2021 IEEE Global Engineering Education Conference (EDUCON) | 978-1-7281-8478-4/20/$31.00 ©2021 IEEE | DOI: 10.1109/EDUCON46332.2021.9454086

Education in Cyber Physical Systems Security: The Case of Connected Autonomous Vehicles

Sophia McCall
Computing and Informatics
Bournemouth University
Poole, UK
s4916645@bournemouth.ac.uk

Cagatay Yucel
Computing and Informatics
Bournemouth University
Poole, UK
cyucel@bournemouth.ac.uk

Vasilios Katos
Computing and Informatics
Bournemouth University
Poole, UK
vkatos@bournemouth.ac.uk

***Abstract*—The automotive industry is a dynamic industry that is constantly evolving and changing with the advancements of technology. As cars become more technology dependent, the threat landscape and likelihood of a cyber-attack becomes greater and inherently larger as issues arise. With the introduction to automation and increased use of embedded systems and infotainment systems, modern cars have become a pillar piece of the Internet of Things network.**

This research details an in-depth study into the vulnerabilities and risks surrounding the current and future state of the automotive industry, highlights the most safety-critical components of the modern car, providing a holistic threat landscape to improve security awareness and posture regarding automotive security. It also demonstrates the utilisation of this analysis with the integration of an education package built on top of a hardware module based on a Raspberry Pi, that emulates its own CAN Bus network that individuals can interact with as if it was a vehicle to provide education on CAN hacking. This device has the potential to be attached to education ranges and labs which can help educate individuals on different security skills to help improve security awareness and knowledge.

***Index Terms*—automotive security, control area network, CAN Bus, risk-based approach, Arduino Teensy**

I. Introduction and motivation

The automotive industry is a rapidly and constantly evolving industry, encompassing and embracing IT networks, computing and, information and communications technology (ICT) systems in general. As cars become more technology dependent and connected, the threat landscape and likelihood of a cyberattack becomes greater and inherently larger. With the introduction to automation and increased use of embedded systems (such as in car infotainment systems), modern cars have become a showcase of Internet of Things (IoT) capabilities. At the same time, the increased connectivity would provide wider opportunities for malicious actors exploiting the devices. With this larger attack surface, it can be argued that the likelihood of a cyberattack is higher, with vehicular entities being more prone and vulnerable to attack and compromise.

In this paper, an approach towards developing a training methodology for the aspects of cyber-physical systems (CPS) security to the undergraduate students in computing degrees is presented. As in CPS, the physical plane interacts with the cyber plane through IoT sensors and actuators that serve as the conduit between these two worlds, it is imperative to realise that it is possible for risks to propagate across these two planes. As such, the traditional assumptions and goals of cybersecurity - pertaining to Confidentiality, Integrity and Availability - will need to be extended to include also safety.

As a vehicle for demonstrating the above concepts, we consider the case of connected autonomous vehicles (CAV). Our methodology follows a risk based approach; that is, we initially enumerate all identifiable risks associated with a CAV environment and we introduce a narrative where a threat actor can attack aspects of this environment. To this end, we develop an education pack with appropriate learning outcomes and show how these risks are met through the deployment and delivery of a test bed using custom hardware based on a Raspberry Pi, a programmed Arduino Teensy 3.2 replicating Engine Control Unit (ECU) heartbeats that are relayed through a Controller Area Network (CAN) Bus Transceiver and a PiCAN Shield to create an isolated CAN Bus network which is interacted through the Raspberry Pi.

II. Cybersecurity in the automotive sector

Modern automotive designs contain hundreds of cyber-physical modules, connectivity components and microprocessors that work together in unison to control a vehicle mechanically and electronically. Security remains to be an ongoing and fundamental challenge in the design and manufacturing process of a vehicle [1]. Automotive security is driven by safety-critical decisions, challenged by the evolution of technology and the need for real-time mitigation against environmental threats [1]. Some identified technologies include the following:

- Infotainment Systems and Components
- Driver Assistance Capabilities (e.g. Collision Detection, Emergency Braking, Engine/Tyre Sensors)
- Physical Security (e.g. CAN Bus/Onboard Diagnostics (OBD)-II Diagnostics)
- Remote Entry Security (e.g. Keyless Entry Attacks)
- Telematics Modules
- GPS/Global Navigation Satellite System (GNSS) (e.g. Navigation System or Positional Sensors)
- Over-The-Air Software/Firmware Updates

This work has received funding from the European Union's Horizon 2020 research and innovation program under the grant agreement no 830943 (ECHO).

As CAVs continue to be introduced into the automotive market, identification of the security risks that these vehicles may impose will keep its importance. Autonomous vehicles heavily rely on a variety of sensors, radars, and camera components to operate correctly and safely. Attacks on these sensors can prove fatal for drivers and passengers alike; it is important to mitigate threats that target these components. Attacks against autonomous sensors can have heavy consequences; in addition to this type of attack, there is also a threat against the Artificial Intelligence (AI) and Machine Learning (ML) aspects of an autonomous vehicle [2]. Malicious actors can tamper the decision-making algorithms within AI or ML features or tamper with the inputs these features receive. Combining these attack vectors constructs a wide attack surface on a typical autonomous vehicle. With the vulnerability surface exacerbated with the introduction of connected and autonomous technologies – security assessments and decisions must be made to protect the safety-critical aspects of the vehicle.

Socio-technical measures must be implemented into the industry to ensure holistic mitigation can be applied to reduce the security risks and raise awareness on a problem realm within security. The interactions between people and technology remain pinnacle to the three pillars of information security management: people, processes, and technology [3]. The introduction and implementation of education and awareness programs will help individuals understand the problem area and realm with deeper comprehension. By creating an education exercise, such as the artefact built in this project, individuals that have a technical and/or security background in the automotive industry can widen their skill set and understanding to initiate principles such as security-by-design – combating initial risks from the design stage in the manufacturing process of vehicles. Education exercises are also fundamental for academia and students within the security field – it is critical that those in education remain up-to-date in developing their skill sets and in-line with the times.

TABLE I
ATTACK SURFACE FOR THE CURRENT STATE OF AUTOMOTIVE SECURITY

System	Asset	Threat	CIA Affected
IVI [5]	USB Port, Connected Device, WiFi Module, Bluetooth, GPS	Unauthorised installation of malicious software/firmware, Sniffing wireless data, Jamming/Spoofing of GPS data	Confidentiality, Integrity and Availability
ADAS Sensors [6]	Anti-lock Braking systems (ABS), Tyre Pressure Sensors, Engine Sensors, Emergency Braking Capabilities, Parking Sensors	Sensor failures, Denial-of-Service, Misconfiguration of Components	Availability and Integrity
OBD-II and CAN Bus Systems [7]	CAN Bus	Tempering /Manipulation of CAN Data via hijacked ECUs or unauthorised access to OBD-II port, Jamming Attacks on OBD-II,	Availability and Integrity
Keyless Technologies [7]–[9]	Vehicle/Key Fob	Relay Attack to Open/Start a Car, Clone Key Fob, Jamming Attack	Confidentiality and Availability
Telematics Modules [10], [11]	Telematics Control Unit (TCU), Vehicle Subscriber Identification Module, Mobile Applications hosted by Telematics Service Platforms (TSPs)	Data Spoofing, Jamming Attacks, Sniffing Attacks in Communications, Account Compromise	Confidentiality, Integrity and Availability
Over-the-Air (OTA) Software/ Firmware Updates [12]	Software/ Firmware Update Package	Intercepting transit (Man-in-the-Middle Attack) to tamper/modify update, DoS during transmission and/or storage	Availability and Integrity

III. ATTACK SURFACE AND RISK ANALYSIS OF CAVS

To establish a deeper understanding of the problem realm and achieve a greater situational awareness of the automotive industry and its relevant security posture, this section details the research undertaken to identify current technologies that can be exploited and future technologies that have the potential in being compromised.

A. Current State Automotive Security

Modern cars are becoming more connected and complex, linking a plethora of connected components and technologies – a new age of vehicles is now available within the automotive market. Connected technologies allow vehicles to be more efficient, passengers and drivers to be more comfortable, and safety systems to be more accurate and reliable – however, with these introduced technologies, connected systems are becoming more vulnerable to security attacks [4]. In the current state of the systems, the attack surface can be extended on the aforementioned identified technologies on security and supportive systems.

Table I outlines the attack surfaces for the contemporary technologies. The attack surface is provided with the assets included in the technology, the threat and the affected information security attribute given in the Confidentiality-Integrity-Availability model.

B. Future State Automotive Security

The future of automotive is moving towards a more connected and autonomous paradigm, with a predicted 8 million fully autonomous vehicles available and roadworthy by 2025 [13]. In addition to current automotive technologies and integration into the Internet of Things (IoT), the design of vehicles is transforming towards an AI-driven future – utilising current connected technologies and exemplifying them to aid the movement towards driverless as the norm. To cope with the pressures of an Intelligent Transport System (ITS), new

TABLE II
ATTACK SURFACE FOR THE FUTURE STATE OF AUTOMOTIVE SECURITY

System	Asset	Threat	CIA Affected
Artificial Intelligence and Machine Learning Systems [2], [15]	Related Sensors and Actuators, Decision making algorithms	DoS on Sensors, Adversarial Perturbation to manipulate algorithms, Malicious inputs during the training of the algorithms	Availability and Integrity
Autonomous Cameras and Sensors [6], [16]	Camera Sensor	DoS via blinding or jamming attack, manipulating scenery to fool sensors (e.g. fake speed signs)	Availability and Integrity
LiDAR Technologies [16], [17]	LiDAR Sensor	DoS via jamming, Replay Attack, Spoofing Attack	Availability and Integrity
V2X Communications [16]	V2X Communication Data	Sniffing attack, MITM attacks, DoS to communications via jamming	Confidentiality, Integrity and Availability

TABLE III
VULNERABILITIES, THEIR CVE CODES AND CVSS SCORES

Vulnerability CVE Id	Affected Systems	CVSS Score
CVE-2017-9633	An Improper Restriction of Operations within the Bounds of a Memory Buffer issue affecting many brands including some models of BMW, Hyundai, Nissan and Ford	8.3
CVE-2017-9647	Stack based buffer overflow on ECUs affecting many brands including some models of BMW, Hyundai, Nissan and Ford	6.6
CVE-2017-14937	Airbag Control Units – through CAN Bus, OBD-II Ports	4.7
CVE-2018-9318	Remote attack via a cellular network to TCU affecting BMW vehicles produced in 2012 through 2018	10
CVE-2019-12797	OBD-II – sending arbitrary commands to the OBD-II Bus through Bluetooth	7.5
CVE-2020-12323	Privilege Escalation on Intel's ADAS IE	7.5

infrastructure technologies such as the introduction of 5G Cellular and ITS-G5 are utilised to support the growth of smart cities and vehicles. V2X communications are a fundamental part of the ITS paradigm of the future. The future of smart cities and V2X data exchange will utilise the current technologies and introduce further IoT components such as roadside technology and real-time traffic and environment data to produce safety alerts and improve the efficiency of the connected and autonomous vehicle [14].

The advancement of such technologies are industry-driven by an expectation for a progress in CAVs and IoT integration. With a predicted growth of an autonomous future, it is paramount to implement the correct safety and security protocols to cope with introduced technologies and their vulnerabilities and/or flaws. Table II detail a selection of technologies exercised in autonomous and connected vehicles of the future, including their identified risks and vulnerabilities.

C. Vulnerabilities

As part of a risk based approach, the research conducted in this section continues with the vulnerability analysis of the components stated in Table I and the vulnerabilities for the components of CAVs are stated in Table III. In most cases, the assets include and utilise components from generic CPU and micro-controller producers such as Intel, ST, Qualcomm etc. In addition to this, the ECUs may contain or interact with the operating systems that are running on those devices such as Android or Linux kernels. As such, these components inherit generic OS vulnerabilities as well, however for the purpose of this paper we focus only to those that are specific to CAV systems.

An example attack vector containing the vulnerabilities listed in Table III can be as follows;

1) Initial access via a cellular network (CVE-2018-9318)
2) Unauthorised code execution with CVE-2017-9647
3) Bluetooth jamming using the vulnerability on OBD-II ports with CVE-2019-12797
4) Destruction and detonation of airbags using malevolent access to OBD-II ports (CVE-2017-14937)

D. Threat actors

In order to complete the risk assessment, we enumerate the threat actors and their motives against CAV assets. Understanding threat actors and their motives can help alleviate the understanding of the security posture in the automotive industry and improve security controls and practices to enhance security defence mechanisms and mitigations. By profiling potential attackers, we can zoom into their capabilities and identify the most detrimental risks and likely attack methods.

The following types of threat agents would be relevant to the automotive ecosystem [18]:

- **Security researchers**. Often from academia, industry or government – security researchers are typical, although not always, recruited through schemes such as bug bounty programs to find security vulnerabilities that haven't been identified yet. Issues identified are usually disclosed to vendors and manufacturers, however, many researchers also opt to share discovered vulnerabilities either through online forums or in large security gatherings such as conferences. By freely sharing such information, cybercriminals or "script kiddies" may utilise this public knowledge to maliciously attack vendors using unknown exploits or attacks.
- **Hacktivism groups**. Hacktivist groups are often large groups that use their hacking abilities to demonstrate or project ideas. Hacktivist groups may be politically charged, often challenging government – or have internal motives of their own to promote social justice.
- **Script kiddies and pranksters**. Individuals with novice hacking ability and minimal resources, script kiddies and

pranksters may attack targets using pre-written tools or exploits to primarily cause nuisance or for prestige in the hacking culture.

- **Owners**. Car-hacking tools are already publicly available for owners to access. Rather than malice, many owners may want to hack their vehicles or attack security features to remove manufacturer implemented restrictions. This may include performance restrictions, e.g. Increasing engine power.
- **Organised crime groups (OCGs)**. OCGs pose as one of the biggest threat actors to automotive. Usually hosting extensive resources, both financially and knowledge-based, organised crime groups tend to gravitate towards a financial motive. Many groups may target stealing cars, to sell on for profit. Cybercrime syndicates such as OCGs usually use a collection of attacks that closely follow the Cyber Kill Chain [19] to achieve objectives. Attacks may also closely follow the MITRE ATT&CK Framework [20].
- **Advanced Persistent Threats (APTs)**. APTs are often backed by hostile nations and governments and are difficult to identify. Mostly used for espionage or cyberwarfare, APT groups are often used to target rival nations and cripple national infrastructure, manufacturers and vendors. Similar to organised crime groups, APT groups closely follow the Cyber Kill Chain to achieve objectives.
- **Cyber terrorist groups**. These actors utilise computers and technology to execute attacks to widespread fear within the general public and/or cause harm or disruption.

The actors' motives can be diverse, from plain hacktivism and pranking, to financial gain driven by stealing the actual car, ransomware (by disabling or assuming control of the car), to more severe cyber terrorism types of attack, by attempting to cause major disruptions and even loss of life. As the potential impact of a compromised car can be severe, it is imperative that CPS security should be woven into the cybersecurity curriculum.

IV. Exercise Design

In what follows we describe the design process of an education exercise created to improve automotive hacking knowledge focused on the CAN Bus manipulation vulnerability within a vehicle as informed by the risk based approach presented above. In an educational setting, individuals will be able to interact with the simulated CAN network created by a custom Raspberry Pi, to represent and replicate how a malicious actor can attack the CAN network of a car. After analysing the risks of a modern vehicle, and the risks of future CAVs - an education exercise focused on a CAN component was selected as the CAN Bus has always traditionally been the easiest way to compromise a vehicle. This is due to the requirement that most cars have to implement CAN as part of the five protocols used in the on board diagnostic (OBD-II) standard for modern vehicles.

A. Hardware customisation and integration

The four main hardware modules are the Raspberry Pi, the PiCAN Shield, the Arduino Teensy and the CAN Transceiver, as elaborated below:

1) Raspberry Pi 3 Model B: The Raspberry Pi was the main component of the custom hardware device. It fulfills the technical requirements needed for the project and has the capability to support the additional hardware components needed for the final device creation. Implemented as a core system, the Raspberry Pi works in unison with the PiCAN shield to deliver interaction with the created CAN network. The Raspberry Pi, through a terminal command prompt, uses the Linux utilities package "can-utils" to send commands to interact through the PiCAN to manipulate the CAN traffic generated by the other hardware components.

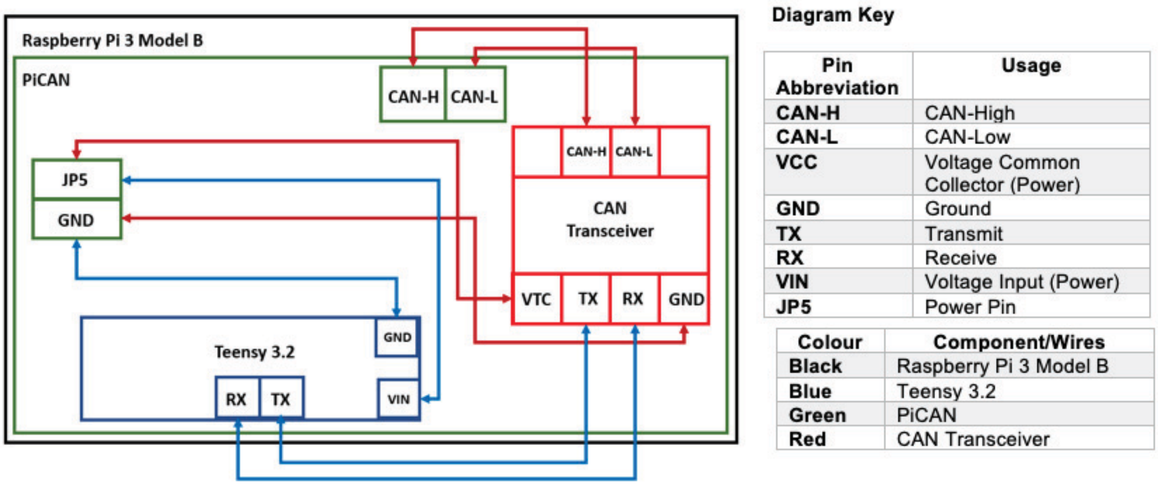
2) PiCAN Shield: A PiCAN shield is attached to the Raspberry Pi in order to provide the capability to interact with CAN. Created by SKPang, the PiCAN uses a MCP2515 CAN controller to allow CAN connections to be created and managed. The Raspberry Pi working with the PiCAN allows interaction with the simulated CAN Bus network. The PiCAN is connected by bolting onto the the Raspberry Pi through a 4 bolt screw terminal and a 40 way connector. The PiCAN is connected to the Ground (GND) and power pins of the CAN transceiver and Teensy hardware components.

3) Arduino Teensy 3.2: The CAN Bus network is simulated using an Arduino Teensy 3.2 programmed to generate fabricated ECU heartbeats that pulsated through the created CAN network. For the purpose of the exercise, the Teensy was coded and flashed using the Arduino IDE (C++) with the added extension library FlexCAN[1]. Originally written by Mathew Levett, the code was modified and simplified for the purpose of this project. The code generated CAN traffic by pulsating three ECU heartbeats through the network of the custom device, rather than the hundreds seen in a regular car. This was done to simplify the CAN traffic produced so that the exercise could be delivered more efficiently to individuals with limited automotive hacking knowledge. The Teensy is connected soldered wires to the CAN Transceiver Transmit (TX) and Receive (RX) pins and the PiCAN GND and Power pins.

4) CAN Transceiver: A CAN transceiver serves as an interface to provide successful transmission between the physical Bus network and the CAN controller. Without a transceiver, arbitration of sending and receiving CAN messages onto the physical bus would not be possible. For this project, a TJA1050 CAN Transceiver was used to achieve this – which was connected via. soldered wires to the Teensy TX and RX pins, the PiCAN CAN-H/CAN-L pins and GND/Power pins.

These four main hardware components were connected as shown in the circuit diagram in Fig. 1. The arrows are representative of the wires added and soldered to connect the different hardware components; colours have also been used to separate components for clarity - any coloured component

[1] https://github.com/collin80/FlexCAN_Library



Pin Abbreviation	Usage
CAN-H	CAN-High
CAN-L	CAN-Low
VCC	Voltage Common Collector (Power)
GND	Ground
TX	Transmit
RX	Receive
VIN	Voltage Input (Power)
JP5	Power Pin

Colour	Component/Wires
Black	Raspberry Pi 3 Model B
Blue	Teensy 3.2
Green	PiCAN
Red	CAN Transceiver

Fig. 1. Hardware components circuit diagram

notated in the diagram represents the components that form the CAN Bus network. The physical arrangement of the components is depicted in Fig 2.

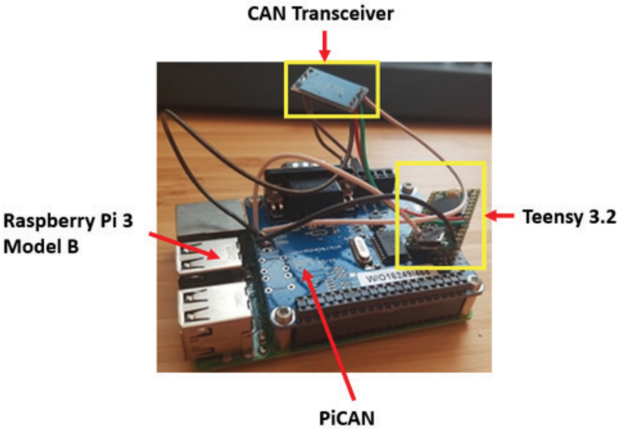


Fig. 2. Annotated hardware

B. Education Pack

Informed by the risk-based approach to identify the threats and vulnerabilities in a CAV environment, an exercise pack was developed to accompany the created hardware device to deliver an education exercise to provide a socio-technical measure in improving security awareness within automotive security. To this end, three Intended Learning Outcomes were specified, as described in Table IV.

Following these ILOs, the approach of developing the education pack and exercises was as follows. Brief introductions to CAN and ECUs were written to contextualise the education exercise in alignment to "real-life" security issues in vehicles.

Critical to the understanding and delivery of the exercises is the learners' familiarity with the terminal, the CAN message format and the `can-utils` command toolkit. Table V is an excerpt from the introduction to the CAN data and toolkit.

TABLE IV
EXAMPLE INTENDED LEARNING OUTCOMES

	Description
ILO1	Appreciate and understand the feasibility to attack a car using the Control Area Network (CAN)
ILO2	Critically understand the vulnerabilities surrounding the Control Area Network (CAN)
ILO3	Gain an understanding on basic Control Area Network (CAN) attacks

TABLE V
CAN DESCRIPTORS

Example CAN message: `can0 123 [8] 11 22 33 44 55 66 77 88`	
message item	**description**
`can0`	network interface
`123`	arbitration ID
`[8]`	data length
`11 22 33 44 55 66 77 88`	CAN data
Command toolkit:	
`candump`	dumps live traffic from CAN network
`cansend`	send CAN data to CAN network

Following the introductory phase, setup instructions were provided to allow the individual completing the exercise to initiate the virtual CAN network. This comprised of bringing up the CAN interface via. the PiCAN. For convenience, a text document was created with the needed command to setup the interface and placed on the user desktop for easy access.

Succeeding the setup phase of exercise pack, three exercises were written to demonstrate basic CAN hacking and manipulating techniques: Dumping CAN Data, Replaying CAN Messages and Injecting Fake CAN Data. The exercises utilised can-utils commands, which were specified in a "Command Toolkit" at the start of each exercise alongside a brief exercise summary specifying how the exercise demonstrates "real-world" CAN hacking techniques. Following the practical part of the exercise, several theoretical questions were written to test the understanding of the individuals completing the exercise.

To complete the exercise pack, a brief exercise summary was written to recap the exercises and align them with "real-world" CAN hacking, dictating the differences between a virtual CAN network and hurdles potentially faced practicing the newly learned skills on an actual vehicle.

V. Training implementation and Evaluation

At the core of the educational activity lie the three exercises. It should be noted that it is recommended that the exercises should be conducted in the order described in this paper, primarily because the first one also relates to testing that the hardware and device in general functions correctly. More specifically, the exercises can be carried out when the `can0` interface is correctly setup, as can be observed by the `ifconfig` command in the Raspbian operating system.

A. Exercise 1: Dumping CAN Data

This exercise primarily focused on familiarising the individual with dumping CAN data as a starting point for CAN analysis and data manipulation.

As with every exercise, a command toolkit detailing the can-utils commands needed to complete the exercise; a brief introduction is also provided to describe how the exercise was relevant in "real-life" CAN attacks. The following follow-up questions may assess the learner's comprehension of the topic:

- Why are we dumping the interface `can0`? (Answer: this is the interface that the virtual CAN network is communicating on)
- How many ECUs are there? (This should be a numerical answer and it relates to the number of simulated ECUs that are spawned on the device. An example number would be three.)

B. Exercise 2: Replay Identified CAN Data

This exercise primarily focuses on identifying CAN messages and replaying them back to the CAN network to manipulate the traffic. Replay attacks are possible in the CAN network and the duplicated messages as not distinguishable from their original. The questions asked would allow the learner to appreciate the need for timestamping, redundancy and authentication information in the CAN messages.

C. Exercise 3: Injecting Fake CAN Data

The final exercise instructs the learner to send random data to the network and is the final exercise to be tested within the pack. Slightly different from the previous exercises, a question is asked before and after the exercise is completed. The question asked before was to help provoke willful thinking surrounding the exercise – regardless, both were completed to gauge the success of the exercise:

- Do you think it is possible to send fake/random data to the network without these being detected? (Answer: yes, and this relates to the lack of authentication)
- How would one detect/see the injected data? (Answer: in the terminal using the `candump` command. However, this still does not mean that the injections can be distinguished from legitimate data)

VI. Conclusion

Combining the exercise pack and the hardware device created allows the education of low-level CAN manipulation. Mainly aimed at academia, or industry professionals that may lack knowledge of automotive CAN hacking – this education exercise was created to empower existing knowledge whilst practicing theoretical concepts that the target audience may have encountered in their research or by other means.

As part of the wider collated issues in automotive security, this artefact provides a socio-technical measure to help reduce risks associated with minimal understanding or security awareness within automotive security whilst addressing the vulnerabilities associated with CAN hacking in publicly available vehicles – focusing on a subset of vulnerabilities identified through the risk analysis exercise. For future work more capabilities can be added (such as additional sensors) to increase its attack surface and expose the learner to a wider set of vulnerabilities and attacks.

References

[1] S. Ray, W. Chen, J. Bhadra, and M. A. Al Faruque, "Extensibility in automotive security: current practice and challenges: invited," in *Proceedings - Design Automation Conference*, vol. Part 128280, (New York, NY, USA), pp. 1–6, Institute of Electrical and Electronics Engineers Inc., jun 2017.

[2] ENISA, "Good practices for security of smart cars," tech. rep., ENISA, 2019.

[3] J. Dutton, "Three pillars of cyber security," 2017.

[4] I. G. Oancea and E. Simion, "Challenges in automotive security," in *Proceedings of the 10th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2018*, Institute of Electrical and Electronics Engineers Inc., apr 2019.

[5] Lin, Tong;Chen, Luhai, "Common attacks against car infotainment systems," 2019.

[6] D. Nassi, R. Ben-Netanel, Y. Elovici, and B. Nassi, "MobilBye: attacking ADAS with camera spoofing," *arXiv*, jun 2019.

[7] F. Sagstetter, M. Lukasiewycz, S. Steinhors, M. Wolf, A. Bouard, W. R. Harris, S. Jha, T. Peyrin, A. Poschmann, and S. Chakraborty, "Security challenges in automotive hardware/software architecture design," in *Proceedings -Design, Automation and Test in Europe, DATE*, pp. 458–463, Institute of Electrical and Electronics Engineers Inc., 2013.

[8] A. Greenberg, "Hackers can steal a Tesla Model S in seconds by cloning its key fob," 2018.

[9] L. Wouters, E. Marin, T. Ashur, B. Gierlichs, and B. Preneel, "Fast, furious and insecure: passive keyless entry and start systems in modern supercars," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 3, pp. 66–85, 2019.

[10] E. Juliussen, "The future of automative telematics," *Business briefing: global automotive manufacturing & technology*, pp. 1–4, 2003.

[11] P. T. Partners, "Vehicle telematics security; getting it right," 2020.

[12] S. Halder, A. Ghosal, and M. Conti, "Secure OTA software updates in connected vehicles: a survey," *arXiv*, apr 2019.

[13] ABIResearch, "ABI research forecasts 8 million vehicles to ship with SAE Level 3, 4 and 5 autonomous technology in 2025," 2018.

[14] NCCGroup, "Automotive," 2019.

[15] Jennifer Shuttleworth, "SAE J3016 automated-driving graphic," 2019.

[16] J. Petit, "Self-driving and connected Cars: fooling Sensors and tracking drivers," tech. rep., 2015.

[17] Anshul Saxena, "How automotive LIDAR works for autonomous vehicles," 2018.

[18] McAfee, “Automotive security best bractices 1 automotive security best practices recommendations for security and privacy in the era of the next-generation car,” tech. rep., 2017.

[19] E. Hutchins, M. Cloppert, and R. Amin, “Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains,” in *6th International Conference on Information Warfare and Security, ICIW 2011*, no. July 2005, pp. 113–125, 2011.

[20] Richard Struse, “The ATT&CK™ navigator: a new open source project,” 2018.