Hack The Automotive Simulator

Setting Up a Simulation Environment for Carrying Out Hacking Attacks on CAN Bus Systems

Dirk Labudde[1], Heiko Polster[2] and Markus Straßburg[3]
Forensic Science Investigation Lab
Hochschule Mittweida – University of Applied Sciences Mittweida, Germany
Email: [1]dirk.labudde@hs-mittweida.de, [2]heiko.polster@hs-mittweida.de, [3]markus.strassburg@hs-mittweida.de

***Abstract*— The paper describes a digital simulation environment to reproduce and test attacks on Controller Area Network (CAN) bus systems. Security researchers repeatedly find vulnerabilities in different software components of networked vehicles. Since investigations on the real system seem too costly, various vehicle functions are to be tested and analysed during attacks with the help of a CAN bus simulator. This simulation environment can also be used to conduct on-site and remote training. For the simulation environment, we use the software Vector CANoe as well as CANUTILS to control and analyse the CAN bus systems. In order to be able to analyse the effects and thus the interrelationships and make them comprehensible, a CAN bus simulator was built in the form of a model car. The connection to the model is realised by means of two ESP32-EVB development boards configured as WLAN CAN gateways and a VN1610 CAN-USB interface. In the model, an AT90CAN128 development board functions as a control unit, which controls the motors and the lighting on the model car. The procedure for setting up a CAN bus simulation environment and using it to analyse and evaluate hacker attacks on automotive bus systems is described. The application possibilities show that the simulation environment can not only be used on-site, but in combination with web conferencing systems for theoretical knowledge transfer with a remote connection for solving practical tasks. It represents the most effective methodology for imparting knowledge in the field of car forensics online. Technological obstacles make it difficult to easily integrate practical tasks on real CAN bus systems into conferencing tools, as this requires a connection to the simulation hardware used. Therefore, this paper also shows how, in addition to the BigBlueButton web conferencing system, the AnyDesk remote maintenance software can be used to establish a remote connection to the control machine. Audio-visual feedback is helpful to clarify the effects of the CAN commands sent. Here, webcams are used to control the model car and a remote connection is used to enter the commands.**

Keywords-cyberattacks; car forensics; can-bus; demonstrator; remote seminar.

I. INTRODUCTION

Today's motor vehicles are no longer controlled by the driver himself using wire ropes, levers or hydraulics, but via digitally networked computer systems. A depressed brake pedal no longer necessarily means that the brakes are actually applied. In modern vehicles, the software systems decide whether this actually happens. Additionally, cars are becoming increasingly networked, both internally and in relation to the outside world. This makes it possible for hackers to locate cars from the network and penetrate their systems. In the worst case, this will make them able to take control of important control systems [21]. Meanwhile, 32% of vehicles in the US are connected to the Internet [1]. In terms of new registrations of the ten largest car brands in the US, 95% are connected cars. By 2020, the three largest manufacturers - General Motors, Toyota and Ford - which together represent almost half of the US car market, had set themselves the goal of installing hardware for connected services in every sold vehicle [2].

In recent years, more and more software vulnerabilities have been found [16]. As software is continously becoming an integral component in modern cars, especially in the area of networked services, it can be assumed that more software vulnerabilities can also be found in vehicles [1][3]. In most cases, attackers penetrate the infotainment centre of vehicles via mobile wireless connections (e.g. for location or emergency call systems), where most security vulnerabilities can be found [2]. From there, it is sometimes possible to penetrate the control electronics. The biggest challenge here is to send vehicle data to the infotainment system without allowing data flow in the other direction [19][20].In this context, the question arises as to what data is actually stored and transmitted in the vehicle? This can be location data, stored routes, telephone data, error messages, time stamps, kilometre statuses or even exact parking locations of a vehicle at a defined point in time [19].

This raises additional questions: which electronic systems are installed in the vehicle under consideration, which interfaces do the various systems have, how can these systems be addressed or evaluated forensically, which hacking and analysis tools enable data evaluation? Furthermore, data is sent to the respective backend with the help of manufacturer-specific online or remote services. This data is not only of great interest to authorities, insurance companies and service providers, but can also help in investigations. The derived information can be assigned to a crime and thus, if necessary, to a person. "Forensic is related to scientific methods of solving crimes, involving examining the objects or substances that are involved in the crime." [22].

 ISBN: 978-1-61208-879-2

The purpose of car forensics is to assist in solving crimes and to find data in vehicle systems that provide conclusions about the use or manipulation of the vehicles.

The motor vehicle and the automotive industry are undergoing a noticeable transformation: electric drives, autonomous driving and smart mobility require comprehensive networking of the vehicle with its environment. According to estimates, there will be 775 million connected vehicles in 2023, which will be connected by means of in-vehicle telematics or via a smartphone app [3]. The number of cyberattacks on motor vehicles is steadily increasing. For example, 73 attacks on vehicles were recorded in the whole of 2018 and 71 vehicles have been the victims of attacks in the first four months of 2019 [4].

Due to the considerable increase in possible attack vectors resulting from the networking of vehicles and the serious security deficiencies in the implementation of the CAN bus protocol, a subdivision must be made based on the access possibilities [17]. For this purpose, the following classification was established for the differentiation of attacks: direct, short-range and longe-range. It matters how these can be carried out by the attacker. Figure 1 shows a summary of the most frequent attack targets.

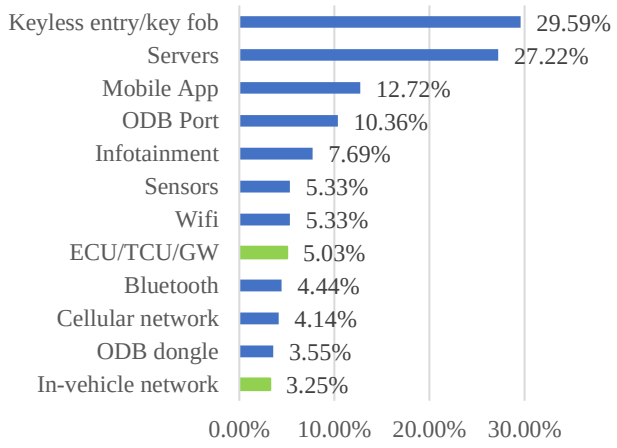

Figure 1. Distribution of attack targets on motor vehicles in descending order of occurrence. Green are the attack targets on which this paper focuses. [5]

In this work, attacks targeting Electronic Control Units (ECU) and in-vehicle network are specifically addressed. However, the represented simulation environment shows the opportunities to address further attack targets. The distance between the attacker and the vehicle plays a decisive role in the attack targets. Therefore, the attacks are classified according to the distance between the attacker and the vehicle. Direct attacks require direct physical access. The attacker must expose himself to the danger of implementing his attack directly, even if sometimes only briefly, on the vehicle itself. In addition, an attacker cannot determine the timing of the attack himself, but is rather dependent on the behaviour of the vehicle owner. This further limits the potential for direct attacks. Short-range attacks, on the other hand, allow the attacker to keep a certain distance from the vehicle by compromising those interfaces of the vehicle that operate wirelessly within a radius (from a few metres for Bluetooth and Tyre Pressure Monitoring System (TPMS) to approx. 100m for WLAN or remote opening systems). Long-range attacks allow the attacker to connect to the vehicle from any point, via the Internet or the mobile network, and carry out an attack. Since the attacker no longer needs to be in the immediate vicinity of a vehicle to be attacked, this greatly increases the chances that a vehicle can be attacked (see Figure 2). Thus, short- and long-range attacks can be carried out remotely. In 2019, 82% of all attacks on motor vehicles were carried out remotely (short-range and long-range) [5]. The number of attacks on motor vehicles increased significantly over the years from 2013 to 2019.

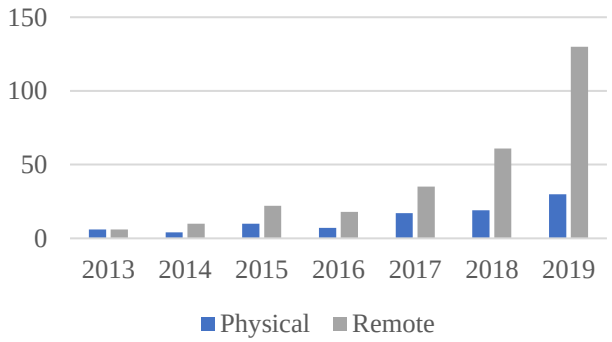

Figure 2. Comparison between physical and remote attacks [4]

Due to cost considerations and the dangers of manipulating vehicle systems during real-world use, a simulator offers a viable alternative to conducting hacking attacks on automotive bus systems. CAN bus simulation provides a safe way to analyse what-if scenarios. For example, the effects of sending CAN bus commands via the On-board Diagnostics (OBD) socket can be tested.

This work focuses on the attack targets of in-vehicle network and ECU. Therefore, direct attacks on automotive systems are primarily carried out and analysed. Dongles on the CAN interface are simulated, which can be used to access CAN bus systems and manipulate their data. In this way, direct attack vectors can be reproduced. In order to be able to use the simulator in the context of remote training, e-learning possibilities are discussed and an environment is described in which participants can access a CAN bus simulator via a remote connection and realise direct connections to this system. A major disadvantage with regard to remote seminars is the lack of feedback for participants. For this reason, this paper additionally describes possibilities to test the effects of CAN commands on a model car using a real CAN interface. The simulation environment was further equipped with cameras that give the participants remote visual feedback on the effects of their CAN commands. To enable several participants to work together on a task, a solution is described that enables remote group work and exchange with experts.

Section II presents the necessary tools that are used to implement the simulation environment. Section III describes

 ISBN: 978-1-61208-879-2

the use case and the concrete structure of the simulator. A conclusion and the outlook can be found in Section IV.

II. TOOLS FOR CAN-SIMULATION

A. Vector CANoe

When developing new control units for motor vehicles, it is possible to simulate CAN bus systems using professional simulation environments. One of the tools used for this is CANoe from Vector [35]. In this tool, virtual ECUs can be tested in real CAN bus environments via a hardware interface, e.g. Vector VN1610. Within the environment, all valid CAN messages for the respective CAN system are stored in a database. These messages can be sent and received in the simulation by means of interactive generators or by virtual ECUs. The virtual ECUs are equipped with their special functionality using Communication Access Programming Language (CAPL). This type of programming is event-oriented. The control of the environment is realised with panels in which corresponding control elements, such as buttons, switches, visual displays, etc., are used. Figure 3 shows a CAN bus simulation of a virtual motor vehicle.

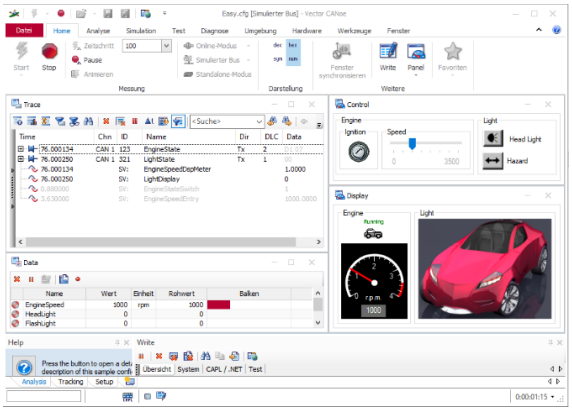

Figure 3. Exemplary representation of a CAN-Simulation with tool CANoe

These controls can be linked to system variables or even message content. Extensive analysis tools are available within the environment. Because of the possibility to freely program the virtual control units and the physical connection to real CAN bus systems, the operation of the CAN bus system can be used for the simulation of an attack.

B. CANUTILS

CANUTILS are available within Linux systems. With this toolset, it is possible to control and analyse virtual and real CAN bus systems. Figure 4 shows an example of an intersection of CAN traffic with CAN identifiers and corresponding data content.

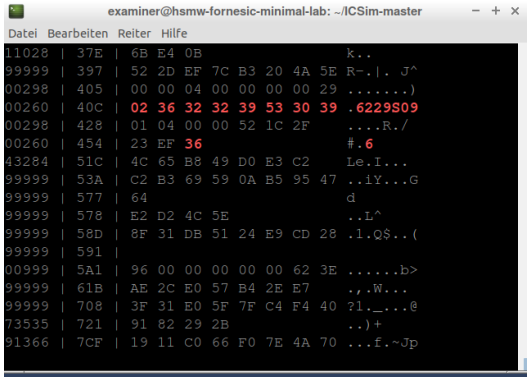

Figure 4. Recoding of CAN traffic using cansniffer from the CANUTILS

The following tools are integrated in the CANUTILS:

- candump: with candump CAN data can be displayed, filtered, logged and saved in log files.
- canplayer: CAN log files can be played back with canplayer.
- cansend: this tool enables the transmission of individual CAN frames.
- cangen: generates random CAN traffic.
- cansniffer: with the cansniffer, the traffic on the bus can be displayed live. In addition, it is possible to filter out individual CAN frames from the data stream

Within the simulation environment, the CANUTILS take care of the analysis of the CAN traffic.

1) Linux with CAN

A prerequisite for working with CAN systems under Linux is the installation of CANUTILS. To connect real CAN systems to the computer, a USB CAN interface is necessary. For example, the PCAN-USB [7] interface from Peak can be used for this. The installation of the corresponding drivers is necessary for use. Further information on this can be found under [25].

2) Red Pitaya

Red Pitaya STEMLAB is a grid-connected Field Programmable Gate Array (FPGA)-based test and measurement board. It can replace many measurement devices in an electronics lab by working as an oscilloscope, spectrum analyser, LCR meter (inductance, capacity, resistance), network analyser or other test and measurement application. Open source software is used and the operating system is based on Linux. Figure 5 shows the Red Pitaya hardware. An FPGA from Xilinx and an ARM Cortex-A processor are implemented on the board. The latter has a CAN interface on the hardware side, which is not activated in the basic configuration. However, this can be realised by means of devmem via the following commands:

- devmem2 0xf8000728 w 0x1221
- devmem2 0xf800072c w 0x1220

 ISBN: 978-1-61208-879-2

Figure 5. Red Pitaya hardware, which provides the backend for generating and recording CAN signals

After that, the pins and CANTX (header E2 pins 3 and 4) of the CAN interface are available on the Red Pitaya board. A CAN transceiver (e.g. Microchip MCP2551) must be connected to these pins to generate the differential CAN signals. To use the CAN interface, the CANUTILS can also be used in this system, since the Red Pitaya works with a Linux system. With a self-implemented application, the CAN interface can be used after installing the CANUTILS. Figure 6 shows an example of the data exchange between the Red Pitaya hardware and the frontend on a personal computer.

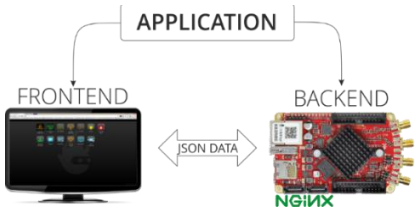

Figure 6. Scheme of data exchange between frontend and backend on the Red Pitaya System [8]

The Red Pitaya Board is connected to a computer via a network interface. The frontend is accessible via a browser with a configured IP address. Figure 7 shows the graphical frontend of the developed Can Sniffer application, which is used to analyse CAN traffic and send individual CAN messages.

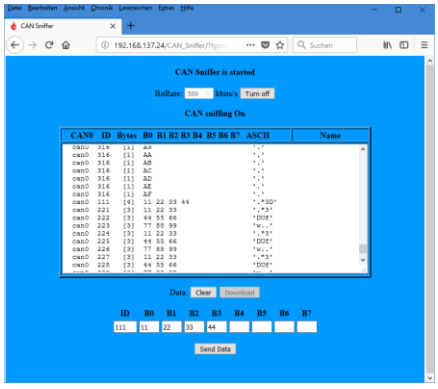

Figure 7. Recording of CAN traffic in the self-created interface for the frontend of the Red Pitaya Systems

The application itself is installed on the RedPitaya system and can be started via the frontend. Figure 7 shows a specially created interface for visualising CAN traffic on the CAN interface of the RedPitaya system. This interface communicates via the network with the RedPitaya application "CAN Sniffer". Via this interface, it is possible to analyse and generate traffic in the CAN simulator. Further information on the RedPitaya system can be found under the following: [13][14].

C. *Bluetooth OBD-Dongle*

Private users can purchase various OBD Bluetooth adapters for vehicle diagnosis on the Internet. These are largely based on the OBD2 interpreter chip ELM327 [26]. This chip includes, among other things, communication via CAN with 11-bit and 29-bit identifier and supports the ISO 15765-4 protocols with 500 kbps and 250 kbps.

Figure 8. Exemplary representation of a common OBD dongle [9]

Figure 8 shows an OBD interface with Bluetooth interface, which makes vehicle data available with a smartphone application. These adapters can communicate with corresponding Android apps, for example to display current data on speed, acceleration, cooling water temperature, etc. in live operation. The technical description of the ELM327 can be found on the Elmelectronics website [11]. In this data sheet, the configuration variants of the chip are described, whereby one variant makes it possible to configure own baud rates for the interface and to send own CAN identifier with own data contents on a CAN bus. The ELM327 itself enables communication via a virtual RS-232 (Recommended Standard 232), which is available on a computer via Bluetooth. Thus, the chip can be configured and controlled with a console application (Putty) and the AT commands described in the data sheet.

D. *Microcontroller*

Another variant of CAN communication can be seen in the use of microcontrollers with an integrated CAN interface. Preconfigured hardware can be purchased, for example, from Olimex. The AT90CAN128 development board, shown in Figure 9, is presented here as an example.

Figure 9. AT90CAN128 development board used to generate CAN messages [10]

The central element on the board is the microcontroller from Microchip ATMEL AT90CAN128. In addition to 128 kB of programmable memory and 4 kB of RAM, this controller offers a hardware-integrated CAN interface that is available to the user via a CAN transceiver MCP2551 and a Sub-D 9. Software libraries for controlling the CAN interface are available from a community [27]. By means of these libraries, it is possible to realise flexible configurations with regard to baud rates, CAN identifier and data contents. Firmware for the microcontroller can be created using Microchip Studio [34], which has an integrated C compiler. A JTAG interface is available on the development board as a programming interface, which can be controlled via the ATMEL ICE programmer [12]. The simulator uses this to generate its own messages on the CAN bus.

CANoe, PCAN-USB, CANUTILS and the AT90CAN128 development board with Microchip Studio were selected for practical use in the seminars. The decision to use these presented tools was made in order to achieve the fastest possible learning success for the seminar participants. The participants should be able to take part in the seminar without extensive programming knowledge. Seminar participants can thus quickly find their way around and only need one software for various functionalities. Furthermore, the CANoe tool is a standard tool in the automotive industry; the use of this tool should ensure practice-oriented teaching. This also results from the possibility of using these tools in a goal-oriented manner with regard to the subtasks to be solved in the seminar without major overhead for the participants. The aims of the seminar are:

- Creation of a CAN bus simulation,
- Analysis of CAN messages,
- Control of a CAN demonstrator and
- Compromising a CAN bus system.

CANoe is also a standard tool for the development of automotive bus systems. In addition to the training version used in the seminar, a demo version with a limited range of functions is available. In this version, for example, no real hardware can be controlled. The advantages of the training version are the possibility of connecting real CAN bus systems, the diverse possibilities for visualising control interfaces and the extensive analysis tools in the tool itself. The tool thus combines various functionalities that could otherwise only be used with different tools.

There are alternative adapters for the PCAN-USB adapter, e.g. USBtin [31] from the open source world or the isCAN USB from Thorsis Technologies GmbH [32], which is available for a fee. PCAN-USB was used in other projects and was therefore available. However, it will be examined later to use the USBtin adapter as an alternative solution.

The Microchip Studio is a proprietary Integrated Development Environment (IDE) for Atmel microcontrollers. It includes the AVR Gnu C compiler (AVRGCC) and is provided free of charge by Microchip.

The CANUTILS under Linux are open source and free to use. AVRStudio is an integrated C compiler and a graphical debugger.

An alternative solution for CANoe is the tool PCAN-Developer 4 from Peak [33], which is also available for a fee.

For programming Atmel microcontrollers, a number of commercial IDEs are available for a fee. These are, for example, IAR Embedded Workbench for AVR, JumpStart C for AVR from Image Craft or CodeVisionAVR from HP InfoTech.

The Red Pitaya system and the Bluetooth-OBD adapter are not used in the seminar. Both systems are only intended to show the possibility of compromising CAN bus systems at this point. They are currently to be classified as experimental status with a limited range of functions and thus cannot be used in a seminar.

III. SIMULATIONS ENVIRONMENT AND USE CASES

Challenges for the simulation environment were the integration of the different systems into the seminar, as these are to be used under the two operating systems Linux and Windows 10. Windows is used to run CANoe, while the ICSIM-VM is based on a Linux system. In addition, there was the challenge of being able to make the seminar, which was actually planned as a face-to-face seminar, available online as well. This section also looks at the effectiveness of delivery formats in e-learning and considers the scenario in which the simulator can be used. Furthermore, a form of live workshop is described, which can use special hardware in remote seminars to solve practical tasks with it. Private and official investigators are particularly conceivable target groups.

First, various mediation formats are considered. The National Training Laboratories Institute for Applied Behavioural Science describes a learning pyramid on this topic, which shows the learning effect depending on the delivery format. The delivery formats range from a lecture to teaching other people. Figure 10 illustrates the learning pyramid [6]. In addition to the learning pyramid, there are a number of approaches to describing the learning process. David Kolb, for example, defines four different learning types and learning phases. In the first phase, practical experience is gained. The second phase involves mental observation and reflection. In the third phase, theory is introduced and problems are defined. Finally, in the fourth phase, the learned knowledge is tested for practical suitability. A solution approach is to be found through active trial and error in order to support the learning effect [28].

Furthermore, the importance of practical application and its feedback can be described with Phil Race's learning model. The five most important factors for successful learning according to Phil Race are as follows:

1. want motivation, interest, enthusiasm
2. need, survival, saving face
3. practice, repetition, experience, trial and error.
4. feedback on other people's reactions to see the results.
5. it makes sense to deal with what you have learned. [29]

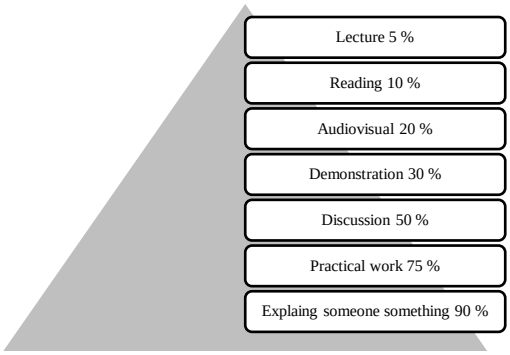

Figure 10. Learning pyramid from National Training Laboratories [6]

With the CAN bus simulator, it is possible to train the application of the learned knowledge on the topic of car forensics and CAN bus systems by means of practical exercises in addition to the classic lecture and discussions among the learners. The simulator can be used in physical presence as well as in online seminars. Remote live workshops offer users the opportunity to work on extensive tasks and solve complex issues. Furthermore, these workshops are characterised by the fact that different teaching formats are used. In addition to the live lecture to convey the theoretical content, the combination of meeting tools and remote connection options such as BigBlueButton and Remote Desktop Connections allow practical exercises to be carried out and the knowledge learned to be applied. In order to be able to expand practical modules in teaching with presence content, the transfer of the practical applications and exercises with an interface into the virtual environment is necessary.

The real learning environment of the module is composed of stationary computer systems with a virtual machine, a development environment for CAN microcontrollers and special simulation software.

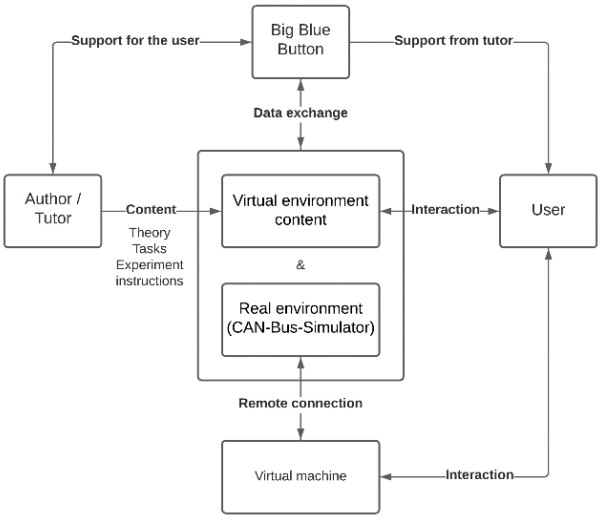

Figure 11. Structure of the virtual and real learning environment, for conducting remote seminars in the field of automotive CAN bus systems.

The target group of the forensics seminars has the possibility to follow implemented tasks on the simulation software via web access. This creates a link between the virtual and the real environment between the learner and the development environment. Support is provided by the tutor, who is on site in the learning lab and can interact with the participants via a conference tool. BigBlueButton was used as the conference tool in this setup. Furthermore, the learners receive the theoretical content via the conference tool. Figure 11 shows an approach in which a virtual machine is accessed via a remote connection. Using this virtual machine, the participants can send CAN commands to the model vehicle. The CAN commands are distributed by the CAN gateway to the individual ECUs and implemented by the hardware. The learner can observe the result of the output via a webcam and thus obtain a direct control of success.

The AnyDesk software is particularly suitable, as different participants can access a desktop at the same time and thus work on a task as a group [36]. Alternatively, the remote software TeamViewer with the same functionality can be used. The remote desktop under Windows is not suitable, as only one user can log in and the on-site tutor has no access to the system. The need for at least two seminar participants and the lecturer to access a desktop is justified by the realisation of group work in the online seminar and the support of the individual groups by the lecturer.

A Linux environment is absolutely necessary for the Instrumentation Cluster Simulator (ICSIM). For this purpose, a Virtual Machine (VM) in Oracle VitualBox was provided for the seminar. In this VM, the drivers for the PCAN-USB adapter, the CANUTILS and the ICSIM are pre-installed to realise a quick use of the tools. This VM can be controlled by the participants via the remote software. The Instrumentation Cluster Simulator (ICSIM) software, shown in Figure 12, is used as a simulation environment for the automotive CAN bus [15]. This simulator is used to display a virtual dashboard with speedometer, indicators and virtual doors, which can be controlled via a control interface.

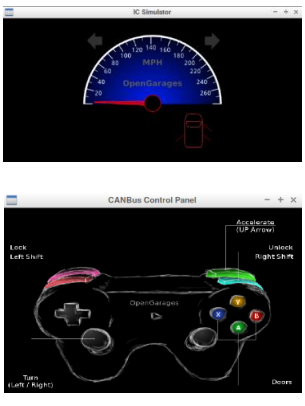

Figure 12. Dashboard simulation for graphical representation of the transmitted CAN commands and control interface for transmitting CAN commands

The simulator can be run under Linux. Within the Linux environment, the CANUTILS are used. The CAN bus is configured in the first part as a virtual CAN bus with the following commands:

- sudo modprobe can
- sudo modprobe vcan
- sudo ip link add dev vcan0 type vcan
- sudo ip link set up vcan0

The respective components of the simulator and the analysis software cansniffer are configured with the commands:

- ./icsim vcan0
- ./controls vcan0
- cansniffer -c vcan0

In the practical exercise, the development board AT90CAN128 from Olimex is connected to this PCAN-USB interface. At the start of the exercises, the firmware of the development board is programmed with a prefabricated firmware, which realises the control of the tachometer in the simulator dashboard. In the further course of the practical exercises, the prefabricated firmware is to be extended and tested with the CAN messages that were determined in the first part. These additional CAN messages can then be used, for example, to control the turn signals and also the doors in the simulation environment via the microcontroller board. Microchip Studio is used as the development environment at this point. This environment is used in the Windows environment of the practical computer, as it is easy and clear to create and debug the microcontroller firmware.

In principle, the Linux environment with integrated CAN simulator can be run independently on a computer as a ready-made virtual machine for VirtualBox or VMware. The computer is made available to the user for an online practical course via remote software. This could be done using Windows Remote Desktop, but this way only one user is able to log in to the computer. However, the aim of the practical session is to enable group work consisting of at least 2 users, whereby the instructor on site should also have access to the desktop in order to be able to monitor the work of the respective groups. For this purpose, Teamviewer and Anydesk were evaluated as remote applications. Both applications enable the realisation of the described online learning scenario.

To make the practical experience of CAN bus programming more tangible, a CAN bus demonstrator in the form of a remote-controlled model vehicle is used. The model car was developed for this simulation environment based on the Robot Car Kit from Joy-It. Figure 13 demonstrates the vehicle.

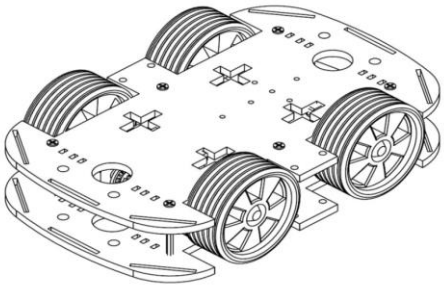

Figure 13. Schematic representation of the Robot Car Kit from Joy-It. [30]

The basic framework for the realisation is the Robot Car Kit from Joy-It including four electric motors. In addition, there is an EVAL6207N evaluation board from STMicroelectronics. Since the demonstrator is to be equipped with a CAN bus, another microcontroller that can handle CAN is needed. With the AVR CAN board from Olimex, the EVAL6207N can receive signals for control. Headlights and turn signals that mimic those of a vehicle are to be added to the setup. The demonstrator is controlled by CANalyzer software, which is used to generate appropriate CAN messages. To transmit messages from a computer with the software to the control unit (AVRCAN board) of the demonstrator, the CAN interface VN1610 from Vector is required. The actual CAN bus exists between the CAN interface and the AVR CAN board. Figure 14 schematically represents the basic structure of the CAN demonstrator.

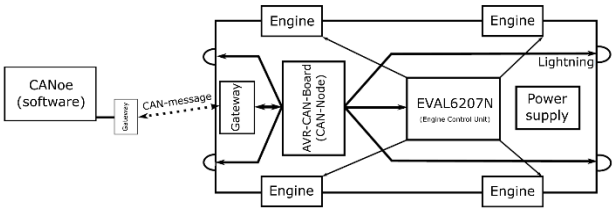

Figure 14. Schematic representation of the complete CAN-Demonstrator

Using the CANoe tool, CAN commands can be sent to the demonstrator via the gateway on a specially created interface. After being received by the gateway, the CAN bus commands are forwarded by the AVR CAN board to the respective control units and implemented there. Figure 15 demonstrates the panel for the experimental setup. Figure 16 shows the complete CAN demonstrator hardware.

Figure 15. Interface to control the CAN-Demonstrators inside the CANoe-Tool

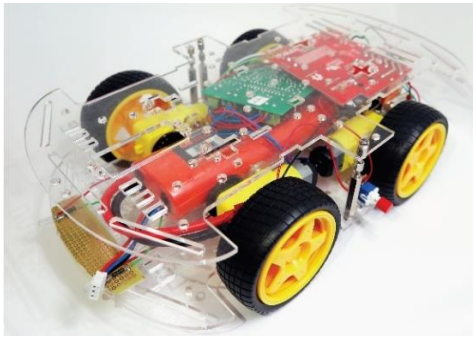

Figure 16. Complete model of the CAN-Demonstrators

IV. CONCLUSION

In this paper, it was shown that a real CAN system can be simulated with the help of the presented tools. In addition, it is possible to use this simulation environment to analyse and carry out cyberattacks on vehicle CAN bus systems. A methodology and a simulation environment were described which, in addition to the classical lecture and audiovisual media, also integrate the practical part with visual feedback. The simulation environment offers the possibility to manipulate the CAN bus of the model car from the outside. Students of general and digital forensics have already been able to test this setup. The feedback was consistently positive. In particular, solving practical tasks on the real CAN bus demonstrator was mentioned positively and motivated the participants to come up with creative solutions, despite the technological obstacles, when integrating it into an online learning environment. In addition to using this in the field of car forensics, it is conceivable that other subject areas can be applied with special hardware. Use in the area of IoT (Internet of Things) forensics is conceivable here. Furthermore, it is possible to use the setup to realise more attack vectors on automotive systems. For example, the CAN bus demonstrator can be equipped with a keyless go system, which would allow replay station attacks to be realised [18]. Additionally, it would be possible to implement an intrusion detection system to filter out malicious messages in the data stream of the CAN bus. With the solution shown, challenges could be solved to be able to integrate Windows and Linux tools in an online seminar.

REFERENCES

[1] The 2020 Digital Auto Report, PwC Network, last access date: 21.05.2021, online available from: https://www.strategyand.pwc.com/de/de/studie/2020/digital-auto-report-2020.html

[2] Consumer WatchDog, “Kill Switch – Why connected cars can be killing machines and how to turn them off“, last access date: 21.05.2021, online available from: https://www.consumerwatchdog.org/sites/default/files/2019-07/KILL%20SWITCH%20%207-29-19_0.pdf

[3] B. Sam, “Consumer Connected Cars: Telematics, In-vehicle Apps & Connected Car Commerce 2018-2023.” Hg. v. Juniper Research Ltd., last access date: 21.05.2021, online available from: https://www.juniperresearch.com/press/press-releases/in-vehicle-commerce-opportunities-exceed-775mn, 2018.

[4] J. Bird, “Car hacking threatens vision of connected mobility”. Hg. v. Financial Times LTD (The Future of the Car), last access date: 21.05.2021, online available from: https://www.ft.com/content/163f08c6-6ce3-11e9-9ff9-8c855179f1c4, 2019.

[5] Upstream, “Upstream Security's 2020 Global Automotive Cybersecurity Report.” Hg. v. Upstream Security Ltd. Last access date: 21.05.2021, online available from: https://www.upstream.auto/upstream-security-global-automotive-cybersecurity-report-2020/, 2019.

[6] J. Lalley and R. Miller “The learning pyramid: does itpoint teachers in the right direction.” Education, vol. 128, no.1, pp. 64-79, 2007.

[7] PEAK-Systems PCAN, last access date: 21.05.2021, online available from: https://www.peak-system.com/typo3temp/pics/61ef9f60a0.jpg

[8] Red Pitaya System overview, last access date: 21.05.2021, online available from: https://redpitaya.readthedocs.io/en/latest/developerGuide/software/sysOver.html

[9] OBDII Interface Lescars, last access date: 21.05.2021, online available from: https://images-na.ssl-images-amazon.com/images/I/811dSmBKHfL._AC_SL1300_.jpg

[10] OLIMEX Development Board, last access date: 21.05.2021, online available from: https://www.olimex.com/Products/AVR/Development/AVR-CAN/images/thumbs/310x230/AVR-CAN-01.jpg

[11] Elmelectronis ELM327DS, last access date: 21.05.2021, online available from: https://www.elmelectronics.com/wp-content/uploads/2016/07/ELM327DS.pdf

[12] ATMEL ICE, last access date: 21.05.2021, online available from: https://www.microchip.com/DevelopmentTools/ProductDetails/ATATMEL-ICE

[13] RedPitaya Docs, last access date: 21.05.2021, online available from: https://redpitaya.readthedocs.io/en/latest/

[14] RedPitaya DevGuide, last access date: 21.05.2021, online available from: https://redpitaya.readthedocs.io/en/latest/developerGuide/devGuideTop.html

[15] Instrumentation Cluster Simulator (ICSIM), last access date: 21.05.2021, online available from: https://github.com/zombieCraig/ICSim

[16] Number of common IT security vulnerabilities and exposures (CVEs) worldwide from 2009 to 2019, last access date: 21.05.2021, online available from: https://www.cvedetails.com/browse-by-date.php

[17] R. Buttigieg, M. Farrugia and C. Meli, “Security Issues in Controller Area Networks in Automobiles”, 2017.

[18] A. Francillon, B. Danev and S. Capkun, “Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars”, IACR Cryptology ePrint Archive. 2010. p. 332., 2010.

[19] K. Koscher et al., “Experimental Security Analysis of a Modern Vehicle.” Hg. v. IEEE. 2010 IEEE Symposium on Security and Privacy. Berkeley/Oakland, 2010

[20] F. Maggi, “A Vulnerability in Modern Automotive Standards and How We Exploited” It. Hg. v. TrendLabs Security Intelligence Blog. TREND MICRO. Bonn, last access date: 21.05.2021, online available from: https://documents.trendmicro.com/assets/A-Vulnerability-in-Modern-Automotive-Standards-and-How-We-Exploited-It.pdf, 2017.

[21] C. Miller and C. Valasek, “Remote Exploitation of an Unaltered Passenger Vehicle.” , last access date: 21.05.2021,

online available from: http://illmatics.com/Remote%20Car%20Hacking.pdf., 2015.

[22] Cambridge Dictionary – forensic , last access date: 21.05.2021, online available from: https://dictionary.cambridge.org/dictionary/english/forensic

[23] S. Checkoway et al., "Comprehensive experimental analyses of automotive attack surfaces", Proceedings of the 2011 Usenix Security, 2011.

[24] K. Koscher et al., "Experimental security analysis of a modern automobile.", in IEEE Symposium on Security and Privacy, pp. 447–462, 2010.

[25] Peaks Systems Driver, last access date: 21.05.2021, online available from: https://www.peak-system.com/fileadmin/media/linux/index.htm

[26] ODB ELM327, last access date: 21.05.2021, online available from: https://www.obd-2.de/elm327.html

[27] Mikrocontroller Community, last access date: 21.05.2021, online available from: https://www.mikrocontroller.net/topic/98697

[28] D. Kolb, "Experiential Learning: Experience as the Source of Learning and Development." New Jersey: Prentice Hall, 1984.

[29] P. Race, "Learning in small groups", last access date: 21.05.2021, online available from: https://www.heacademy.ac.uk/system/files/resources/id475_learning_in_small_groups_Race.pdf

[30] Joy-It Robot Car Kit 01, last access date: 21.05.2021, online available from: https://cdn-reichelt.de/documents/datenblatt/A300/ANLEITUNGROBOT03.pdf

[31] T. FIschl, "USBtin - USB to CAN interface", last access date: 21.05.2021, online available from: https://www.fischl.de/usbtin/

[32] CAN USB Adapter – isCAN USB, last access date: 21.05.2021, online available from: https://www.thorsis.com/de/can/can-usb-adapter-iscan-usb/

[33] PCAN-Developer 4, last access date: 21.05.2021, online available from: https://www.peak-system.com/PCAN-Developer-4.461.0.html

[34] Microchip Studio for AVR® and SAM Devices, last access date: 21.05.2021, online available from: https://www.microchip.com/en-us/development-tools-tools-and-software/microchip-studio-for-avr-and-sam-devices

[35] Vector CANoe, last access date: 21.05.2021, online available from: https://www.vector.com/int/en/products/products-a-z/software/canoe/

[36] AnyDesk, last access date: 21.05.2021, online available from: https://anydesk.com/en