Symbiotic Analysis of Security Assessment and Penetration Tests Guiding Real L4 Automated City Shuttles

Benyahya, Meriem; Bergerat, Pierre; Collen, Anastasija; Nijdam, Niels Alexander

How to cite

BENYAHYA, Meriem et al. Symbiotic Analysis of Security Assessment and Penetration Tests Guiding Real L4 Automated City Shuttles. In: Telecom, 2023, vol. 4, n° 1, p. 198–218. doi: 10.3390/telecom4010012

This publication URL: https://archive-ouverte.unige.ch//unige:171906

Publication DOI: 10.3390/telecom4010012

telecom

MDPI

Article

Symbiotic Analysis of Security Assessment and Penetration Tests Guiding Real L4 Automated City Shuttles

Meriem Benyahya, Pierre Bergerat, Anastasija Collen * and Niels Alexander Nijdam

Centre Universitaire d'Informatique, Geneva School of Economics and Management, University of Geneva, Route de Drize 7, CH-1227 Carouge, Switzerland

* Correspondence: anastasija.collen@unige.ch

Abstract: The Connected Automated Vehicle (CAV)'s deployment is proof of the wide evolution of autonomous driving technologies enabling vehicles to gradually dispose of their drivers. Within the scope of smart cities, such innovation has given rise to a new type of CAV: the Automated City Shuttle (ACS). Foreseen as the new paradigm aiming to shape the public transport model, the ACS elicits a plurality of new applications, such as the on-demand service in which a driverless shuttle offers the desired ride without human intervention. However, such a model raises cybersecurity concerns through the numerous attack surfaces and vehicle hyperconnection. This phenomenon was highlighted in several studies on CAVs, but very few research works tackled the specific case of ACSs, whose challenges and risks far exceed those of personal vehicles. The present work offers a comprehensive investigation of cybersecurity attacks, demonstrates a performed risk assessment based on the ISO/SAE 21434 standard, and showcases a penetration test over a real ACS of automation level four (L4) according to the Society of Automotive Engineering (SAE)'s ranking. Based on our experiments, we leverage fundamental cybersecurity recommendations with a focus on the ACS's physical security.

Keywords: automated city shuttles; connected automated vehicles; cybersecurity; ISO/SAE 21434; penetration testing; risk analysis

Citation: Benyahya, M.; Bergerat, P.; Collen, A.; Nijdam, N.A. Symbiotic Analysis of Security Assessment and Penetration Tests Guiding Real L4 Automated City Shuttles. *Telecom* **2023**, *4*, 198–218. https://doi.org/10.3390/telecom4010012

Academic Editor: Philip Branch

Received: 15 February 2023
Revised: 6 March 2023
Accepted: 10 March 2023
Published: 20 March 2023

1. Introduction

Connected Automated Vehicles (CAVs) are motorised vehicles with embedded technologies aiming to assist and handle the driving functionality on behalf of drivers. In recent decades, the CAV industry has been increasing annually by 16% at a global scale [1]. Such a market aims to generate $300 to $400 billion by 2035 [2] with a market share of 20–35% of new vehicles by 2030 [3] and even up to 50% by 2050 [4]. Along with the ambitious forecasts and the related economic growth, the Society of Automotive Engineering (SAE) defined six levels of automation, ranging from no automation at L0 to full automation of driving at L5, in which each level gradually assists the driving performance [5]. CAVs of L4 denote a level of automation capable of conducting all driving functions under certain conditions, while L5 vehicles can fully perform automated driving under any condition. Such automation is accomplished through sensors, Electronic Control Units (ECUs) and Artificial Intelligence (AI) units [6]. Not limited to internal components, CAVs rely also on numerous communications with external entities to accomplish autonomous driving functions, namely Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Everything (V2X) [7]. In the present paper, we focus our research on exploring the Automated City Shuttle (ACS) as a sub-class of CAV, suitable for coping with today's public transportation needs [8].

ACSs are foreseen as the next generation of smart mobility for public transportation, offering on-demand services tailored for citizens. Putting into perspective the simplicity of ordering a shuttle service while preserving the high quality of transportation, ACSs

are also shown to be safer [9], to reduce traffic congestion, and to decrease pollution in comparison to conventional vehicles [10]. More specifically, ACSs are well suited for the transportation of the elderly and people with disabilities or reduced mobility [11]. Therefore, the wide deployment of ACSs could be a paradigm shift in achieving a cheap, reliable, always available, and accessible new way of transport for smart cities. Driven by these advantages, several cities throughout the world have already started testing ACSs in their fleets in multiple pilot projects [12]. However, such technologies introduce multiple security concerns threatening the passengers' safety and security as demonstrated by several researchers. To illustrate, Bec et al. [13] reported attacks on the Chevrolet Camaro; Miller and Valasek [14] implemented a remote takeover of the braking and steering systems of a Jeep Cherokee; and Yan et al. [15] demonstrated a blinding attack over the Tesla S sensors leading the vehicle to crash.

For an in-depth understanding of the ACS threats, we had the opportunity, under the umbrella of the AVENUE project [16], to investigate, analyse and conduct penetration testing over the L4 vehicle depicted in Figure 1. Our study relies on the Threat Analysis and Risk Assessment (TARA) methodology provided by the standard ISO/SAE 21434 "Road vehicles—Cybersecurity engineering" [17]. The methodology supports with building threat scenarios, rating the attacks' impact, and determining the risk related to the ACS's assets. We then elevate the TARA's findings further by performing penetration tests (hereinafter, referred to as pentests) over the vehicle's Global Navigation Satellite System (GNSS) and 4G connections. From the obtained results, we provide our recommendations to mitigate the risks and the identified weaknesses.

Figure 1. Example of the Automated City Shuttle investigated in the present work.

This paper aims to answer the following research questions:

RQ 1. *Is the TARA methodology suitable for identifying L4 specific threats?*

RQ 2. *Would the execution of penetration tests confirm the resilience of the mitigations applied to the high-risk scenarios defined by the TARA?*

The remainder of this paper is structured as follows: Section 2 discusses related works and identifies knowledge gaps in the domain. Section 3 provides background information on the materials used to perform the TARA approach on the L4 vehicle and describes the implementation methodology of the pentests. Section 4 discusses the obtained results,

while Section 5 debates the research questions and outlines our recommendations. Finally, Section 6 offers concluding remarks on our findings.

2. Related Work

With the pervasive technologies leading to ACS deployment in smart cities and their associated cybersecurity challenges, the ISO/SAE 21434 [17] is considered the prominent standard for automotive cybersecurity governance. This standard, as well as the mandatory United Nations Economic Commission for Europe (UNECE) R155 regulation [18], introduced the TARA and security testing as an efficient way to keep systems at an acceptable level of risk. Therefore, we highlight in the present research three main avenues, namely the cybersecurity challenges within the ACS ecosystem, security assessments based on the TARA from ISO/SAE 21434, and automotive pentests.

2.1. Cyber Threats in the ACS Landscape

While there are extensive efforts to identify the cybersecurity challenges within the CAV ecosystem, the research domain tackling ACSs specifically is only just emerging. Fysarakis et al. [19] spotlighted their concerns about the concept of ACSs and proposed a threat model as well as generic mitigation solutions for CAVs at a general scope. More focused on ACSs, Marin-Plaza et al. [20] offered a comprehensive analysis of the ACSs deployment, signalled about the cybersecurity risks, and discussed their social implications within modern cities. However, the review was conducted from the social science perspective without a thorough cybersecurity analysis. An in-depth research study was conducted by Benyahya et al. [6] in which a holistic state of the art of the ACSs cybersecurity and data privacy threats were provided. The authors also presented a review of relevant mitigation strategies and regulations to consider within the ACSs environment. Nonetheless, concrete and non-theoretical cyber attack implementations on ACSs are still lacking.

2.2. Assessments Based on the TARA from ISO/SAE 21434

Although several research works have provided reviews on risk assessment approaches, a limited number of researchers has showcased methods compliant with ISO/SAE 21434 on highly automated vehicles. Islam et al. [21] conducted a threat modelling and risk assessment on the vehicle speed limiter (the unit that supports a driver to not exceed a set speed limit).Wang et al. [22] performed a risk assessment on the vehicle T-Box (which is responsible for the automotive remote-control functions, such as contactless door opening). Both publications presented systematic risk assessment frameworks; however, the proposed models do not align with recent standards. More compliant approaches to the trending ISO/SAE 21434 were proposed by Lautenbach et al. [23] and Vogt et al. [24]; however, they are limited to conventional vehicles without targeting either CAVs or ACSs assets.

2.3. Automotive Pentests

Motivated by testing how robust vehicles are from cyber attacks, several researchers simulated attacks on isolated CAVs' components while very few asserted pentests over real vehicles. Cao et al. [25] mimicked physical removal attacks on a Light Detection and Ranging (LiDAR) sensor aiming to deceive the obstacle-detecting system [25]. Petit et al. [26] conducted jamming and spoofing attacks on isolated LiDARs and cameras under lab conditions [26]. As real pentests, Andersson [27] performed a grey-box pentest (in which the pentester has partial knowledge of the target vulnerabilities) on the in-vehicle infotainment system of a conventional Volvo car. Similarly, Moukahal et al. [28] conducted grey-box tests, only virtually, on the vehicular software system using OpenPilot, an automated driving simulator [29]. Fowler et al. [30] conducted a black-box test (in which the pentesters have no idea about the target vulnerabilities) on a Controller Area Network (CAN) testbed. Unfortunately, most of these works had neither a real highly automated vehicle of SAE L4 or L5 nor combined multiple pentests over several surface attacks.

To that end, our contribution differs from the aforementioned by:

- Exploring the cybersecurity concerns of the ACS as a barely studied CAV model;
- Conducting the TARA method, which is compliant to ISO/SAE 21434 standard;
- Yielding real pentests over a highly automated vehicle of SAE L4.

3. Material & Methods

This section describes the methodological approach followed, which is also depicted in Figure 2.

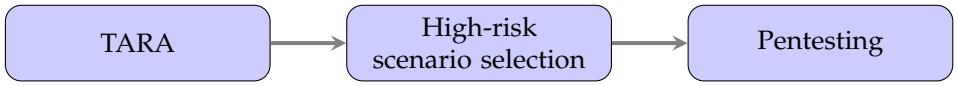

Figure 2. Methodology.

3.1. L4 Evaluation Vehicle

To demonstrate the TARA methodology, we analysed an ACS of automation L4, annotated throughout this paper as *L4 Evaluation Vehicle (L4V)*. The selected vehicle was used for testing highly automated driving and on-demand services for public transport on a pilot site in Geneva (Switzerland). The vehicle has a capacity of 15 passengers and drives at an average speed of 18 km/h within a predefined region of 38 hectares. Thanks to its several sensors, which include cameras, GPS, RADAR, LiDAR, and odometers, the vehicle is capable of autonomously building a picture of its surroundings, recognising obstacles, and bypassing them [31]. However, due to legal obligations, a safety operator remains required to intervene if needed, which makes it an L5 instead of an L5 vehicle.

3.2. Risk Analysis

We have performed the risk analysis of the *L4V* with the TARA framework provided by the standard ISO/SAE 21434. The TARA permits high-level technology agnostic risk analysis with a focus on the vehicle itself, instead of surrounding components, such as V2I/V2X or any of the backend infrastructures used by the system. The TARA includes six successive steps, depicted in Figure 3, in which each step relies on the findings of the preceding step. In the following sections, we describe each step that we followed and present a condensed version of our findings.

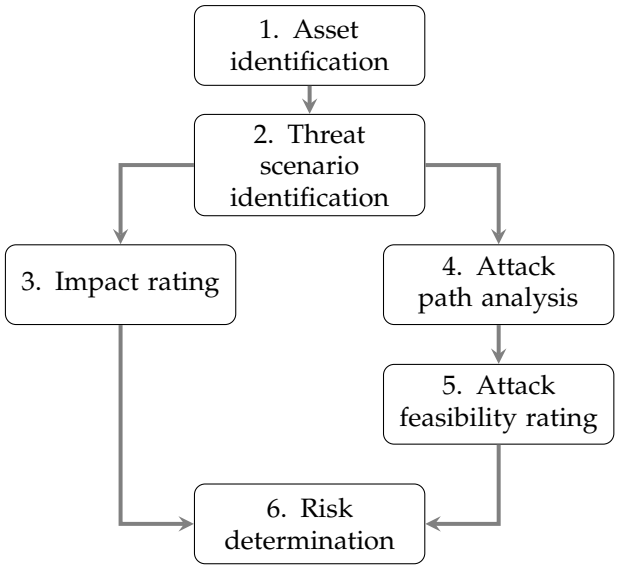

Figure 3. TARA steps provided by ISO/SAE 21434.

3.2.1. Asset Identification

As the name suggests, this step is dedicated to the identification of valuable assets, which must be protected from potential damage. *L4V* was identified as having seven key

assets: a 3G/4G antenna, a GNSS antenna, a 3D LiDAR, an odometer, cameras, and an on-board computer. These assets are considered to be the main entry points for an attacker and constitute every component the vehicle uses to drive autonomously. As such, any alteration to any of those components can lead to safety issues and consecutive damages. The completeness of this first step is essential as it forms the basis for determining the potential threats to the system and evaluating the likelihood and impact of those threats. It should be noted that most of these components, on the vehicle, are directly exposed to the outside environment and thus are prone to physical attacks.

3.2.2. Threat Scenario Identification

Each of the assets identified in the previous step needs to be further analysed for possible damage scenarios, leading to compromise of the cybersecurity triad Confidentiality, Integrity, Availability (CIA). Using the Spoofing, Tampering, Repudiation, Information disclosure, Denial-of-service and Elevation of privilege (STRIDE) threat modelling framework [32], we found 27 scenarios in total. A sample outline of the threat scenarios is shown in Table 1.

Table 1. Sample threat scenarios for 3G/4G antenna.

Asset ID	Damage Scenario ID	Description
A.1	D.1	Erroneous data are received and provoke full stop of the vehicle
	D.2	The data cannot be received and provoke full stop of the vehicle
	D.3	An external attacker modifies transmitted data or an update
	D.4	An external attacker captures the data transmitted between vehicle and the backend
	D.5	An external attacker modifies the data transmitted between vehicle and the backend
	D.6	An external attacker stops the communication between vehicle and the backend

3.2.3. Impact Rating

The next step implies the value estimation of a potential damage scenario, performed through qualitative conversion tables provided by the TARA. It permits the assignment of a label to each scenario ranging between "Negligible" and "Severe" based on the scenario's impact on Safety (**S**), Financial (**F**), Operational (**O**), and users' Privacy (**P**). These criteria are then aggregated to obtain the "Impact Level" (**IL**), also ranging between "Negligible" and "Severe". To that end, such rankings allow us to prioritise both the economical and human repercussions to consider in order to adequately mitigate the risks based on the severity of the scenarios. An example of such a rating is provided in Table 2 in which severe and a major damage scenarios are depicted.

Table 2. Impact rating example for damage scenarios applicable to 3G/4G antenna.

Damage Scenario ID	Impact Category				Impact Level	Justification
	S	F	O	P		
D.3	Severe	Severe	Severe	Severe	Severe	If the vehicle's software stack is modified, all data can become accessible with a risk of compromising secure driving functions such as braking, maximum speed limit, and respect of signal panels. Serious financial consequences are forecasted, as well as the loss of end-users' trust.
D.5	Severe	Severe	Severe	Negligible	Major	Active modification of ongoing communications can cause an unexpected behaviour of the vehicle or generate erroneous data for the operator.

3.2.4. Attack Path Analysis

The fourth step is designated for the synthesis of the possible implementation of damage scenarios. The resulting attack paths are a sequence of actions needed to execute an attack, as illustrated in Figure 4. To establish valid attack paths, one can use previous analyses from known vulnerabilities, such as the Common Vulnerabilities and Exposure (CVE) databases [33], vulnerability classifications, or taxonomies as per Sommer et al. [34]'s attack categorisation. The analysis can be built on a parent–child representation afterwards to meet the ISO/SAE 21434 recommendations. An example of the attack path analysis result for D.3 is demonstrated in Table 3 in which every path leading to the parent node from Figure 4 demonstrates an attack path.

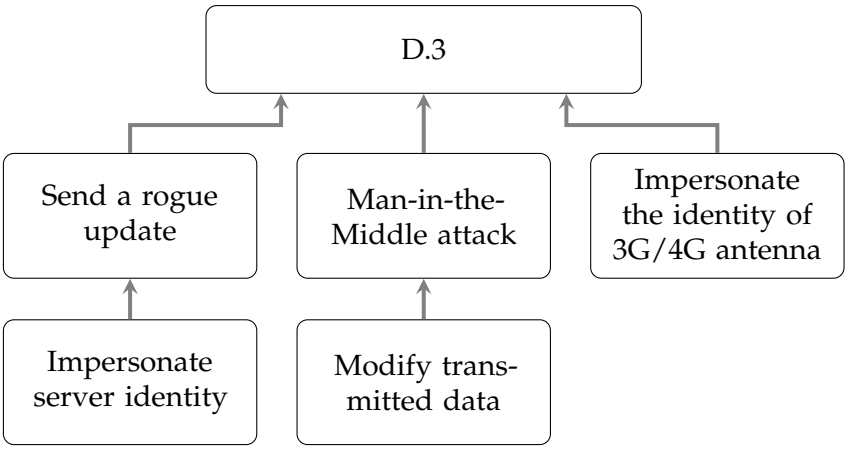

Figure 4. Attack tree showing three attack paths, each from lowest child to root.

Table 3. Sample attack path scenarios for damage scenario where attacker modifies transmitted data.

Damage Scenario ID	Attack Path Scenario ID	Attack Path Description
D3	AP.3	An attacker can impersonate the server identity to send a rogue update, thereby compromising the integrity of the legitimate data.
	AP.4	An attacker can execute a Man-in-the-Middle attack to modify transmitted data, compromising, as a result, the integrity of the legitimate data.
	AP.5	An attacker can impersonate the identity of a 3G/4G antenna and send falsified data, compromising, as a result, the integrity of the legitimate data.

3.2.5. Attack Feasibility Rating

The fifth step of the framework conducts a rating of an attack path's feasibility. This rating is based on the following criteria, listed below, in which each criterion is split into different possible ranges. Those ranges are then converted into a quantitative value and summed up to obtain the Aggregated Attack Feasibility Level (AAFL), as shown in Table 4. This rating represents the overall feasibility of the attack based on each of the criteria that composes it:

- Elapsed time: how much time the attack execution requires (1 week/1 month/6 months/3 years/more than 3 years);
- Expertise: skill and experience required to execute the attack, as well as how many people are needed (Layman/Proficient/Expert/Multiple experts);

- Equipment: availability of the tools needed to perform the attack (Standard/Specialised/Bespoke/Multiple Bespoke);
- Knowledge of the item or component: how much information is needed to perform the attack (Public information/Restricted information/Confidential information/Strictly confidential information);
- Window of opportunity: ease of access and time limitation (Unlimited/Easy/Moderate/Difficult).

Table 4. AAFL rating criteria.

Attack Feasibility	Sum
High	0–13
Medium	14–19
Low	20–24
Very low	≥25

An illustration of the previously outlined attack paths and their feasibility ratings is provided in Table 5.

Table 5. Sample of attack feasibility rating for damage scenario in which attacker modifies transmitted data.

Attack Path Scenario ID	Time	Expertise	Knowledge	Window Opportunity	Equipment	Value	Attack Feasibility
AP.3	1	6	7	4	0	18	Medium
AP.4	0	3	3	1	4	11	High
AP.5	0	3	3	1	4	11	High

3.2.6. Risk Determination

The final step of the TARA implies the determination of the associated risk value for each damage scenario by using a risk matrix (Table 6). The sample output is depicted in Table 7.

Table 6. Risk matrix scale used to obtain the final risk determination.

Impact/Attack Feasibility	Very Low	Low	Medium	High
Severe	1	3	4	5
Major	1	2	3	4
Moderate	1	2	2	3
Negligible	1	1	1	1

Table 7. Final risk determination related to D.3.

Damage Scenario ID	Attack Path Scenario ID	AAFL	Impact Level	Risk Value
D.3	AP.3	Medium	Severe	4
D.3	AP.4	High	Severe	5
D.3	AP.5	High	Severe	5

3.3. Pentesting

The performed risk analysis is permitted to identify multiple scenarios implying high attack feasibility levels and high impact, as demonstrated in Table 8. Four pentest scenarios were chosen, namely AP.6, AP.11, AP.13, and AP.14, for pentest execution based on several criteria. First, the low cost and accessibility of the necessary hardware were given the highest priority as the vehicles are operating in public spaces. Second, the attacker can easily stay out of sight and has no need to physically interact with the vehicle. Finally, these scenarios can be carried out by 'script kiddies' since the software tools and documentation needed are easily accessible on the internet via open-source programs. This is why our focus was given to these wireless attack scenarios.

The pentest period was allocated outside the operating hours of the vehicles, without them being in motion, and took place at a restricted site from the public transport operator. The different attacks were carried out in a black-box environment, which is the real environment in which an external attacker could operate. The test equipment was therefore deliberately limited so as not to require hardware that was too heavy and/or too expensive. We also assume that the attacker has limited time and access to the vehicle and that no logging or system configuration information is available. The only information used to carry out the attacks is the information freely available on the internet and on the manufacturer's website.

3.3.1. Equipment and Tools

Software Defined Radio (SDR) technologies have become mainstream. These consist of radio communication systems in which components that have been traditionally implemented in hardware (e.g., mixers, filters, amplifiers, modulators/demodulators, detectors, etc.) are instead implemented using software on a computer. This allows for more flexibility in the design of the radio system and the ability to easily change its functionality. SDRs are used in a variety of applications, including wireless communication, navigation, and radio astronomy. In recent years, many new SDRs have been produced, the most well-known being HackRf, Ubertooth, or BladeRF, which we used (see Figure 5). The model we chose (BladeRFx40) cost us CHF 520 (≈USD 565) with two quad band antennas and was able to perform all of the attacks that we implemented. To use this equipment efficiently, we also used multiple tools, listed hereafter:

- BladeRF-cli [35]: tool required to program the BladeRF.
- GNU radio [36]: widely used open-source SDR software.
- GPS Test [37]: GNSS app for phone and tablet.
- Gps-sdr-sim [38]: generates custom GPS data streams.
- Gqrx [39]: radio waves visualization tool.
- RfCat [40]: Python library for easier programming of the BladeRF.
- Ubuntu [41]: main operating system.
- YateBTS [42]: allows the creation of one's own GSM base station.
- Wireshark [43]: open-source packet analyser.

Table 8. Risk determination.

Asset	Damage Scenario	Attack Path	*An attacker could …*	† C	I	A	AAFL	IL	RV	Risk Treatment
A.1	D.1	AP.1	retransmit past data using an SDR transmitter so that the vehicle receives erroneous data	✗	✓	✓	High	Moderate	3	Integrity controls
A.1	D.2	AP.2	use an SDR transmitter or a more conventional jammer to prevent the vehicle from connecting to the network antennas	✗	✗	✓	High	Moderate	3	Offline automated mode
A.1	D.3	AP.3	impersonate the backend server in order to send a rogue update to the vehicle	✗	✓	✗	Medium	Severe	4	Integrity controls Authentication Cryptography
A.1	D.3	AP.4	perform a Man-In-The-Middle attack between the vehicle and the backend server to modify the data sent by the server	✓	✓	✗	High	Severe	5	Integrity controls Authentication Cryptography
A.1	D.3	AP.5	impersonate a 3G/4G antenna and send data to the vehicle	✗	✓	✗	High	Severe	5	Integrity controls Authentication Cryptography
A.1	D.4	AP.6	perform a Man-In-The-Middle attack between the vehicle and the backend server to listen to the data sent by the server	✓	✓	✗	High	Moderate	3	Cryptography Authentication
A.1	D.4	AP.7	perform an auxiliary channel attack by "listening" to the electromagnetic emanations of the on-board computer	✓	✗	✗	Low	Moderate	2	Side channel attacks mitigations
A.1	D.5	AP.8	impersonate the backend server in order to transmit arbitrary data	✗	✓	✓	High	Major	4	Cryptography Authentication
A.1	D.5	AP.9	perform a Man-In-The-Middle attack between the vehicle and the backend server to modify the data in transit	✓	✓	✗	High	Major	4	Cryptography Authentication
A.1	D.5	AP.10	impersonate a 3G/4G antenna and send data to the vehicle	✗	✓	✓	Medium	Major	3	Cryptography Authentication
A.1	D.6	AP.11	use an SDR transmitter or a more conventional jammer to prevent the vehicle from connecting to the network antennas	✗	✗	✓	High	Moderate	3	Offline automated mode
A.2	D.7	AP.12	use an SDR transmitter to replay previously received signals in place of the actual signals	✗	✓	✗	High	Moderate	3	Data timestamping
A.2	D.8	AP.13	use an SDR transmitter to play custom signals instead of real GNSS signals	✗	✓	✗	High	Moderate	3	Military GPS technologies
A.2	D.9	AP.14	use an SDR transmitter or a more conventional jammer to prevent the vehicle from connecting to the GNSS	✗	✗	✓	High	Negligible	1	Offline automated mode
A.3	D.10	AP.15	throw an object or hit the camera to damage it	✗	✗	✓	High	Moderate	3	Camera shielding
A.3	D.11	AP.16	throw a sticky object or other obscuring material (e.g., paint) at the camera	✗	✗	✓	High	Moderate	3	Camera shielding Hydrophobic material
A.3	D.12	AP.17	use an acoustic device to disrupt the vehicle's in-built image processing software	✗	✓	✗	Low	Moderate	2	Phonic isolation
A.4	D.13	AP.17	disrupt a gyroscope with sound, causing the vehicle to change speed due to false information about climbing or descending	✗	✓	✗	Low	Negligible	1	Phonic isolation

Table 8. *Cont.*

Asset	Damage Scenario	Attack Path	*An attacker could…*	† C	I	A	AAFL	IL	RV	Risk Treatment
A.5	D.14	AP.18	use lasers to disrupt the operation of the LiDARs and cause the vehicle to stop	✗	✗	✓	High	Moderate	3	Faster LiDAR tick rate Photochromic lens
A.5	D.15	AP.19	throw a sticky object or other obscuring material (e.g., paint) at a LiDAR	✗	✗	✓	High	Moderate	3	LiDAR shielding Hydrophobic material
A.5	D.16	AP.20	throw an object or hit a LiDAR to damage it	✗	✗	✓	High	Moderate	3	LiDAR shielding
A.6	D.17	AP.21	use an acoustic device to distort the vehicle's speed measurement, which could cause it to speed up or slow down	✗	✓	✗	Low	Moderate	2	Phonic isolation
A.7	D.18	AP.22	perform an auxiliary channel attack by "listening" to the electromagnetic emanations emitted by the on-board computer	✓	✗	✗	Low	Moderate	2	Random CPU noise
A.7	D.18	AP.23	use direct access to the on-board computer to read the computer's memory continuously	✓	✗	✗	High	Moderate	3	Group policies Computer tray shielding
A.7	D.19	AP.24	use the keyboard provided in the vehicle to exit the navya program and install other programs	✓	✓	✓	High	Severe	5	Remove keyboard Computer tray shielding USB port security
A.7	D.19	AP.25	disconnect the hard drive from the on-board computer and plug in another one	✗	✓	✗	High	Severe	5	Alarm system Computer tray shielding
A.7	D.19	AP.26	use a live USB to bypass boot passwords and modify disk contents	✗	✓	✗	High	Severe	5	Bitlocker Secure boot BIOS/CMOS password USB port security
A.7	D.20	AP.27	use a live USB to bypass boot passwords and modify disk contents	✓	✗	✓	High	Moderate	3	Bitlocker Secure boot BIOS/CMOS password USB port security
A.7	D.20	AP.28	use the keyboard provided in the vehicle to exit the navya program and observe the contents of the disk	✓	✗	✓	High	Moderate	3	Service account Group Policies Computer tray shielding
A.7	D.20	AP.29	disconnect the hard drive from the onboard computer and read it on his own device	✓	✗	✓	High	Moderate	3	Bitlocker Computer tray shielding
A.7	D.21	AP.30	use the keyboard provided in the vehicle to turn off the on-board computer	✗	✗	✓	High	Major	4	Computer tray shielding
A.7	D.21	AP.31	physically damage the on-board computer	✗	✗	✓	High	Major	4	Computer tray shielding
A.7	D.21	AP.32	use the I/O button to turn off the on-board computer	✗	✗	✓	High	Major	4	Computer tray shielding
A.7	D.21	AP.33	disconnect the on-board computer	✗	✗	✓	High	Major	4	Computer tray shielding
A.7	D.21	AP.34	could install malware on the on-board computer	✗	✗	✓	High	Major	4	Computer tray shielding USB port security

† Confidentiality, Integrity, Availability (CIA); Aggregated Attack Feasibility Level (AAFL); Impact Level (IL); Risk Value (RV).

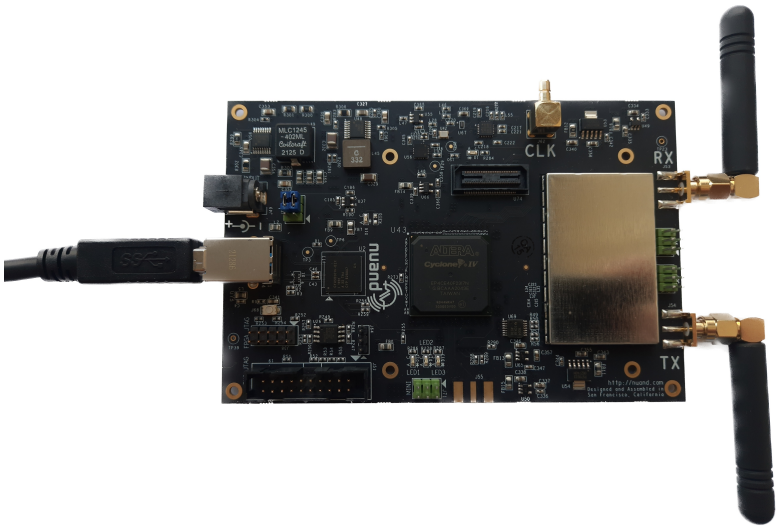

Figure 5. SDR BladeRFx40 used for our experiments.

To test whether the attacks were functioning, we also used an Android phone and an Apple tablet as references. In the next sections, we show how we used those tools to perform four attacks on the *L4V*, including GNSS spoofing, GNSS jamming, rogue Base Transceiver Station (BTS), and downgrade attacks.

3.3.2. GNSS Spoofing

Differential Global Positioning Systems (GPS), which are a widely used GNSS, provide positioning, navigation, and time services to ACSs [44]. Accurate GPS positioning data are one of the critical inputs enabling safe self-driving, yet such technology has been potentially concerned with cyber attacks such as spoofing and jamming [45]. In general, spoofing is a falsified successful identification. In the case of GPS/GNSS spoofing, a radio wave transceiver is used to broadcast false signals to a GPS/GNSS receiver, which will then determine a false position. Indeed, there is no authentication method for a GNSS signal, and it can be created without much difficulty since it contains only three types of information:

- A measurement signal for position, speed, and timing.
- The ephemeris, which contains the precise positioning information of a single satellite and which has a maximum lifetime of 4 h. Each satellite broadcasts only its own ephemeris. It is sufficient for the receiver to know the position of four satellites to propose a position [46].
- The almanack, which contains less precise information from all the satellites as well as predictions of atmospheric conditions that could change the travel time or direction of the signal. Each satellite broadcasts the almanack for all satellites. It allows the receiver to obtain data on the position of all satellites by reading only one almanack [47].

Using the published ephemeris data available on the National Aeronautics and Space Administration (NASA) website [48], it is possible to create new fictitious positions by modifying them to match the data that would actually be received if the receiver is at the simulated position. Because of its proximity to the receiver, the generated signal will be preferred to legitimate GNSS signals and will therefore modify the position announced by the receiver. This process can be used in a recreational fashion to cheat in some games that award points/bonuses based on GPS position or distance travelled, but it can also be used by attackers to disrupt the trajectory of an automated system, such as drones or CAVs. Such attacks have already been observed in Switzerland on private and commercial aircraft as well as on drones [49].

To execute a controlled GNSS spoofing attack, GNSS signals based on three positions (actual vehicle position, vehicle position offset by 4 metres and Geneva water jet) and two configurations (cold start vehicle, i.e., from a switched-off vehicle without a GNSS connection and vehicle already connected to GNSS) were transmitted using gps-sdr-sim and BladeRFx40. To accomplish this, the ephemeris was first downloaded from the NASA servers before being decompressed and used as a data source for gps-sdr-sim (see Figure 6). The data thus created is exported in a bitstream and then read by the BladeRF-cli program thanks to the code shown in Figure 7. This one sets the frequency with which the information is transmitted (1575.42 MHz) and broadcasts the data provided by gps-sdr-sim (simulation.bin). Once the program was launched, its correct operation was tested using an Android phone and an iPad to check that the spoofing was functional. For each of the tests, the two mobile devices were consistently able to lock onto the simulated position in less than 30 s, with a claimed accuracy of ±4 m.

```
wget --no-check-certificate "ftps://gdc.cddis.eosdis.nasa.gov/gnss/data/
    ↪ daily/$(date -u +%Y)/brdc/brdc$(date -u +%j0.%g)n.gz"
gzip -d brdc$(date -u +%j0.%g)n.gz
mv brdc$(date -u +%j0.%g)n ephemeris
./gps-sdr-sim -e ephemeris -l [longitude,latitude,altitude] -d [
    ↪ simulation_length] -o simulation.bin
```

Figure 6. Script used to obtain ephemerises and create the gps-sdr-sim bitstream.

```
set frequency tx 1575.42M
set samplerate 2.6M
set bandwidth 2.5M
set gain tx 32
tx config file=simulation.bin format=bin
tx start
tx wait
```

Figure 7. Script used to spoof the GNSS positionning.

3.3.3. GNSS Jamming

A radio jamming attack aims to completely cut off radio communications between two points by sending powerful radio waves (noise) on the same frequencies as those used by the targeted system [44]. Thus, a jammer could target Wi-Fi, telephone communications, or RADAR, depending on the chosen frequency. Similar to GNSS spoofing attacks, jamming attacks have become more common with the advent of smaller, inexpensive solutions that can be easily set up and hidden in a bag or mounted on a wall. It should be noted that jammers are prohibited in Switzerland and more generally in Europe, from their use to their mere possession [50]. These strict measures are intended to prevent any blockage of radio waves, which are used by emergency services and aviation, among others. However, SDR devices are not subject to such restrictions, since their use as jammers is not their primary function. Thus, despite the law of 1st of January 2018 banning the import of conventional jammers, these SDR devices can be easily obtained. A BladeRF-type device cost CHF 500 (≈USD 545.2) at the time of writing, compared to several thousand francs for a conventional jammer.

The jamming attack was performed using RfCat (see Figure 8) in order to create noise on the desired radio frequency. This tool was used as it allows easy scripting to customise the operations of SDR platforms, whether for recording, replaying, or creating signals, as is the case here. As we already know which frequency to jam, this one is simply stored in a constant (JAMMING_FREQUENCY_IN_HZ), making this script a point jammer. If needed, it would also be possible to add in an incremental loop in order to make it a sweep jammer.

Running the script resulted in a successful loss of position on both the Android phone (in "GPS only" mode) and the iPad.

```
from rflib import *

JAMMING_FREQUENCY_IN_HZ = 1575420000
_rfCat = RfCat()
_rfCat.setMdmModulation(0x30)
_rfCat.setMdmSyncMode(0)
_rfCat.setMdMRate(4800)
_rfCat.setFreq(JAMMING_FREQUENCY_IN_HZ)
_rfCat.setMaxPower()
_rfCat.makePktFLEN(0)
_rfCat.setModeTX()
```

Figure 8. RFlib is used to jam a predefined frequency, here 1575.42 MHz.

3.3.4. Rogue BTS

A rogue BTS is another method of spoofing, aiming to impersonate a telephone antenna to read the data passing through it. Victims send data through this antenna thinking it is legitimate, and the attacker can then decrypt it in offline mode and obtain compromising information while continuing to transmit the information to the legitimate network [51]. This type of attack is simple to implement, although it requires certain information about the victim's system, in particular their mobile provider, which can usually be determined from the phone number code and therefore requires knowledge of the victim's telephone number. This information is necessary because each operator transmits on different frequencies, which must be known when the attack is set up. Once again, the arrival of SDR technologies has made the implementation of such attacks much easier. With today's technology, it is possible to create a fully portable Rogue BTS with a raspberry Pi (or any other microcomputer) and external batteries, making the system lightweight and able to fit into a backpack. Because of this ease of implementation, several attacks have already been executed, notably at DEFCON 2016, where several fake antennas were spotted [52].

The Rogue BTS attack was once again carried out with BladeRF, this time using YateBTS, which is a "Software-defined Mobile Network". This tool allows for the creation of a personally owned mobile phone antenna and thus acts as the perceived operator. Once set up, it is possible to create and manage a mobile communication network and to freely communicate with any node of the network without any fees. YateBTS is highly customised which allows the impersonation of other operators. In this case, the local operator's 3G network information was inserted in order to spoof one of their antennas. The data on the frequencies and positioning of the antennas was found using Cellmapper [53], and the use of 3G was decided by watching the screen of the ACS, which used a 3G connection rather than 4G. We chose the local operator's network after reading a document from the Federal Roads Office indicating problems when using a similar vehicle due to the failure of the local operator's antenna [54]. The mobile operator was then crosschecked and confirmed by our contacts from the public transport operator. Once the dummy antenna was in place, we performed a packet analysis using Wireshark.

3.3.5. Downgrade Attack

To increase the security of communications, 3G/4G networks encrypt communications. Although it is possible to decrypt them with brute force attacks, the time required for decryption is often too long for the attacks to be considered cost-effective. However, when network coverage is not good enough to guarantee 3G or 4G communications, many devices default to 2G or EDGE connections to continue providing their communication services. Although useful for the user, this fallback solution has security limitations as

it uses the vulnerable A5/1 data encryption protocol [51]. Indeed, there are now many tools that can decrypt A5/1 encrypted data quickly and easily [55]. To achieve this goal of relying on 2G technologies, the simplest method is to degrade 3G and 4G connections by jamming their frequencies. This can again be completed with an SDR device and will, if the device allows it, force a switch to the less secure technology.

As explained, connectivity downgrade attacks rely on jamming the newest protocols (3/4/5G). Therefore, we followed the same method and code that we used for the GNSS jamming (see Section 3.3.3) by replacing the frequency to jam with the correct one.

4. Results

4.1. The TARA Showcasing

As demonstrated in Table 8, the TARA framework assessed different risks threatening the ACS security that we classify into three main groups: (i) high risks of values 4 and 5, (ii) medium risks of values 2 and 3, and (iii) low risks of value 1. The first group concerns mainly communication with the backend servers, enabling real-time data transfer and Over-the-Air (OTA) updates, and the on-board computer, on which all vehicle subsystems depend. Those attacks do not represent the vast majority of the state-of-the-art use cases, which usually imply an internal communication medium, such as CAN or Local Interconnect Network (LIN), or attacks on sensors and, hence, obtained lower values of two and three, which are significant yet unexpected. Finally, further specific attacks obtained the lowest rating value of one, as they involve tools that are difficult to put in place or have low impact.

4.1.1. High-Risk Scenarios

The scenarios obtaining the highest scores concern attacks on the means of communication as well as attacks involving physical access to the on-board computer. The former remains relatively simple to deal with as mitigation methods, such as data encryption, can be enough and are likely to be implemented by the Original Equipment Manufacturer (OEM). Consequently, it is impossible to determine whether data in transit are authenticated and whether integrity checks are carried out. However, as the encryption, authentication, and integrity checks are software-based without requiring any hardware substitution, such a setup can be implemented promptly by a team dedicated to system hardening. On the contrary, attacks involving physical access to the on-board computer require different mitigation strategies that require further hardware changes.

As many of the current CAVs are prototypes, physical security for access to the digital systems is not a high priority at the moment. This can be attributed to the experimental nature and rapid development requirements of the vehicle, which include relatively easy access to the on-board computer. However, we have to mark this as a major security risk, and it may remain a high risk if no proper anti-tempering solutions are employed. The L4V is supplied with a keyboard that can allow the user to escape the OEM's program and access the host operating system. On the same note, several active USB ports are present on the machine attracting malicious intentions to plug a rogue device into the vehicle to damage the system or steal information. Theoretically, access to such ports allows the total destruction of the on-board computer via a "USB Killer", which is able to physically destroy a computer by several 240 V discharges sent into the USB port. Nowadays, such attacks can even be performed remotely and without the computer being turned on, thanks to the USB Killer V.4 [56].

4.1.2. Medium-Risk Scenarios

Scenarios with a score of two and three are attacks that have a much lower immediate impact if carried out, although they are not without consequences. These attacks fall into two categories: attacks that cause the vehicle to stop and lead to damages and eavesdropping.

In the current framework of operations, involving a restricted route at low speed (18 km/h) with few or no other vehicles on the road, such attacks do not induce major risks for the users' safety, yet a sudden stop can cause minor disturbances. However, in a more dense traffic context, such attacks can impact both user and pedestrian lives. As with the high-risk scenarios, the mitigation strategies should encapsulate physical and software upgrades, including the implementation of cryptographic protocols for data security, as well as reinforcements to the vehicle's sensors.

Attacks that eavesdrop on data between the vehicle and the OEM's servers would not have an immediate impact on ongoing operations but would allow an attacker to obtain information about the operation of the vehicle for future privacy attacks. By decrypting the communications' keys, or if they were simply not encrypted, more knowledge about the data can be sucked up leading to new attack scenarios, such as vehicle tracking. Similarly, cryptographic protocols remain the key mitigation technique to consider.

4.1.3. Low-Risk Scenarios

Scenarios that have been given a minimum score of one do not necessarily require immediate intervention, yet they should not be underestimated. The impact of low-risk scenarios is asserted to be moderate because of their difficulty in implementation with the means currently available to attackers. However, with the emerging technologies that the attackers can afford, such risks can evolve in the near future and considerably facilitate the feasibility of the attacks in question.

To that end, several mitigation methods are proposed for the three scored groups as shown in Table 8. The suggested treatments are mainly related to the implementation of software measures, such as data encryption and hardening solutions for vehicle software components, in addition to efficient shielding of the core automated driving units, such as the on-board computer and sensors. The implementation of a fully automated mode without a wireless connection is also recommended as it decreases the jamming risk, though, it limits the chances for cooperative automated driving, which is an essential aim of smart cities. As concerns remain about the trade of maximising the readiness of self-driving operations and minimising associated cyber risks, it is crucial to set up testing tools carrying out continuous or frequent risk assessments as per pentests. The next session showcases the results of the conducted GNSS spoofing and jamming in addition to the Rogue BTS and Downgrade attacks corresponding to AP.6, AP.11, AP.13, and AP.14, respectively.

4.2. Penetration Outcome

Jamming the radio signals was the prime motivation of our pentests. One of the goals of our research was to evaluate the vehicle reaction upon a jammed signal. This was successfully demonstrated through the GNSS jamming attack. The Rogue BTS and the Downgrade attacks showcased the fairly efficient mitigation solutions in place. Moreover, the attempted black-box GNSS spoofing did not disrupt the vehicle operations pushing for further grey- or white-box bids. Another piece of evidence of the vehicle's great resistance is that no sensitive data (such as usernames or passwords) were leaked due to the pentests, which indicated the presence of a minimum of security on the vehicle. Consequently, the pentest we provide here only tests some of the vehicle's on-board systems and is constrained to a black-box environment. More extensive testing should be explored before deploying fleets of these ACSs on the road. Such matters are discussed in the following section, as well as a detailed overview of the outcome of each conducted scenario.

4.2.1. GNSS Spoofing

Whether the vehicle is in an active GNSS connection or not, the spoofing attack did not reflect a noticeable change in vehicle behaviour or metrics. In a disconnected state, the GNSS signal information remained the same according to the on-board monitor. Such a status is displayed through an orange symbol indicating that the vehicle is not receiving valid GNSS data. The main reasoning for such results is the dismissed access to the system

logs and the limitation on the testing equipment or system knowledge. A lack of power in BladeRFx40, a safety device set up by the vehicle, or the angle of arrival of the signals to the GNSS antenna can be examples of such reasoning. On the same note, as the GNSS antenna is located on the roof of the vehicle, it is possible that the radio waves emitted by BladeRF were not received. Without access to the on-board computer logs or indications of the exact position of the vehicle, it is difficult to state the exact reason for the shuttle's inability to connect to our signals. These results were identical for all three positions and both vehicle configurations, totalling six tests.

4.2.2. GNSS Jamming

The jamming attacks produced results fulfilling our expectations yet without great surprises. As the vehicle requires radio communication systems, it is conspicuous that blocking such signals implies that the vehicle will be disrupted or forced to a halt. This point implies a fundamental modification of the vehicle program through the implementation of a fully automated offline mode. In fact, jamming attacks are frighteningly easy to set up despite the legal constraints on their use. Therefore, in the current configuration, any owner of an SDR platform is capable of completely blocking the operations of the vehicle as it is set to stop immediately when the signal is lost. A remote control system can be considered to support the circumvention of such situations; however, the use of radio waves alone would not solve the problem since it would again be possible to jam this particular connection and thus prevent remote troubleshooting. Therefore, a fully automated offline mode allowing the vehicle to move to the side of the road or to an area suitable for dropping off its passengers should be considered. This system, therefore, leaves the door open for various improvements with other radio communication information.

4.2.3. Rogue BTS

The packets captured by Wireshark during the implementation of the Rogue BTS confirmed the encrypted network connection. Cross-checked with the public transport operator team, it was asserted that a Virtual Private Network (VPN) connection is built between the OEM's backend servers and the vehicle. Hence, the data in transit is encrypted from end to end and can therefore be considered secure. However, it is still possible to break the encryption keys in offline mode using existing tools such as Hashcat. Such a practice is usually too time-consuming to be cost-effective [51]. Additionally, if the encryption keys are changed regularly (respecting perfect forward secrecy), breaking one of them will not allow the decryption of all communications but only those of the specific session of the key. Thus, Rogue BTS attacks can be considered ineffective against this vehicle.

4.2.4. Downgrade Attack

Despite successfully jamming of the mobile network, we can see that the vehicle did not have a fallback function on a 2G (GSM) network as it simply indicated that no mobile connection was available. It is therefore not possible to exploit downgrade attacks on connectivity in that case.

5. Discussion & Future Work

Our research goal was twofold: provide recommendations upon the findings from the TARA and the pentests and study the identified research questions to set up comprehensive insight on the correlation between the cyber risks impacts and the vehicle automation level. Our work aims to support in reinforcing security requirements for a future concrete deployment of the ACS going beyond pilot site testing.

5.1. Recommendations

Based on the results from the TARA and the pentests, we believe that human intervention, and hence the vehicle automation level, have a direct impact on the assessed risks. Some attacks, particularly GNSS spoofing, are only applicable if there is no driver who can

immediately take control of the vehicle if it goes off the road. In other words, a moderate risk in a vehicle of L4 can be considered severe in an L5 vehicle unless robust and flawless mitigation strategies are implemented.

To strengthen the entire cybersecurity governance for L4V and support the ACS L5 readiness, the following crucial, yet non-exhaustive, recommendations are delineated:

- **Physical strengthening:** where LiDARs, cameras, USB ports, and the on-board computer are unreachable and protected from any unwarranted access.
- **Fully automated offline and resilient mode:** providing high protection against jamming attacks and unjustified halt or vehicle stops at a complete connectivity loss.
- **Confidentiality and integrity of communications:** where Private Key Infrastructure (PKI) and digital signatures can be used to secure authentications in addition to HTTPS and IPSec tunnel mode (such as VPN) establishment.
- **Hardening of the on-board computer:** which relies on (i) protecting the BIOS through Root of Trust for Update (RTU) and Trusted Platform Module (TPM) usage during the firmware update [57], (ii) shielding the disk protection through Bitlocker [51], and (iii) adopting operating system best practices, such as the installation of a Host-based Intrusion Prevention System (HIPS) and applying restrictive policies on the listing of files and their modification.
- **Standardised security procedures and certifications:** varying from conducting Cybersecurity Management System (CSMS) [18] and Software Update Management System (SUMS) [58] certifications mandated by the UNECE to comply with ISO/SAE 21434 [17] and ISO/PAS 5112 [59].
- **Security monitoring:** where continuous and frequent assessments are conducted and risks are monitored using the integration of a Security Information and Event Management (SIEM), for example.

5.2. Research Questions Analyses

To answer RQ1, the present work provided a systematic categorisation and analyses of cybersecurity risks by applying the TARA to the ACS domain. Three main groups were identified: high (risk values of four and five), medium (risk values of two and three) and low (risk value of one). Although the TARA is suitable for threat modelling and analysing risks, it remains limited in assigning an objective risk value with regard to the automation level. In fact, the weight of the automation level depends on the experts' opinion and their expertise. Furthermore, being an asset-based methodology, the automation features are impossible to evaluate as a single asset from the TARA.

Limited by several real-life pilot restrictions, the pentests that we managed to execute confirm the ease of the necessary setup for an attacker to execute high-risk scenarios. In particular, the affordability of the equipment required as well as the short timespan in which an attacker can perform an attack, have been demonstrated. As far as 3/4/5G jamming is concerned, the most cost-effective solution in terms of time is sweep jamming on the frequencies of the most widely used operators, which means that it is not necessary to find out which operator is used by the manufacturer. However, the black-box penetration tests and vehicle resilience that we observed did not provide any additional insights into whether the vehicle was affected by the intended malicious activities. Therefore, we cannot confirm if the mitigations applied to the vehicle were sufficient; hence, black-box penetration testing is not suitable. To answer RQ2, we believe that the openness of the OEM's ACS ecosystem towards elevating the restriction on internal data access (e.g., logs) is required to both execute physical attacks and cross-check the effectiveness of the conducted wireless pentests.

5.3. Limitations & Future Work

Following up on the discussion about unwarranted on-board access, it is noteworthy to highlight that diving deeply into the vehicle logs represents a real limitation to verifying the evident effects of our pentests. The restrictions also made the entire pentest more complex

as it was limited to being pushed in a black-box manner. Therefore, our future efforts are focused on conducting grey and white pentests. More specifically, it is envisioned to target further assets varying from automated driving decision-making units, V2X components, on-demand service applications, and the fleet management system.

Additionally, considering the continuous upgrades impacting the vehicle operating systems, supplementary tests are foreseen to accomplish future comparative analyses with the present findings. On the same note, for a more granular and uniform analysis, it is planned to complement the attack tree paths with an additional detailed level linking CVEs to each damage scenario. Such a future work would provide consonant comparisons and an evolution of the identified vulnerabilities based on the universal CVE databases [33].

Another shortcoming to highlight in the present research is the impact of the rapidly evolving technologies on the pertinence of our findings. Being a pilot vehicle under regular emerging changes, the L4V has been subject to several modifications and upgrades. Consequently, our findings reflect the risk analysis and pentests results on the assessed vehicle configuration at the time of the elaboration of our experiments. As a future work, we intend to build an automated TARA framework supporting with continuous assessment of the L4V risks with a possible comparison of current risk values to the historical records. Such a solution aims to help keep risks at an acceptable level while coping with the technological progression.

To that end, the present work can be considered a valuable path and a starting point advertising the implementation of frequent risk assessments and the importance of penetration testing on approaching the full deployment of L4 and L5.

6. Conclusions

The objective of this work is to provide the first example of a cybersecurity analysis on an L4 ACS. Based on the TARA framework, threat modelling and risk analysis of the ACS were outlined on the selected vehicle assets. We elevated further the risk analyses findings by conducting four pentest scenarios focused on GNSS and 4G connections. Based on the implementation results, we proposed several mitigation solutions and technical recommendations to be implemented in future iterations. The outcome showed that the automation level is still a missing attribute throughout the TARA process, yet it has a direct impact while selecting accurate mitigation strategies with consideration of human intervention. We further identified a set of limitations that trigger motivation for future efforts.

Author Contributions: Conceptualization, M.B., A.C. and N.A.N.; methodology, M.B., A.C. and N.A.N.; software, P.B.; formal analysis, M.B., A.C. and N.A.N.; investigation, P.B.; writing—original draft preparation, P.B.; writing—review and editing, M.B., A.C. and N.A.N.; visualization, P.B. and N.A.N.; supervision, A.C. and N.A.N.; All authors have read and agreed to the published version of the manuscript.

Funding: This work has received funding from the European Union's Horizon 2020 Research and Innovation Programme (grant agreement no. 875530) and the Swiss State Secretariat for Education, Research and Innovation (SERI) co-funded by the European Union (grant agreement no. 101077587). The views and opinions expressed herein are, however, those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the transports publics genevois (TPG) for their collaboration, more specifically Melisa Fazlic and Jeroen Beukers, without whom it would not have been possible to carry out this project.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AAFL	Aggregated Attack Feasibility Level
ACS	Automated City Shuttle
AI	Artificial Intelligence
BTS	Base Transceiver Station
CAN	Controller Area Network
CAV	Connected Automated Vehicle
CIA	Confidentiality, Integrity, Availability
CSMS	Cybersecurity Management System
CVE	Common Vulnerabilities and Exposure
ECU	Electronic Control Unit
GNSS	Global Navigation Satellite System
GPS	Differential Global Positioning Systems
HIPS	Host-based Intrusion Prevention System
L4V	L4 Evaluation Vehicle
LiDAR	Light Detection and Ranging
LIN	Local Interconnect Network
NASA	National Aeronautics and Space Administration
OEM	Original Equipment Manufacturer
OTA	Over-the-Air
PKI	Private Key Infrastructure
RTU	Root of Trust for Update
SAE	Society of Automotive Engineering
SDR	Software Defined Radio
SIEM	Security Information and Event Management
STRIDE	Spoofing, Tampering, Repudiation, Information disclosure, Denial-of-service and Elevation of privilege
SUMS	Software Update Management System
TARA	Threat Analysis and Risk Assessment
TPG	transports publics genevois
TPM	Trusted Platform Module
UNECE	United Nations Economic Commission for Europe
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VPN	Virtual Private Network

References

1. Gruyer, D.; Orfila, O.; Glaser, S.; Hedhli, A.; Hautière, N.; Rakotonirainy, A. Are Connected and Automated Vehicles the Silver Bullet for Future Transportation Challenges? Benefits and Weaknesses on Safety, Consumption, and Traffic Congestion. *Front. Sustain. Cities* **2021**, *2*, 607054 . [CrossRef]
2. Deichmann, J.; Ebel, E.; Heineke, K.; Heuss, R.; Kellner, M.; Steiner, F. *Autonomous Driving's Future: Convenient and Connected*; Technical Report; McKinsey: Atlanta, GA, USA, 2023.
3. Simpson, C.; Ataii, E.; Kemp, E.; Zhang, Y. *Mobility 2030: Transforming the Mobility Landscape*; Technical Report; KPMG International: Zurich, Switzerland, 2019.
4. Litman, T. *Autonomous Vehicle Implementation Predictions*; Technical Report; Victoria Transport Policy Institute: Victoria, BC, Canada, 2013.
5. SAE. *J3016B Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*; Technical Report; SAE International: Warrendale, PA, USA, 2018.
6. Benyahya, M.; Collen, A.; Kechagia, S.; Nijdam, N.A. Automated city shuttles: Mapping the key challenges in cybersecurity, privacy and standards to future developments. *Comput. Secur.* **2022**, *122*, 102904. [CrossRef]
7. Khanam, S.; Ahmedy, I.B.; Idna Idris, M.Y.; Jaward, M.H.; Bin Md Sabri, A.Q. A Survey of Security Challenges, Attacks Taxonomy and Advanced Countermeasures in the Internet of Things. *IEEE Access* **2020**, *8*, 219709–219743. [CrossRef]
8. Ainsalu, J.; Arffman, V.; Bellone, M.; Ellner, M.; Haapamäki, T.; Haavisto, N.; Josefson, E.; Ismailogullari, A.; Lee, B.; Madland, O.; et al. State of the art of automated buses. *Sustainability* **2018**, *10*, 3118. [CrossRef]
9. NHTSA. *Automated Vehicles for Safety*; National Highway Traffic Safety Administration: Washington, DC, USA, 2022 .

10. Duarte, F.; Ratti, C. The Impact of Autonomous Vehicles on Cities: A Review. *J. Urban Technol.* **2018**, *25*, 3–18. [CrossRef]
11. Al-Sabaawi, A.; Al-Dulaimi, K.; Foo, E.; Alazab, M. Addressing Malware Attacks on Connected and Autonomous Vehicles: Recent Techniques and Challenges. In *Malware Analysis Using Artificial Intelligence and Deep Learning*; Springer: Cham, Switzerland, 2020; pp. 97–119. .[CrossRef]
12. Iclodean, C.; Cordos, N.; Varga, B.O. Autonomous shuttle bus for public transportation: A review. *Energies* **2020**, *13*, 2917. [CrossRef]
13. Bec, P.; Borzan, A.I.; Frunză, M.; Băldean, D.L.; Berindei, I. Study of vulnerabilities in designing and using automated vehicles based on SWOT method for chevrolet camaro. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Oradea, Romania, 28–29 May 2020; Volume 898, p. 12008.
14. Miller, C.; Valasek, C. Remote Exploitation of an Unaltered Passenger Vehicle. *Defcon 23* **2015**, *2015*, 1–91.
15. Yan, C.; Xu, W.; Liu, J. Can You Trust Autonomous Vehicles: Contactless Attacks against Sensors of Self-driving Vehicle. *DEFCON* **2016**, *24*, 109.
16. The Avenue Consortium. AVENUE—EU Funded Project Under Horizon 2020. 2022. Available online: https://h2020-avenue.eu/ (accessed on 29 November 2022).
17. *ISO/SAE 21434*; Road Vehicles-Cybersecurity Engineering. ISO: Geneva, Switzerland; SAE International: Warrendale, PA, USA, 2021.
18. *R155*; UN Regulation No. 155—Cyber Security and Cyber Security Management System. UNECE: Geneva, Switzerland, 2020.
19. Fysarakis, K.; Askoxylakis, I.; Katos, V.; Ioannidis, S.; Marinos, L. *Security Concerns in Cooperative Intelligent Transportation Systems*; CRC Press: Boca Raton, FL, USA, 2017; pp. 487–522. [CrossRef]
20. Marin-Plaza, P.; Yaguüe, D.; Royo, F.; de Miguel, M.A.; Moreno, F.M.; Ruiz-de-la Cuadra, A.; Viadero-Monasterio, F.; Garcia, J.; San Roman, J.L.; Armingol, J.M. Project ARES: Driverless Transportation System. Challenges and Approaches in an Unstructured Road. *Electronics* **2021**, *10*, 1753. [CrossRef]
21. Islam, M.M.; Lautenbach, A.; Sandberg, C.; Olovsson, T. A risk assessment framework for automotive embedded systems. In Proceedings of the CPSS 2016—Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security, Co-Located with Asia CCS 2016, Xi'an China, 30 May 2016; pp. 3–14. [CrossRef]
22. Wang, Y.; Wang, Y.; Qin, H.; Ji, H.; Zhang, Y.; Wang, J. A Systematic Risk Assessment Framework of Automotive Cybersecurity. *Automot. Innov.* **2021**, *4*, 253–261. [CrossRef]
23. Lautenbach, A.; Almgren, M.; Olovsson, T. Proposing HEAVENS 2.0—An automotive risk assessment model. In Proceedings of the Proceedings—CSCS 2021: ACM Computer Science in Cars Symposium, Ingolstadt, Germany, 30 November 2021. [CrossRef]
24. Vogt, T.; Spahovic, E.; Doms, T.; Seyer, R.; Weiskirchner, H.; Pollhammer, K.; Raab, T.; Rührup, S.; Latzenhofer, M.; Schmittner, C.; et al. A Comprehensive Risk Management Approach to Information Security in Intelligent Transport Systems. *SAE Int. J. Transp. Cybersecur. Priv.* **2021**, *4*, 39–58. [CrossRef]
25. Cao, Y.; Bhupathiraju, S.H.; Naghavi, P.; Sugawara, T.; Mao, Z.M.; Rampazzi, S. You Can't See Me: Physical Removal Attacks on LiDAR-based Autonomous Vehicles Driving Frameworks. *arXiv* **2022**. [CrossRef]
26. Petit, J.; Stottelaar, B.; Feiri, M.; Kargl, F. Remote Attacks on Automated Vehicles Sensors: Experiments on Camera and LiDAR; BlackHat Europe: Amesterdam, The Netherlands, 2015; pp. 1–13.
27. Andersson, P. Penetration Testing of an In-Vehicle Infotainment System. Ph.D. Thesis, KTH Royal Institute of Technology, Stockholm, Sweeden, 2022.
28. Moukahal, L.J.; Zulkernine, M.; Soukup, M. Vulnerability-Oriented Fuzz Testing for Connected Autonomous Vehicle Systems. *IEEE Trans. Reliab.* **2021**, *70*, 1422–1437. [CrossRef]
29. Openpilot. Open Source Advanced Driver Assistance System. 2023. Available online: https://comma.ai/openpilot (accessed on 9 February 2023).
30. Fowler, D.S.; Bryans, J.; Cheah, M.; Wooderson, P.; Shaikh, S.A. A Method for Constructing Automotive Cybersecurity Tests, a CAN Fuzz Testing Example. In Proceedings of the Companion of the 19th IEEE International Conference on Software Quality, Reliability and Security, QRS-C 2019, Sofia, Bulgaria, 22–26 July 2019; pp. 1–8. [CrossRef]
31. Zinckernagel, C.; Lutgens, E. *AVENUE: D2.2 Gap aNalysis and Recommendations on Autonomous Vehicles for Public Service*; Technical Report; Autonomous Mobility: Copenhagen, Denmark, 2019.
32. Microsoft . Microsoft Threat Modeling Tool. 2023. Available online: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats#stride-model (accessed on 13 January 2023).
33. National Institute of Standards and Technology; US Department of Commerce. National Vulnerability Database . Available online: https://nvd.nist.gov/vuln/full-listing (accessed on 3 March 2023).
34. Sommer, F.; Dürrwang, J.; Kriesten, R. Survey and classification of automotive security attacks. *Information* **2019**, *10*, 148. [CrossRef]
35. Nuand. bladeRF x40 . 2023. Available online: https://www.nuand.com/product/bladerf-x40/ (accessed on 14 February 2023).
36. GNU Radio Project. GNU Radio—The Free & Open Source Radio Ecosystem GNU Radio. 2022. Available online: https://www.gnuradio.org/ (accessed on 23 February 2023).
37. Google Play. GPS Test Applications sur Google Play. 2023. Available online: https://play.google.com/store/apps/details?id=com.chartcross.gpstest (accessed on 23 February 2023).

38. GitHub. Software-Defined GPS Signal Simulator. 2023. Available online: https://github.com/osqzss/gps-sdr-sim (accessed on 8 February 2023).
39. Csete, A. Welcome to Gqrx. 2023. Available online: https://gqrx.dk/ (accessed on 8 February 2023).
40. PyPi. Welcome to the rfcat Project. 2023. Available online: https://pypi.org/project/rfcat/ (accessed on 8 February 2023).
41. Canonica. Enterprise Open Source and Linux Ubuntu. 2023. Available online: https://ubuntu.com/ (accessed on 8 February 2023).
42. YateBTS. LTE & GSM Mobile Network Components for MNO & MVNO. 2021. Available online: https://yatebts.com/ (accessed on 8 February 2023).
43. Wireshark. About Wireshark. 2023. Available online: https://www.wireshark.org/ (accessed on 8 February 2023).
44. Elliott, D.; Keen, W.; Miao, L. Recent advances in connected and automated vehicles. *J. Traffic Transp. Eng.* **2019**, *6*, 109–131. [CrossRef]
45. Li, C.; Fu, Y.; Yu, F.R.; Luan, T.H.; Zhang, Y. Vehicle Position Correction: A Vehicular Blockchain Networks-Based GPS Error Sharing Framework. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1–15. [CrossRef]
46. Mangialardo, M.; Jurado, M.M.; Hagan, D.; Giordano, P.; Ventura-Traveset, J. The Full Potential of an Autonomous GNSS Signalbased Navigation System for Moon Missions. In Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation, St. Louis, MI, USA, 20–24 September 2021; pp. 1039–1052. [CrossRef]
47. Karki, B.; Won, M. Characterizing Power Consumption of Dual-Frequency GNSS of Smartphone. In Proceedings of the 2020 IEEE Global Communications Conference, GLOBECOM 2020, Taipei, Taiwan, 7–11 December 2020. [CrossRef]
48. National Aeronautics and Space Administration. NASA's Archive of Space Geodesy Data. 2023. Available online: https://cddis.nasa.gov/ (accessed on 13 February 2023).
49. Le Conseil fédéral. Protection de Récepteurs GPS Contre Des Cyberattaques. 2022. Available online: https://www.admin.ch/gov/fr/accueil/documentation/communiques.msg-id-69896.html (accessed on 8 February 2023).
50. Office Fédéral De La Communication OFCOM. Perturbateurs (Jammers). 2022. Available online: https://www.bakom.admin.ch/bakom/fr/page-daccueil/appareils-et-installations/equipements-particuliers/perturbateurs-jammers.html (accessed on 8 February 2023).
51. Knight, A. *Hacking Connected Cars: Tactics, Techniques and Procedures*; John Wiley & Sons: Hoboken, NJ, USA, 2020.
52. Cox, J. Surprise! Scans Suggest Hackers Put IMSI-Catchers All Over Defcon. 2022. Available online: https://www.vice.com/en/article/vv7zn9/surprise-scans-suggest-hackers-put-imsi-catchers-all-over-defcon (accessed on 8 February 2023).
53. CellMapper. Swisscom (Switzerland)—Cellular Coverage and Tower Map. 2023. Available online: https://www.cellmapper.net/ (accessed on 8 February 2023).
54. Office Feédeéral Des Routes. *Complément Au Rapport Final De L'étude De Suivi HEIA-FR*; Technical Report; Transports Publics Fribourgeois: Fribourg, Switzerland, 2020.
55. GitHub. GSM Description. 2023. Available online: https://github.com/0xh4di/GSMDecryption (accessed on 8 February 2023).
56. USBKill. USBKill V4. 2022. Available online: https://usbkill.com/products/usbkill-v4?variant=32836117397586 (accessed on 8 February 2023).
57. CITS. Secure Firmware Update. 2017. Available online: https://cts-labs.com/secure-firmware-update (accessed on 14 February 2023).
58. *R156*; UN Regulation No. 156—Software Update and Software Update Management System. UNECE: Geneva, Switzerland, 2020.
59. *ISO/PAS 5112*; Guidelines for Auditing Cybersecurity Engineering. ISO: Geneva, Switzerland, 2022.