya, K., Uluagac, A. S. (2016). A survey on smart grid cyber-physical system testbeds. IEEE Communications Surveys Tutorials, 19(1), 446-464.
14. Daily, J., Gamble, R., Moffitt, S., Raines, C., Harris, P., Miran, J., . Johnson, J. (2016). Towards a cyber assurance testbed for heavy vehicle electronic controls. SAE International Journal of Commercial Vehicles, 9(2016-01-8142), 339-349.
15. Dürrwang, J., Braun, J., Rumez, M., Kriesten, R., & Pretschner, A. (2018). Enhancement of automotive penetration testing with threat analyses results. SAE International Journal of Transportation Cybersecurity and Privacy, 1(11-01-02-0005), 91-112.
16. Eiza, M. H., & Ni, Q. (2017). Driving with sharks: Rethinking connected vehicles with vehicle cybersecurity. IEEE Vehicular Technology Magazine, 12(2), 45-51.
17. Elrad, T., Filman, R. E., & Bader, A. (2001). Aspect-oriented programming: Introduction. Communications of the ACM, 44(10), 29-32.
18. Felderer, M., Zech, P., Breu, R., Büchler, M., & Pretschner, A. (2016). Model-based security testing: a taxonomy and systematic classification. Software Testing, Verification and Reliability, 26(2), 119-148.
19. Felderer, M., Büchler, M., Johns, M., Brucker, A. D., Breu, R., & Pretschner, A. (2016). Security testing: A survey. In Advances in Computers (Vol. 101, pp. 1-51). Elsevier.
20. Fisher, K. (2012, December). HACMS: high assurance cyber military systems. In ACM SIGAda Ada Letters (Vol. 32, No. 3, pp. 51-52). ACM.
21. Fowler, D. S., Bryans, J., Shaikh, S. A., & Wooderson, P. (2018, June). Fuzz testing for automotive cyber-security. In 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W) (pp. 239-246). IEEE.
22. Fowler, D. S., Cheah, M., Shaikh, S. A., & Bryans, J. (2017, March). Towards a Testbed for Automotive Cybersecurity. In 2017 IEEE International Conference on Software Testing, Verification and Validation (ICST) (pp. 540-541). IEEE.
23. Fowler, D. S., Bryans, J., Cheah, M., Wooderson, P., & Shaikh, S. A. (2019, July). A Method for Constructing Automotive Cybersecurity Tests, a CAN Fuzz Testing Example. In 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C) (pp. 1-8). IEEE.
24. Josephlal, E. F. M., & Adepu, S. (2019, January). Vulnerability Analysis of an Automotive Infotainment System's WIFI Capability. In 2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE) (pp. 241-246). IEEE.
25. Haas, R. E., & Möller, D. P. (2017, May). Automotive connectivity, cyberattack scenarios and automotive cyber security. In 2017 IEEE International Conference on Electro Information Technology (EIT) (pp. 635-639). IEEE.

26. , Hafeez, A., Malik, H., Avatefipour, O., Rongali, P. R., & Zehra, S. (2017). Comparative study of can-bus and flexray protocols for in-vehicle communication (No. 2017-01-0017). SAE Technical Paper.
27. Holm, H., Karresand, M., Vidström, A., Westring, E. (2015, October). A survey of industrial control system testbeds. In Nordic Conference on Secure IT Systems (pp. 11-26). Springer, Cham.
28. Hoppe, T., Kiltz, S., & Dittmann, J. (2008, September). Security threats to automotive CAN networks–practical examples and selected short-term countermeasures. In International Conference on Computer Safety, Reliability, and Security (pp. 235-248). Springer, Berlin, Heidelberg.
29. Ishtiaq Roufa, R. M., Mustafaa, H., Travis Taylora, S. O., Xua, W., Gruteserb, M., Trappeb, W., & Seskarb, I. (2010, August). Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study. In 19th USENIX Security Symposium, Washington DC (pp. 11-13).
30. KASTEBO, M., & NORDH, V. Model-based Security Testing in Automotive Industry.
31. Klinedinst, D., & King, C. (2016). On board diagnostics: Risks and vulnerabilities of the connected vehicle. Software Engineering Institute-Carnegie Mellon University, 10.
32. Li, J., Zhao, B., & Zhang, C. (2018). Fuzzing: a survey. Cybersecurity, 1(1), 6.
33. Manès, V. J. M., Han, H., Han, C., Cha, S. K., Egele, M., Schwartz, E. J., & Woo, M. (2019). The art, science, and engineering of fuzzing: A survey. IEEE Transactions on Software Engineering.
34. Miller, C., & Valasek, C. (2013). Adventures in automotive networks and control units. Def Con, 21, 260-264.
35. Miller, C., & Valasek, C. (2014). A survey of remote automotive attack surfaces. black hat USA, 2014, 94.
36. Miller, C., Valasek, C. (2015). Car hacking: for poories. Technical report, IOActive Report.
37. Miller, C., & Valasek, C. (2015). Remote exploitation of an unaltered passenger vehicle. Black Hat USA, 2015, 91.
38. Mundhenk, P., Steinhorst, S., Lukasiewycz, M., Fahmy, S. A., & Chakraborty, S. (2015, June). Security analysis of automotive architectures using probabilistic model checking. In Proceedings of the 52nd Annual Design Automation Conference (p. 38). ACM.
39. Munera, J., Fuentes, J. M. D., & González-Tablas, A. I. (2011). Towards a comparable evaluation for VANET protocols: NS-2 experiments builder assistant and extensible test bed.
40. Nazir, S., Patel, S., & Patel, D. (2017). Assessing and augmenting SCADA cyber security: A survey of techniques. Computers & Security, 70, 436-454.
41. Nilsson, D. K., & Larson, U. E. (2008, August). Simulated attacks on can buses: vehicle virus. In IASTED International conference on communication systems and networks (AsiaCSN) (pp. 66-72).
42. Oka, D. K., Fujikura, T., & Kurachi, R. (2018). Shift Left: Fuzzing Earlier in the Automotive Software Development Lifecycle using HIL Systems.
43. Oruganti, P. S., Appel, M., Ahmed, Q. (2019, March). Hardware-in-loop based Automotive Embedded Systems Cybersecurity Evaluation Testbed. In Proceedings of the ACM Workshop on Automotive Cybersecurity (pp. 41-44). ACM.
44. Kim, H. Y., Choi, Y. H., & Chung, T. M. (2012, February). Rees: Malicious software detection framework for meego-in vehicle infotainment. In 2012 14th International Conference on Advanced Communication Technology (ICACT) (pp. 434-438). IEEE.
45. Patel, Shwetak and Kohno, Tadayoshi and Checkoway, Stephen and McCoy, Damon and Kantor, Brian and Anderson, Danny and Shacham, Hovav and others: Experimental Security Analysis of a Modern Automobile. 2010 IEEE Symposium on Security and Privacy.
46. Penetration Testing Execution Standard, PTES Technical Guidelines, 2014.
47. Riggs, C., Rigaud, C. E., Beard, R., Douglas, T., & Elish, K. (2018). A Survey on Connected Vehicles Vulnerabilities and Countermeasures. Journal of Traffic and Logistics Engineering Vol, 6(1).
48. Rizvi, S., Willet, J., Perino, D., Marasco, S., & Condo, C. (2017). A threat to vehicular cyber security and the urgency for correction. Procedia computer science, 114, 100-105.

49. Santos, E. D., Simpson, A., & Schoop, D. (2018). A formal model to facilitate security testing in modern automotive systems. arXiv preprint arXiv:1805.05520.
50. Scarfone, K., Souppaya, M., Cody, A., Orebaugh, A. (2008). Technical guide to information security testing and assessment. NIST Special Publication, 800(115), 2-25.)
51. Salfer, M., and Eckert, C. (2015, July). Attack surface and vulnerability assessment of automotive Electronic Control Units. In 2015 12th International Joint Conference on e-Business and Telecommunications (ICETE) (Vol. 4, pp. 317-326). IEEE.
52. Schneier, B. (1999). Attack trees. Dr. Dobb's journal, 24(12), 21-29.
53. Schieferdecker, I., Grossmann, J., & Schneider, M. (2012). Model-based security testing. arXiv preprint arXiv:1202.6118.
54. Smith, C. (2016). The car hacker's handbook: a guide for the penetration tester. No Starch Press.
55. Studnia, I., Nicomette, V., Alata, E., Deswarte, Y., Kaâniche, M., & Laarouchi, Y. (2013, June). Survey on security threats and protection mechanisms in embedded automotive networks. In 2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W) (pp. 1-12). IEEE.
56. Toyama, T., Yoshida, T., Oguma, H., Matsumoto, T. PASTA: Portable Automotive Security Testbed with Adaptability.
57. Wasicek, A., Derler, P., & Lee, E. A. (2014, June). Aspect-oriented modeling of attacks in automotive cyber-physical systems. In 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC) (pp. 1-6). IEEE.
58. Yan, W. (2015, October). A two-year survey on security challenges in automotive threat landscape. In 2015 International Conference on Connected Vehicles and Expo (ICCVE) (pp. 185-189). IEEE.
59. X. Zheng, L. Pan, H. Chen, R. D. Pietro and L. Batten, A Testbed for Security Analysis of Modern Vehicle Systems, 2017 IEEE Trustcom/BigDataSE/ICESS, Sydney, NSW, 2017, pp. 1090-1095. doi: 10.1109/Trustcom/BigDataSE/ICESS.2017.357