A Cybersecurity Education Platform for Automotive Penetration Testing

Philipp Fuxen†, Stefan Schönhärl*, Jonas Schmidt†, Mathias Gerstner*, Sabrina Jahn*, Julian Graf†, Rudolf Hackenberg† and Jürgen Mottok*

†*Department of Computer Science and Mathematics*
Ostbayerische Technische Hochschule
Regensburg, Germany

**Department of Electrical Engineering and Information Technology*
Ostbayerische Technische Hochschule
Regensburg, Germany

Email:{stefan1.schoenhaerl, mathias.gerstner}@st.oth-regensburg.de
Email:{philipp.fuxen, sabrina.jahn, jonas.schmidt, julian.graf, rudolf.hackenberg, juergen.mottok}@oth-regensburg.de

***Abstract*—The paper presents a penetration testing framework for automotive IT security education and evaluates its realization. The automotive sector is changing due to automated driving functions, connected vehicles, and electric vehicles. This development also creates new and more critical vulnerabilities. This paper addresses a possible countermeasure, automotive IT security education. Some existing solutions are evaluated and compared with the created Automotive Penetration Testing Education Platform (APTEP) framework. In addition, the APTEP architecture is described. It consists of three layers representing different attack points of a vehicle. The realization of the APTEP is a hardware case and a virtual platform referred to as the Automotive Network Security Case (ANSKo). The hardware case contains emulated control units and different communication protocols. The virtual platform uses Docker containers to provide a similar experience over the internet. Both offer two kinds of challenges. The first introduces users to a specific interface, while the second combines multiple interfaces, to a complex and realistic challenge. This concept is based on modern didactic theories, such as constructivism and problem-based/challenge-based learning. Computer Science students from the Ostbayerische Technische Hochschule (OTH) Regensburg experienced the challenges as part of a elective subject. In an online survey evaluated in this paper, they gave positive feedback. Also, a part of the evaluation is the mapping of the ANSKo and the maturity levels in the Software Assurance Maturity Model (SAMM) practice Education & Guidance as well as the SAMM practice Security Testing. The scientific contribution of this paper is to present an APTEP, a corresponding learning concept and an evaluation method.**

***Keywords*—IT-Security Education; Automotive; Penetration Testing; Education Framework; Challenge-based Learning.**

I. Introduction

The following paper is an extended paper of the Thirteenth International Conference on Cloud Computing, GRIDs, and Virtualization contribution [1].

Automotive security is becoming increasingly important. While Original Equipment Manufacturer (OEM)s have developed vehicles for a long time with safety as a central viewpoint, security only in recent years started becoming more than an afterthought. This can be explained by bringing to mind those historic vehicles that used to be mainly mechanical products. With the rising digitalization of vehicles, however, the circumstances have changed.

Recent security vulnerabilities based on web or cloud computing services, such as Log4j, can be seen as entry points into vehicles, which an attacker can use to cause significant harm to the vehicle or people. To combat this, the development and release of new standards are necessary. The International Organization for Standardization (ISO) 21434 standard [2] and United Nations Economic Commission for Europe (UNECE) WP.29 [3], manifest the importance of automotive security in recent years. They require OEMs to consider security over a vehicle's whole life cycle.

However, there are different ways in which automotive security can be improved. Jean-Claude Laprie defines means of attaining dependability and security in a computer system, one of these being fault prevention, which means to prevent the occurrence or introduction of faults [4]. This can be accomplished by educating current and future automotive software developers. Since vulnerabilities are often not caused by systemic issues, but rather by programmers making mistakes, teaching them about common vulnerabilities and attack vectors can improve security. Former research shows furthermore that hands-on learning not only improves the learning experience of participants but also increases their knowledge lastingly. Therefore, a framework for IT-security education has been developed, APTEP, which was derived from penetration tests on modern vehicles.

The ANSKo was developed as an implementation of this framework with the focus on needed competencies and skill sets of penetration testers, e.g., [5], [6], like network knowledge, hardware knowledge, and information gathering. It is a hardware case, in which communicating Electronic Control Units (ECUs) are simulated, while their software contains deliberately placed vulnerabilities. In the first step, users are introduced to each vulnerability, before being tasked with

exploiting them themselves.

The ANSKo was integrated into a problem-based/challenge-based learning environment for teaching automotive security and penetration testing concepts in academic education. Computer science students of the OTH Regensburg were able to work with the ANSKo as part of an elective course for the 6th & 7th semesters. The course resulted in participants gaining a deeper understanding of security and penetration testing in the automotive context.

This paper aims at establishing a realistic and effective learning platform for automotive security education. Therefore, the following research questions are answered:

- (RQ1) - Which Educational Design is appropriate for Security Education for IT Students?
- (RQ2) - What content is appropriate for an automotive penetration testing framework for IT security education?
- (RQ3) - How could an automotive security education platform be implemented?

The structure of the paper starts with the related work in Section II. Section III introduces an architecture derived from modern vehicle technologies. Those technologies are then classified into layers and briefly explained in Section IV. The structure and used software of the ANSKo itself are presented in Section V. Section VI presents the learning concept and its roots in education theory. After that Section VII gives an overview of the implemented challenges and describes one in detail. The penultimate Section VIII deals with the evaluation. The paper ends with a conclusion and future work in Section IX.

II. Related Work

The demand for an automotive security dedicated learning platform arises from a large number of vulnerabilities that have become known in recent years. As vehicles become increasingly connected, the risks of these vulnerabilities also continue to increase. In addition, the complexity is also growing. Classic slide-based learning approaches for automotive IT security are no longer sufficient. More innovative and constructive learning concepts are needed. Since the automotive security education has different aspects to be considered, this section is split into three parts.

A. Work on other educational frameworks

Table I compares different hands-on security learning platforms based on specified criteria. The table also shows some of the main objectives of APTEP. Hack The Box (HTB) is a hands-on learning platform with several vulnerable virtual systems that can be attacked by the user. Thereby, a big focus of this platform is gamification. They do not offer automotive-specific systems and access to physical hardware is also not possible [7].

One approach that focuses on hardware-specific attacks is the Hardware Hacking Security Education Platform (HaHa SEP). It provides practical exploitation of a variety of hardware-based attacks on computer systems. The focus of HaHa SEP is on hardware security rather than automotive security. Students who are not present in the classroom can participate via an online course. A virtual version of the hardware cannot be used [8].

TABLE I
Comparison of the different approaches

	HTB	HaHa SEP	RAMN	APTEP
Virtual approach	YES	NO	NO	YES
Hardware approach	NO	YES	YES	YES
Automotive specific	NO	NO	YES	YES
Gamification	YES	NO	NO	YES
IT-Security	YES	YES	YES	YES

The Resistant Automotive Miniature Network (RAMN) includes automotive and hardware-related functions. The hardware is very abstract and is located on a credit card-sized Printed Circuit Board (PCB). It provides closed-loop simulation with the CARLA simulator but there is no way to use RAMN virtually. The focus of RAMN is to provide a testbed that can be used for education or research. However, it is not a pure education platform [9].

Another differentiation from ANSKo are the cybersecurity awareness platforms. One example from the industrial environment is the SiFu platform. One focus here is on training software developers to comply with the guidelines for secure coding [10].

B. Attacks in the automotive domain

The fundamental and related work for the APTEP are real-world attack patterns. The technologies used for connected vehicles represent a particularly serious entry point into the vehicle, as no physical access is required. Once the attacker has gained access to the vehicle, he will attempt to penetrate further into the vehicle network until he reaches his goal. This can be done with a variety of goals in mind, such as stealing data, stealing the vehicle, or even taking control of the vehicle. The path along which the attacker moves is called the attack path. Such a path could be demonstrated, for example, in the paper ”Free-Fall: Hacking Tesla from wireless to Controller Area Network (CAN) bus” by Keen Security Labs. The researchers succeeded in sending messages wirelessly to the vehicle’s CAN bus [11].

The same lab was also able to identify more vulnerabilities that demonstrate that systems in vehicles are vulnerable to remote attacks. For example, Bluetooth, Global System for Mobile Communications (GSM) and some BMW-specific services such as BMW ConnectedDrive were used as entry points into the vehicle. By exploiting further vulnerabilities in the vehicle network, it was possible to find an attack path to gain control of the CAN bus [12].

One of the best-known publications, ”Remote Exploitation of an Unaltered Passenger Vehicle” highlighted the risks associated with connected vehicles back in 2015. Valasek and Miller demonstrated the vulnerability of a vehicle’s infotainment system. Using various attack paths, they managed to

make significant changes to the vehicle. They were able to control the air conditioning, the brakes, the acceleration and even the steering in reverse gear [13].

C. Security education

Teaching at universities is often theory-based. As a result, many graduates may lack the practical experience to identify vulnerabilities. But it is precisely this experience that is of great importance in the professional field of software development, security testing, and engineering. The idea is to develop the competence level from a novice to an experts level, which can be guided by "Security Tester" certified Tester Advanced Level Syllabus. The described APTEP presents an ecosystem to establish such learning arrangements in which constructivism-based learning will happen [14][15].

In its 2016 IT-Grundschutz-Kataloge, the Bundesamt für Sicherheit in der Informationstechnik (BSI) proposes the measure "Implementation of information security simulation games" (M3.47). This measure is preferable to classic slide presentations, leading to more concise and sustainable learning success. In addition, they help to illustrate threats and typical vulnerabilities and to point out possible solutions. Measure M3.47 no longer exists in the current BSI-Grundschutz. However, it has been replaced by ORP.3 "Awareness and training on information security". ORP.3.A4 "Design and plan an information security awareness and training program" states that information security awareness and training programs should be targeted to specific audiences. It should be possible to tailor training to specific needs and diverse backgrounds. ORP.3.A8 "Measurement and Assessment of Learning Success" also states that information security learning success should be measured and assessed on a target group basis to determine the extent to which the objectives described in information security awareness and training programs have been achieved. APTEP is intended to make precisely this possible [16][17].

There are many different teaching and learning designs used in practice today to support learning. Some of the most commonly used are listed in Table II.

TABLE II
LEARNING/TEACHING DESIGN CATEGORIZATION BASED ON [18]. THE SYMBOL "+" INDICATES IF THE GIVEN CRITERIA IS VALID. C = CONTEXT, Q = QUESTION, A = APPROACH, S = SOLUTION

Learning/teaching Design	C	Q	A	S
Ex-Cathedra	+	+	+	+
Simulation Games	+	+	+	
Term Paper	+	+		
Learning by Teaching	+	+		
Expert Discussions	+	+		
Problem-based/Challenge-based Learning	+			
Discovery-/Research-based Learning				

Students who ask questions, solve problems, create solutions, propose alternatives, engage in hands-on activities, and participate in learning groups are likely to learn more and retain information and skills longer than students who sit passively listening to a lecture in the format of Ex-cathedra teaching [18].

Problem-based/Challenge-based learning focuses on complex real-world problems and their solutions. Inductive teaching describes those student activating approaches [19]. The challenge selects a security problem that is well-defined and that requires sustained investigation and collaboration.

Students are not given a list of resources but must conduct their own searches and distinguish relevant from irrelevant information [20]. These authentic activities engage students in making choices, evaluating competing solutions, and creating a finished penetration test in the goal of security hardening. The summary of criteria given to the student is shown in Table II.

III. ARCHITECTURE

The attacks from the previous section show that attacks follow a similar pattern. There is an entry point through which the attacker gains access to the vehicle. He then tries to move through the vehicle network by exploiting further vulnerabilities. He does this until he reaches his target. To represent this procedure in the architecture of APTEP, it was divided into different layers.

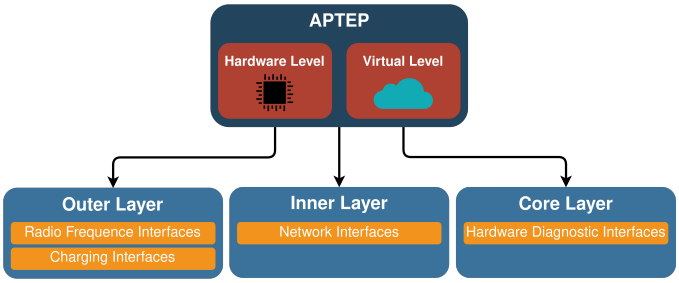


Fig. 1. APTEP Architecture

As shown in Figure 1, the following three layers were chosen: Outer layer, inner layer, and core layer. They delimit the respective contained interfaces from each other.

A. Outer Layer

The automotive industry is currently focusing heavily on topics, such as automated driving functions, Vehicle-to-Everything (V2X) networking, and Zero-Emission Vehicles (ZEV). In these areas, new trend technologies can lead to valuable new creations. But unfortunately, this development also favors the emergence of new and more critical points of attack. For this reason, the outer layer was included in the APTEP as part of the architecture. It contains all the functionalities that enable the vehicle to communicate with its environment. This includes the two V2X technologies Cellular-V2X and Wireless Local Area Network (WLAN)-V2X as well as other communication protocols, such as Bluetooth and GSM. In addition to the communication protocols, there are also interfaces, such as various charging interfaces, sensors, and much more.

The outer layer represents an important component because many interfaces contained in it represent a popular entry point for attacks. This is the case because the technologies used there are usually an option to potentially gain access to the vehicle without having physical access to the vehicle. Even if the sole exploitation of a vulnerability within the outer layer does not always lead to direct damage in practice, further attack paths can be found over it. In most cases, several vulnerabilities in different areas of the vehicle system are combined to create a critical damage scenario from the threat. Therefore, vehicle developers need to be particularly well trained in this area.

B. Inner Layer

The inner layer of the APTEP represents the communication between individual components. While modern vehicles implement different forms of communication, bus systems like CAN, Local Interconnect Network (LIN), and FlexRay used to be predominant. Since modern vehicle functions connected to the Outer Layer, like image processing for rear-view cameras or emergency braking assistants [21], require data rates not achievable by the previously mentioned bus systems, new communication technologies, like Ethernet, have been implemented in vehicles.

Depending on the scope, the mentioned bus systems are still in use because of their low cost and real-time capabilities. From those communication technologies, different network topologies can be assembled. Individual subsystems connecting smaller components, e.g., ECUs, are themselves connected through a so-called backbone. Gateways are implemented to connect the subsystems with the backbone securely.

After gaining access to a vehicle through other means, the inner layer represents an important target for attackers since it can be used to manipulate and control other connected components. While the target components can be part of the same subsystem, it is also possible, that it is part of a different subsystem, forcing the attacker to communicate over the backbone and the connected gateways. The inner layer thus represents the interface between the outer - and core layer.

C. Core Layer

Manipulating the ECU of a vehicle themselves results in the greatest potential damage and therefore represents the best target for a hacker. In the APTEP, this is represented as the core layer.

Vehicles utilize ECUs in different ways, e.g., as a Body Control Module, Climate Control Module, Engine Control Module, Infotainment Control Unit, Telematic Control Unit. In addition, electric vehicles include further ECUs for special tasks, such as charging and battery management.

If attacks on an ECU are possible, its function can be manipulated directly. Debugging and diagnostic interfaces, like Joint Test Action Group (JTAG) or Unified Diagnostic Services (UDS), are especially crucial targets since they provide functions for modifying data in memory and reprogramming of ECU firmware.

The impact of arbitrary code execution on an ECU is dependent on that ECUs function. While taking over, e.g., a car's infotainment ECU should only have a minor impact on passengers' safety, it can be used to attack further connected devices, via inner layer, from an authenticated source. The goal of such attack chains is to access ECUs where safety-critical damage can be caused. Especially internal ECUs interacting with the engine can cause severe damage, like shutting off the engine or causing the vehicle to accelerate involuntarily.

IV. Interfaces

This section describes some chosen interfaces of the previously presented layers. The selection was made from the following three categories: "Radio Frequency and Charging Interfaces", "Network Interfaces" and "Hardware Diagnostic Interfaces".

Implemented in the ANSKo is one interface from each architecture layer - CHAdeMO from the outer layer (Section IV-A3a, CAN from the inner layer (Section IV-B1), and UDS from the core layer (Section IV-C2). This facilitates the cross-domain challenges described in Section VI.

A. Radio Frequency and Charging Interfaces

The outer layer contains the interfaces of the category "radio frequency and charging interfaces". They all have in common that they enable the vehicle to communicate with its environment. Furthermore, the included interfaces can be divided into the following classes: short-range communication, long-range communication, and charging interfaces.

1) Short-range Communication:

a) Bluetooth: Bluetooth is a radio standard that was developed to transmit data over short distance wireless. In the vehicle, the radio standard is used primarily in the multimedia area. A well-known application would be, for example, the connection of the smartphone to play music on the vehicle's internal music system.

b) RFID: Radio Frequency Identification (RFID) enables the communication between an unpowered tag and a powered reader. A powered tag makes it possible to increase the readout distance. RFID is used, for example, in-vehicle keys to enable key-less access.

c) NFC: Near Field Communication (NFC) is an international transmission standard based on RFID. The card emulation mode is different from RFID. It enables the reader to also function as a tag. In peer-to-peer mode, data transfer between two NFC devices is also possible. In vehicles, NFC is used in digital key solutions.

d) WLAN-V2X: The WLAN-V2X technology is based on the classic WLAN 802.11 standard, which is to be used in short-range communication for V2X applications. However, almost all car manufacturers tend to focus on Cellular-V2X because long-range communication is also possible in addition to short-range communication.

2) Long-range Communication:

a) GNSS: The Global Navigation Satellite System (GNSS) comprises various satellite navigation systems, such as the Global Positioning System (GPS), Galileo, or Beidou. Their satellites communicate an exact position and time using radio codes. In vehicles, GNSS is mainly used in onboard navigation systems. Furthermore, it is increasingly used to manage country-specific services. In the autonomous driving context the position is mandatory to locate the vehicle from distance by a technical supervisor.

b) Cellular-V2X: An increasingly important technology of the future is Cellular-V2X. Cellular-V2X forms the communication basis for V2X applications. It uses the cellular network for this purpose. In contrast to WLAN-V2X, it enables both Vehicle-to-Vehicle (V2V) and Vehicle-to-Network (V2N) communication.

3) Charging Interfaces: To enable charging or communication between an electric vehicle and a charging station, a charging interface is required. Due to the high diversity in this area, there is not just one standard.

a) CHAdeMO: The CHAdeMO charging interface was developed in Japan where it is also used. The charging process can be carried out with Direct Current (DC) charging. Mainly Japanese OEMs install this charging standard in their vehicles. Some other manufacturers offer retrofit solutions or adapters.

b) ChaoJi: A proposed and further developed standard of CHAdeMO is ChaoJi. It allows for even higher charging performance and greater compatibility. The design is backward compatible with CHAdeMO and the GB/T DC charging system, using a separate input adapter for each system. ChaoJi's circuit interface is also fully compatible with Combined Charging System (CCS).

c) Tesla: Tesla predominantly uses their own charging interface, which allows both Alternating Current (AC) and DC charging. However, due to the 2014/94 EU standard, Tesla is switching to the CCS Type-2 connector face in Europe.

d) GB/T: The Chinese charging standard is GB/T. It is used exclusively for charging electric vehicles in China. It covers both AC and DC charging. The plug standard for AC is reminiscent of the European Type 2 plug, the DC version is very similar to CHAdeMO.

e) CCS: The official European charging interfaces CCS Type-1 and CCS Type-2 are based on the AC Type-1 and Type-2 connectors. The further development enables a high DC charging capacity in addition to the AC charging.

B. Network Interfaces

Network interfaces describe the technologies used to communicate between components, like ECUs or sensors. It represents the inner layer.

1) CAN: CAN is a low-cost bus system, that was developed in 1983 by Bosch. Today it is one of the most used bus systems in cars since it allows acceptable data rates of up to 1 Mbit/s while still providing real-time capabilities because of its message prioritization. Its design as a two-wire system also makes it resistant to electromagnetic interference.

Traditionally in a vehicle CAN is often used as the backbone, providing a connection between the different subsystems. It is also used in different subsystems itself, like engine control and transmission electronics.

2) LIN: The LIN protocol was developed as a cost-effective alternative to the CAN bus. It is composed of multiple slave nodes, which are controlled by one master node, which results in a data rate of up to 20 Kbit/s.

The comparatively low data rate and little fault resistance result that LIN being mainly used in non-critical systems, like power seat adjustment, windshield wipers, and mirror adjustment. The communication is synchronous - the master requests data from the slave, which answers the request afterwards.

3) MOST: The Media Oriented System Transport (MOST) bus provides high data rates of 25, 50, or 150 Mbit/s depending on the used standard. It was developed specifically for use in vehicles and is typically implemented as a ring.

As the name suggests the field of application for the MOST bus is not in safety-critical systems, but in multimedia systems of a vehicle. Since transmission of uncompressed audio and video data requires high data rates, MOST is suited best for those tasks.

4) FlexRay: FlexRay offers data transmission over two channels with 10 Mbit/s each. They can be used independently or by transmitting redundant data for fault tolerance. Furthermore, FlexRay implements real-time capabilities for safety-critical systems.

FlexRay was developed with future X-by-Wire (steer, brake, et al.) technologies in mind [22]. Even though FlexRay and CAN share large parts of their requirements, FlexRay improves upon many aspects, leading to it being used as a backbone, in powertrain and chassis ECUs and other safety-critical subsystems.

5) Ethernet: The Ethernet protocol is the backbone of today's society. It was introduced commercially in 1980 and is a family of wired networking technologies. Speeds range from 3 MBit/s to more than 1 TBit/s.

a) Standard Ethernet: The Ethernet network technologies used in public are also present in cars. Due to the constant increase in required data rates of new technologies, such as image processing, Ethernet has been adapted for use in vehicles. The widespread use outside of vehicles has the advantage that many functions are already programmed and can be reused.

The underlying physical layer of the Ethernet protocol is not suitable for use in systems with electromagnetic interference, nor does it offer real-time capabilities, but this can be remedied by using the Audio-Video-Bridging (AVB) standard. The main use of standard Ethernet in the car is for simple high-speed access to Diagnostics over Internet Protocol (DoIP) and logging of ECU output, or direct access to an ECU via Secure Shell (SSH) during development.

b) Automotive Ethernet: The goal of Automotive Ethernet was to provide a lower cost transmission protocol with high data rates of up to 1 GBit/s that could withstand

electromagnetic interference while taking advantage of the long established functionality of the upper layers of Ethernet. Currently, there are three types that differ in speed:

- 10Base-T1 (10 MBit/s)
- 100Base-T1 (100 MBit/s)
- 1000Base-T1 (1 GBit/s)

To achieve low cost, speed and resistance to electromagnetic interference, a different physical layer such as BroadR-Reach is used, which uses a single twisted pair cable.

6) USB: Universal Serial Bus (USB) is mainly used by the cars' infotainment system. Smartphones can be connected and technologies such as Apple Carplay or Android Auto are used to extend the vehicle's functions through popular smartphone apps. Depending on the age of the vehicle, different USB types are used, with the latest vehicles using Type C USB.

C. Hardware-Diagnostic Interfaces

The hardware-diagnostic interfaces are classified in the core layer. They describe technologies that allow interaction between a person, such as a programmer, and an ECU to allow, e.g., reprogramming of the software.

1) Debug: Debug interfaces are used in embedded development to allow debugging, reprogramming, and reading out error memory of the circuit boards. Vehicles implement various debug interfaces, depending on their integrated circuit boards. The most common interfaces include JTAG, Serial Wire Debug (SWD), Universal Asynchronous Receiver Transmitter (UART), and USB.

Interacting with the debug interfaces requires special equipment, like adapters.

2) UDS: Modern vehicles implement a diagnostic port as well to allow independent car dealerships and workshops functionalities similar to the debug interfaces while not being unique to one particular OEM. It uses the communication protocol UDS, defined in the ISO 14229 standard.

UDS utilizes CAN as the underlying protocol to transmit messages. To prevent unauthorized access to the diagnostic port, UDS provides different tools, like "Diagnostic Session Control" which defines different sessions, such as default, diagnostic, or programming. OEMs can choose which service is available in each session. Security-critical services can also be further guarded by using the "Security Access" which protects the respective service through a key seed algorithm.

In newer vehicles, UDS is also implemented on the Ethernet network, the underlying transport protocol is DoIP. UDS over Ethernet has the advantage that the transmission speed is faster than over CAN.

3) OBD: The On-board diagnostics (OBD) offers access to multiple network interfaces of a vehicle. It can be used to read diagnostic information and also various parameters such as the current engine revolutions per minute (rpm) or the control module voltage.

4) CAM: A Cooperative Awareness Messages (CAM) contains information about the current situation of the vehicle like speed, driving direction, geographic position and the general conditions. They were sent periodically from self driving vehicles to surrounding vehicles or a technical supervisor. The period depends from different environmental parameters. For example a higher speed can lead to a higher frequency of sending the messages, to ensure that fast changing environment can be detected in detail.

5) DENM: The other way round the vehicle is able to receive Decentralized Environmental Messages (DENM) from outside. They are sent from the technical supervisor depending on the situation. Especially with the purpose to bring the car to a state of minimum risk if needed. But they also provide the possibility to request special information from the on board electronic or to decide between two or more possible driving maneuvers. Cars can send DENM to warn other cars from special conditions like black ice.

6) Side Channels: Side channels are also a relevant interface in the core layer. A computing unit emits certain side-channel data while performing operations, such as the consumed energy while encrypting data. They allow attackers to gain information about secret parts of the computer system like the used keys for cryptographic operations. Side-channel data can therefore be used to attack otherwise secure computer systems. Possible different side channels include time, power, fields, and temperature.

V. Structure

The presented APTEP is implemented in the ANSKo, which consists of a hardware and a virtual level. Their required components and used software are described in the following.

A. Hardware-Level

The goal of the ANSKo is to provide a low-cost learning environment for automotive security. The case consists of two Raspberry Pis simulating the ECUs of the respective challenge. They are connected via CAN, which represents the main communication in modern vehicles. Users can interact with the CAN bus by connecting USB cables to the included Embedded 60 microcontroller. To modify the running software or install required libraries, an Ethernet switch connecting to the Raspberry Pis is present as well. In the future, other challenges will be implemented using the Ethernet connection as Automotive Ethernet. To allow participants to work with the case without requiring them to install virtual machines with multiple software packages, a preconfigured laptop is included. Distinguishing between the master and slave Raspberry Pis is done by attaching a resistor to the PiCan2 Duo board, which can afterward be read on pin 11.

A picture of the hardware contents can be seen in Figure 2. The currently included components are marked by color boxes. It is intended to further extend the platform by the listed interfaces in Section IV.

- **Yellow - Ethernet Switch:** The Ethernet switch connects to both Raspberry Pis and allows additional connections to the user.
- **Red - Display and Raspberry Pis:** The main components of the case are two Raspberry Pis, which simulate ECUs in a vehicle. They possess a PiCAN Duo board

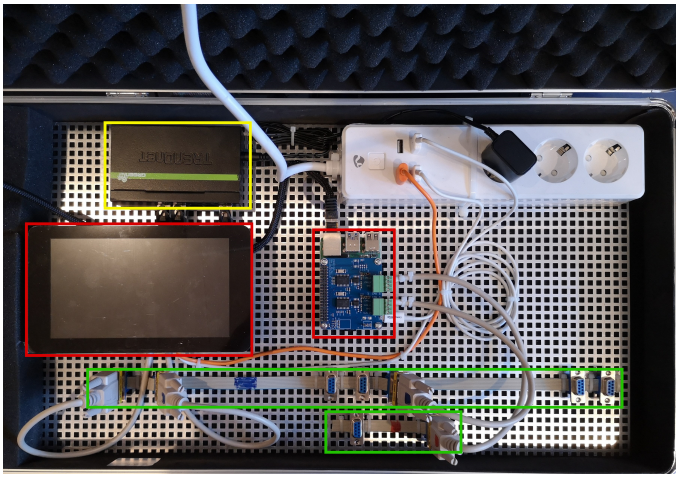


Fig. 2. ANSKo Hardware

allowing two independent CAN connections. One of the Raspberry Pis possesses a display, simulating a dashboard with a speedometer and other vehicle-specific values.

- **Green - CAN bus:** The CAN bus is the main communication channel in the current structure. Connected devices can be disconnected by removing the respective cables.

One example of an implemented challenge in the ANSKo is a Man-in-the-Middle attack. The goal is to lower the displayed mileage of the car to increase its value. A user working with the ANSKo needs to read the messages being sent between the simulated ECUs. They can interact with the CAN bus by connecting to the CAN bus via USB cable and the included microcontroller. The challenges are described in more detail in Section VII.

The operating system running on the Raspberry Pis was built by using pi-gen. It is a tool for generating and customizing a Raspberry Pi Operating System (OS) image. Pi-gen splits the settings into different stages. Starting at stage 0, where the firmware and language dependent files are loaded, to stage 5, which contains needed software packages for the challenges. Additionally pi-gen allows setting the Wi-Fi Service Set Identifier (SSID), Wi-Fi password, first username and user password via a config file [23].

Configuring the Wi-Fi settings is necessary, because installing challenges on multiple cases is a time consuming process. To allow the delivery via SSH, the Raspberry Pis need to have a static Internet Protocol (IP) address. As mentioned before, the master and slave Raspberry Pi are distinguished by reading out pin 11, which allows setting their respective IP address automatically. By using the automation software Ansible, challenges can be installed on all cases simultaneously [24]. Challenges are started as a systemd service after copying the required files to the cases.

B. Virtual-Level

During the Covid-19 pandemic holding education courses hands-on was not possible. To still provide the advantages of the ANSKo during lockdowns, an online platform with identical challenges has been realized.

The virtual challenges are accessible through a website, which allows the authentication of users. A user can start a challenge, which creates a Docker container. This ensures an independent environment for users while also protecting the host system [25].

Users can receive the necessary CAN messages by using the socketcand package, providing access to CAN interfaces via Transmission Control Protocol/Internet Protocol (TCP/IP) [26].

The unique docker containers for each user allow them to stop and start working on the challenge at any time but limits the maximum amount of users attempting the challenges concurrently. Validation of a correct solution also does not have to be carried out manually because the sending of a unique string of characters on the CAN bus signals the challenge has been solved to the user.

VI. Didactics Learning Concept

In this section, the learning concept of ANSKo is described. Evaluation will be given in Section VIII. ANSKo's concept of learning is based on the theory of constructivism. This theory is about learners constructing their own understanding by developing existing knowledge to gain a deeper understanding. It allows learners to achieve the higher-order learning goals of Bloom's Taxonomy [27]. They are more capable of analyzing facts and problems, synthesizing known information, and evaluating their findings [28].

Learning concepts that are following the theory of constructivism are used to encourage learners to actively think rather than passively absorb knowledge, e.g., Problem-Based Learning (PBL). ANSKo consists of several real-world problems, so-called challenges. Problem-based/challenge-based learning begins with a problem or task that determines what students study. The problems derive from observable phenomena or events, which students come to understand as they learn about the underlying explanatory theories [20].

Therefore, students will learn in a relevant security context. In our learning arrangement problem solving support is provided using the scaffolding approach in a self-directed education process: Learners initially select or receive the theoretical knowledge needed to solve the problem in collective learning providing one another with feedback. Then they work independently to solve the problem and can support each other within the groups. The teacher stimulates reflexion, guides the learning process and gives insights in acquiring the knowledge to solve the problem [28]. Figure 3 shows the process of the described problem-based/challenge-based learning.

The challenges can be divided into two categories: "Domain-specific challenges" and "Cross-domain challenges". The two types each pursue different learning objectives.

As shown in Figure 4, "Domain-specific challenges" are about learning the functionalities and vulnerabilities of a single interface within a domain. A challenge is considered complete when the learner has found and exploited the vulnerability.

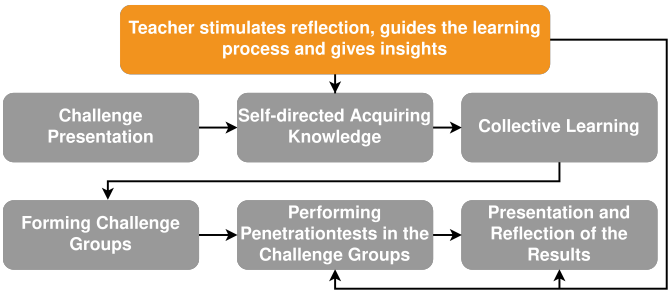


Fig. 3. Learning Concept

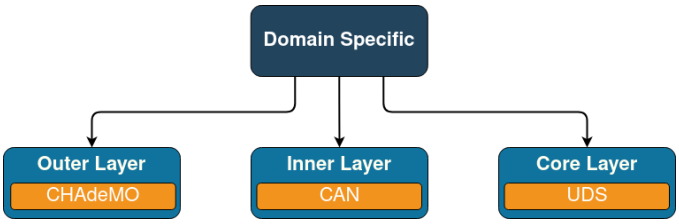


Fig. 4. Domain-specific Challenge

Cross-domain challenges aim to teach the learner how to find and exploit attack paths. Figure 5 shows an example of a cross-domain challenge. Here, interfaces from the different layers are combined. The difficulty level of these challenges is higher and therefore the respective domain-specific challenges for the required interfaces have to be solved first.

VII. Technical Challenges

Currently, a total of six different challenges have been implemented (see Table III). The challenges are divided into various difficulty levels from easy to hard. With the currently realized challenges, levels 3 (Apply), 4 (Analyze), and 5 (Evaluate) of Bloom's Taxonomy can be achieved. Predominantly, challenges have been designed and developed following the type domain specific. The CHAdeMO challenge corresponds to cross-domain. In the future, the ANSKo will be extended by further challenges, with the goal to give the students access to most technologies described in Section IV.

To illustrate the learning concept, this section presents an example of a challenge implemented on the ANSKo platform. The presented challenge is the introduction to hacking a automotive network. The background of the challenge is the following: Person A (the student) would like to sell his car to

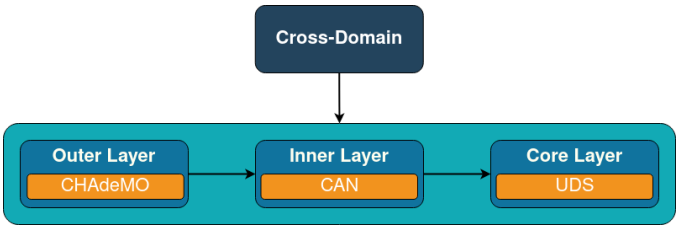


Fig. 5. Cross-domain Challenge

TABLE III
ANSKo Challenges

No.	Name	Type	Difficulty	Bloom's Taxonomy
1	CAN Man-in-the-middle attack (MITM) Attack	Domain-specific	Easy	Level 4: Analyze
2	ISO-TP Entry	Domain-specific	Easy	Level 4: Analyze
3	UDS Scanning	Domain-specific	Medium	Level 4: Analyze
4	Eavesdropping	Domain-specific	Medium	Level 3: Apply
5	Denial of Service	Domain-specific	Medium	Level 3: Apply
6	Charging Interface CHAdeMO	Cross-domain	Hard	Level 5: Evaluate

person B. However, the car has a very depreciating feature: It already has 100.000 km on the tachometer. To solve this "problem" person A wants to use a MITM to reduce the displayed kilometers.

This idea has already been demonstrated in research, and also occurs in reality, with shady car sellers increasing the value of their cars [29] [30]. The structural reason for this hack working is the distributed storage of information in the vehicle. The tachometer reads the mileage from the CAN bus, which is sent by the engine control unit. On the ANSKo platform the Raspberry Pi with display simulates the tachometer, the other Raspberry Pi simulates the engine control unit, which sends the total mileage.

The student is given a short description of the tasks goal: *"You want to sell your old car. It's pretty used and it will probably not sell for a lot of money. To counter this you want to set the amount of driven kilometers back by a certain amount, 50.000 in this case. This will make it more valuable and more buyers might be interested."* with some tips to make the task easier. These tips contain a description of how to connect to the CAN bus to a PC and the following statement's:

- Different messages are send over the CAN bus.
- Try to align the identifiers to the values on the display.
- Some values might not make sense to you.
- Use the EMB60 as a interface to look at them.
- Your goal is it to try a man in the middle attack between the two ECUs.
- Try to set the amount of driven kilometers back by 50.000.
- You need to separate the two ECUs from each other, be careful when removing the interfaces.
- Scapy has functions for sniffing and bridging two CAN networks, check the docs!

First, the student must listen to the CAN bus to identify the message that transmits the mileage. This is the first challenge that must be overcome: In the automotive field CAN messages

are coded extremely efficiently. That means that a message transmits many different values and these have exactly the bit lengths needed in the message. Reading the messages byte by byte can therefore result in no meaningful values. To analyze the CAN bus messages, the Embedded 60 has to be connected to the CAN bus between the Raspberry Pis (see Figure 6).

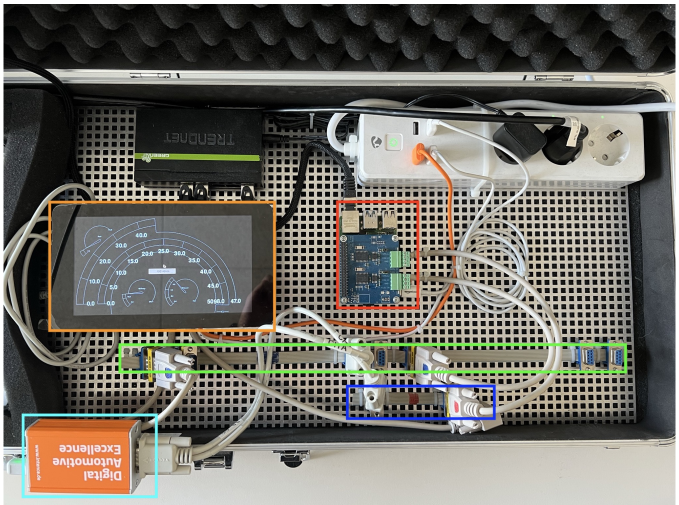
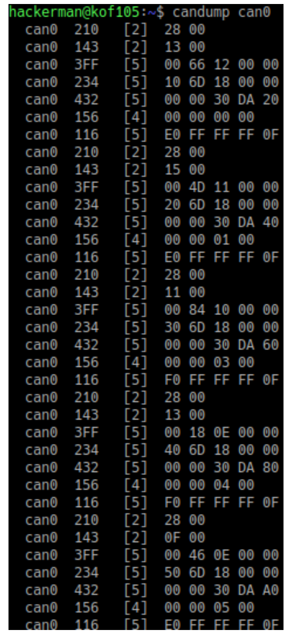


Fig. 6. ANSKo MITM Setup

- **Orange - Tachometer:** The Raspberry Pi with display acts as tachometer and shows extra information if the challenge is complete.
- **Red - Engine Control Unit:** The Raspberry Pi without display acts as engine control unit an sends the mileage an other messages to the tachometer.
- **Green - CAN bus:** The green CAN bus is the communication channel between tachometer and engine control unit.
- **Blue - CAN bus:** The blue CAN bus is a additional communication line that is not used in the default setup, and can help for the MITM attack.
- **Cyan - Embedded 60:** The Embedded 60 is used to connect the CAN Bus to the PC.

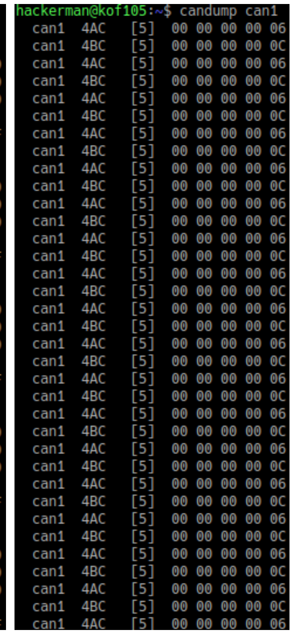
To make it easier to find out the mileage message, the student can disconnect the tachometer or the engine control unit from the CAN Bus and connect it to the spare bus. Since both Raspberry Pis are sending messages on the CAN, this eliminates the tachometer messages from being analyzed, since they don't contain the mileage (see Figures 7 and 8).

To find the mileage message ID the student has to analyze the contents of the messages. First a look at periodical increasing data in the messages can eliminate the ones that don't change or increase step by step. The correct message is the ID 234, the mileage is in the data, for example 30 6D 18. But how can that data be translated into the 100.000 km? One clue of the message is that the first number is increasing while in the mileage its the last. This is a indication that little endian is used to encode the message. Translating it back gives the result 18 6D 30, still to big by a factor x16 for the mileage.



```
hackerman@kof105:~$ candump can0
  can0  210   [2]  28 00
  can0  143   [2]  13 00
  can0  3FF   [5]  00 66 12 00 00
  can0  234   [5]  10 6D 18 00 00
  can0  432   [5]  00 00 30 DA 20
  can0  156   [4]  00 00 00 00
  can0  116   [5]  E0 FF FF FF 0F
  can0  210   [2]  28 00
  can0  143   [2]  15 00
  can0  3FF   [5]  00 4D 11 00 00
  can0  234   [5]  20 6D 18 00 00
  can0  432   [5]  00 00 30 DA 40
  can0  156   [4]  00 00 01 00
  can0  116   [5]  E0 FF FF FF 0F
  can0  210   [2]  28 00
  can0  143   [2]  11 00
  can0  3FF   [5]  00 84 10 00 00
  can0  234   [5]  30 6D 18 00 00
  can0  432   [5]  00 00 30 DA 60
  can0  156   [4]  00 00 03 00
  can0  116   [5]  F0 FF FF FF 0F
  can0  210   [2]  28 00
  can0  143   [2]  13 00
  can0  3FF   [5]  00 18 0E 00 00
  can0  234   [5]  40 6D 18 00 00
  can0  432   [5]  00 00 30 DA 80
  can0  156   [4]  00 00 04 00
  can0  116   [5]  F0 FF FF FF 0F
  can0  210   [2]  28 00
  can0  143   [2]  0F 00
  can0  3FF   [5]  00 46 0E 00 00
  can0  234   [5]  50 6D 18 00 00
  can0  432   [5]  00 00 30 DA A0
  can0  156   [4]  00 00 05 00
  can0  116   [5]  E0 FF FF FF 0F
```

Fig. 7. Engine Control Unit Messages



```
hackerman@kof105:~$ candump can1
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
  can1  4BC   [5]  00 00 00 00 0C
  can1  4AC   [5]  00 00 00 00 06
```

Fig. 8. Tachometer Messages

By shifting the result 4 bits to the left, the result 18 6D 3 is exactly the 100.051 km mileage. In the real world the 0 that is not used further in this example would have a meaning too, the 4 bits could be used for flags or other data to use the available message payload perfectly.

After the message ID has been found, the MITM attack can be prepared. For that, the Raspberry Pis have to be separated on two different CAN buses, and the Embedded 60 connected to both acting as a bridge between them. The software part of the attack can be done over Scapy, or other network packet manipulating tools [31]. All messages except the mileage message need to be forwarded to the other CAN bus, the mileage is read and then altered to set down the driven km by 50.000. The result after a successful MITM attack is shown in Figure 9.

VIII. Evaluation

In this section, the learning platform is evaluated based on various criteria. On the one hand, the learning platform has already been used in some courses at the OTH Regensburg. The students were surveyed at the end of the course. The results are presented in this paper. On the other hand, the learning platform is evaluated with the help of the Open Web Application Security Project (OWASP) SAMM.

A. *Evaluation in the context of the course*

Computer science students from the OTH Regensburg were able to work with the ANSKo as part of a special topic course for the 6th & 7th semesters. The course evaluation, which was answered by the students, showed the benefit of the learning platform.

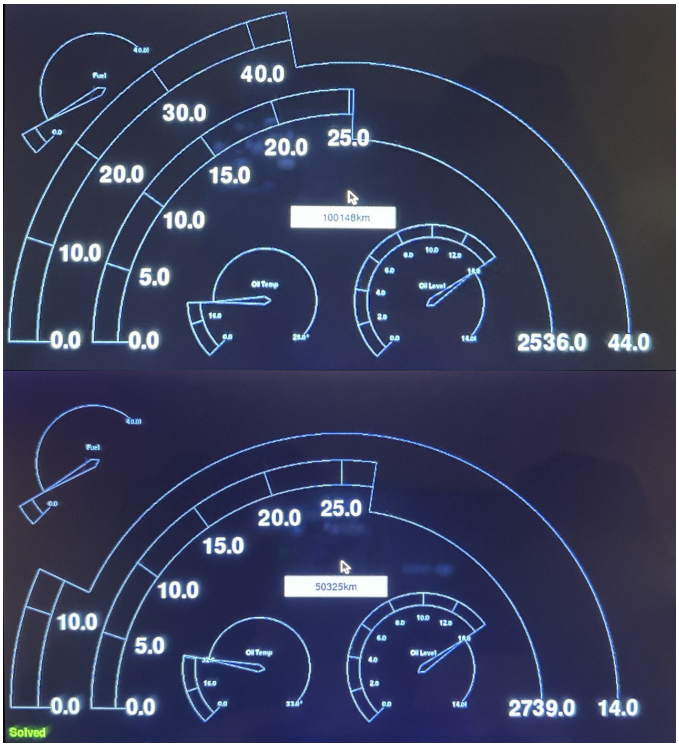


Fig. 9. ANSKo Tachometer (Top: Before; Bottom: After)

A dedicated online evaluation questionnaire was designed to ensure the quality of teaching with the learning platform. This questionnaire is filled out anonymously by 21 learners at the end of the course. The questions are closed and allow a selection within a rating scale. This scale goes from 1 to 5, with 1 being the most negative selection and 5 being the most positive. The following questions are included:

- **EQ1** - Did you like the course overall?
- **EQ2** - Are you satisfied with your learning progress regarding security?
- **EQ3** - Can you independently reproduce the topics covered?
- **EQ4** - How do you rate the principle of "problem-based/challenge-based Learning" compared to traditional forms of teaching?
- **EQ5** - How do you evaluate the work in small groups?
- **EQ6** - Some of the security vulnerabilities shown occur when programmers write buggy code. Do you think your code will be free of these errors in the future?
- **EQ7** - How satisfied were you with the automotive part of the course?
- **EQ8** - How do you rate the topicality of the subject of "automotive security"?
- **EQ9** - Was the level of difficulty of the automotive topics covered appropriate?
- **EQ10** - Was the level of difficulty of the exercise tasks automotive appropriate?

Figure 10 shows the evaluation results in the form of a Kiviat diagram. The different evaluation questions EQ1-EQ10 are visualized on the axes. There is a grid for each of the five steps of the evaluation scale. The green area in the diagram is defined by the mean value of the survey results per question. The further out the green box is on the axis, the better the question was rated.

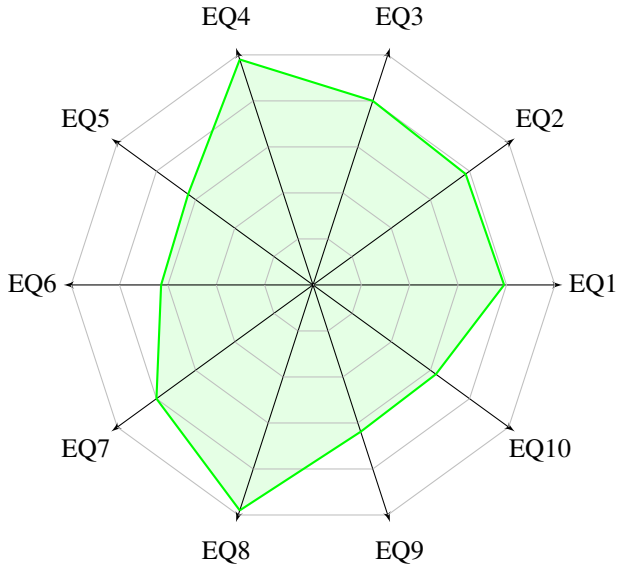


Fig. 10. Evaluation of the ANSKo

It can be seen from the diagram that most of the questions were answered very positively. The students reported a positive experience when working with the ANSKo, e.g., when asked about understanding the importance of automotive security or their learning progress. It should be noted that questions EQ9 and EQ10 are moderately rated. It can be concluded from this that the difficulty level of the course is appropriately challenging. EQ6 was given an average grade of 3.38. This indicates that the students understood that errors can occur during programming and that software must therefore be tested. This is an important understanding, as errors can occur even with a high level of maturity and strict security controls. Another striking feature is that EQ5 was given an average score of 3.19. This indicates that some students would have preferred a different grouping. In the field of pentesting, group work is essential. In practice, large teams work on common tasks. They do best when their knowledge complements each other as much as possible. Unfortunately, this can only be realized to a limited extent at the university due to the given general conditions.

B. Evaluation based on the OWASP SAMM

As maturity model for software assurance SAMM can be used in the presented IT-Security education framework. The Education & Guidance (EG) practice focuses on arming personnel involved in the software lifecycle with knowledge and resources to design, develop, and deploy secure software. With improved access to information, project teams can proactively

identify and mitigate the specific security risks that apply to their organization [32].

In the following Table IV we present a mapping of the maturity levels in the SAMM practice Education & Guidance (EG) to the ANSKo approach. With the presented IT-Security education framework it is possible to achieve SAMM level 3 in the practice EG.

TABLE IV
MAPPING OF SAMM PRACTICE EG TO THE ANSKO APPROACH

Maturity Level	SAMM EG: Description of given maturity level	Presented IT-Security education framework (ANSKo approach)
1	Offer staff access to resources around the topics of secure development and deployment.	The presented IT-Security education framework gives access to non-compliant and compliant examples of secure software engineering.
1	Provide security awareness training for all personnel involved in software development.	The presented cursus is useable in an industrial context for all software engineers.
1	Identify a "Security Champion" within each development team.	Define a responsibility for IT-Security in the team.
2	Educate all personnel in the software lifecycle with technology and role-specific guidance on secure development.	The presented cursus is useable for different roles in a software organization.
2	Offer technology and role-specific guidance, including security nuances of each language and platform.	There is a scaffolding approach possible for different roles.
2	Develop a secure software center of excellence promoting thought leadership among developers and architects.	The presented IT-Security education framework can be extended in the team for new challenges.
3	Develop in-house training programs facilitated by developers across different teams.	The challenges in the presented IT-Security education framework can be matched to the focus of different teams.
3	Standardized in-house guidance around the organization's secure software development standards.	The presented IT-Security education framework can be adopted in the organization's secure software development standards.
3	Build a secure software community including all organization people involved in software security.	The presented IT-Security education framework can generate room and time for the communication of all organization people.

The Security Testing (ST) practice leverages the fact that, while automated security testing is fast and scales well to numerous applications, in-depth testing based on good knowledge of an application and its business logic is often only possible via slower, manual expert security testing [33].

In the following Table V we present a mapping of the maturity levels in the SAMM practice ST to the ANSKo approach. With the presented IT-Security education framework it is possible to achieve SAMM level 3 in the practice ST.

TABLE V
MAPPING OF SAMM PRACTICE ST TO THE ANSKO APPROACH

Maturity Level	SAMM ST: Description of given maturity level	Presented IT-Security education framework (ANSKo approach)
1	Perform security testing (both manual and tool based) to discover security defects.	Both is possible with our approach.
1	Make security testing during development more complete and efficient through automation complemented with regular manual security penetration tests.	The presented IT-Security education framework can be extended for automation.
1	Embed security testing as part of the development and deployment processes.	The presented IT-Security education framework can be included in the secure development process.
2	Perform security testing (both manual and tool based) to discover security defects.	The presented IT-Security education framework enables software engineers to perform tests.
2	Employ application-specific security testing automation.	The presented IT-Security education framework can be extended for automation.
2	Integrate automated security testing into the build and deploy process.	The presented IT-Security education framework can be integrated in the secure development process.
3	Perform manual security testing of high-risk components.	The presented IT-Security education framework contents challenges with different risk level.
3	Conduct manual penetration testing.	The presented IT-Security education framework allows manual penetration testing.
3	Integrate security testing into development process.	The presented IT-Security education framework can be integrated in the secure development process.

IX. CONCLUSION AND FUTURE WORK

The presented vulnerabilities at the beginning of this paper and the listing of strengths and weaknesses of existing learning platforms justify the need for an automotive-specific IT security learning platform. For this reason, an APTEP was developed on which participants can learn about vulnerabilities in practice.

To realize this, an architecture for the APTEP was chosen that maps the described attacks. The architecture consists of three layers - outer layer, inner layer, and core layer. Each of them contains different interfaces, such as the Radio Frequency interface as well as the Charging interface in the outer layer, Network interfaces in the inner layer, and Hardware-Diagnostic interfaces in the core layer.

The APTEP is implemented on the Hardware level to provide a realistic learning environment, but also offers a virtual level, which allows users to work with the platform remotely since the COVID-19 pandemic prevented hands-on work.

The ANSKo learning concept is based on the theory of constructivism. This allows the learner to develop a deeper understanding. It also enables the learner to achieve the higher learning goals of Bloom's Taxonomy. ANSKo consists of a variety of challenges and follows the concept of problem-based/challenge-based learning. To keep the challenges as realistic as possible while providing learners with an appropriate level of complexity, the tasks were divided into two categories. There are "Domain-specific challenges," which deal with only one interface per challenge. A "Cross-domain challenge" cannot be solved until the associated "Domain-specific challenges" have been solved for each included interface. The "Cross-domain challenges" combine different interfaces and teach learners to find and exploit attack paths.

Currently implemented are five Domain-specific challenges and one Cross-domain challenge that combines several Domain-specific into one. The challenges are divided into various difficulty levels from easy to hard. With the currently realized challenges, levels 3 (Apply), 4 (Analyze), and 5 (Evaluate) of Bloom's Taxonomy can be achieved.

Evaluation of the APTEP framework and the ANSKo implemented from it was conducted through a university lecture survey. The results were mostly positive. There were moderate responses to the difficulty questions, suggesting that the content was appropriately challenging. Based on the survey results, it was possible to determine that the majority of students recognized that software errors happen. In addition, an evaluation was also performed using the OWASP SAMM.

Future work includes the implementation of electric vehicle-specific challenges, e.g., charging interfaces. Side-channel attack challenges will be included as well. In addition, other challenges are to be added. For example, a Bluetooth attack, an RFID attack, and a fuzzing attack. Another optimization is the integration of a vehicle simulation. This enables a Hardware in the Loop (HiL) approach. Also, a challenge to simulate the communication between a self driving vehicle and a technical supervisor will be developed and included into the ANSKo. Learners then could comprehend which future tasks automotive driving brings to developer as well as to authorities. To support the individual learning progress eye tracking will be included and analyzed. The learner's cognitive load will be determined by Artifical Intelligence (AI)-based classification results. Finally, this will improve individual learning success.

References

[1] S. Schönhärl, P. Fuxen, J. Graf, J. Schmidt, R. Hackenberg, and J. Mottok, "An automotive penetration testing framework for it-security education," *CLOUD COMPUTING 2022*, vol. 13, pp. 1–6, 2022.

[2] "ISO/SAE 21434:2021 Road vehicles — Cybersecurity engineering," International Organization for Standardization and SAE International, Standard, Aug. 2021.

[3] "UN Regulation No. 155 - Cyber security and cyber security management system," United Nations Economic Commission for Europe, Standard, Mar. 2021.

[4] A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on dependable and secure computing*, vol. 1, no. 1, 2004.

[5] Bundesamt für Sicherheit in der Informationstechnik, "Personenzertifizierung: Programm IS-Penetrationstester," Tech. Rep., 2021.

[6] InfosecMatter, *Top 25 penetration testing skills and competencies (detailed)*, 2020. [Online]. Available: https://www.infosecmatter.com/top-25-penetration-testing-skills-and-competencies-detailed/ (retrieved: 10/25/2022).

[7] Hack the Box, *Hack the box*. [Online]. Available: https://www.hackthebox.com/ (retrieved: 10/25/2022).

[8] S. Yang, S. D. Paul, and S. Bhunia, "Hands-on learning of hardware and systems security.," *Advances in Engineering Education*, vol. 9, no. 2, 2021. [Online]. Available: https://files.eric.ed.gov/fulltext/EJ1309224.pdf (retrieved: 10/25/2022).

[9] C. Gay, T. Toyama, and H. Oguma, "Resistant automotive miniature network," [Online]. Available: https://fahrplan.events.ccc.de/rc3/2020/Fahrplan/system/event_attachments/attachments/000/004/219/original/RAMN.pdf (retrieved: 10/25/2022).

[10] T. Espinha Gasiba, U. Lechner, and M. Pinto-Albuquerque, "Cybersecurity Sifu-a cybersecurity awareness platform with challenge assessment and intelligent coach," DOI: 10.1186/s42400-020-00064-4. [Online]. Available: https://doi.org/10.1186/s42400-020-00064-4 (retrieved: 10/25/2022).

[11] S. Nie, L. Liu, and Y. Du, "Free-fall: Hacking tesla from wireless to can bus," *Briefing, Black Hat USA*, vol. 25, pp. 1–16, 2017. [Online]. Available: https://www.blackhat.com/docs/us-17/thursday/us-17-Nie-Free-Fall-Hacking-Tesla-From-Wireless-To-CAN-Bus-wp.pdf (retrieved: 10/25/2022).

[12] Z. Cai, A. Wang, W. Zhang, M. Gruffke, and H. Schweppe, "0-days & mitigations: Roadways to exploit and secure connected bmw cars," *Black Hat USA*, vol. 2019, p. 39, 2019. [Online]. Available: https://i.blackhat.com/USA-19/Thursday/us-19-Cai-0-Days-And-Mitigations-Roadways-To-Exploit-And-Secure-Connected-BMW-Cars-wp.pdf (retrieved: 10/25/2022).

[13] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, no. 91, 2015.

[14] F. Simon, J. Grossmann, C. A. Graf, J. Mottok, and M. A. Schneider, *Basiswissen Sicherheitstests: Aus- und Weiterbildung zum ISTQB® Advanced Level Specialist – Certified Security Tester*. dpunkt.verlag, 2019.

[15] International Software Testing Qualifications Board, *Certified tester advanced level syllabus security tester, international software testing qualifications board*, 2016. [Online]. Available: https://www.german-testing-board.info/wp-content/uploads/2020/12/ISTQB-CTAL-SEC_Syllabus_V2016_EN.pdf (retrieved: 10/25/2022).

[16] Bundesamt für Sicherheit in der Informationstechnik, *It-grundschutz katalog*, 2016. [Online]. Available: https://download.gsb.bund.de/BSI/ITGSK/IT-Grundschutz-Kataloge_2016_EL15_DE.pdf (retrieved: 10/25/2022).

[17] Bundesamt für Sicherheit in der Informationstechnik, *It-grundschutz-bausteine*, 2022. [Online]. Available: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/IT-Grundschutz-Kompendium/IT-Grundschutz-Bausteine/Bausteine_Download_Edition_node.html (retrieved: 10/25/2022).

[18] J. Mottok, J. Merk, and T. Falter, "A multi dimensional view of the graves value systems model on teaching and learning leading to a students-centered learning: Graves model revisited," in *2016 IEEE Global Engineering Education Conference (EDUCON)*, 2016, pp. 503–512. DOI: 10.1109/EDUCON.2016.7474600.

[19] M. Prince and R. Felder, "M.R.: Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases.," *Journal of Engineering Education*, vol. 95, pp. 123–138, 2006.

[20] Davis, Barbara Gross, *Tools for Teaching*, 2nd ed. 2009, ISBN: 978-0787965679.

[21] P. Hank, S. Müller, O. Vermesan, and J. Van Den Keybus, "Automotive ethernet: In-vehicle networking and smart mobility," in *2013 Design, Automation Test in Europe Conference Exhibition*, 2013, pp. 1735–1739. DOI: 10.7873/DATE.2013.349.

[22] W. Zimmermann and R. Schmidgall, *Busssysteme in der Fahrzeugtechnik [Bus systems in automotive engineering]*, ger. Springer Vieweg, 2014, p. 96.

[23] GitHub, *Pi-gen*. [Online]. Available: https://github.com/RPi-Distro/pi-gen (retrieved: 10/25/2022).

[24] Red Hat, *Ansible*. [Online]. Available: https://www.ansible.com/ (retrieved: 10/25/2022).

[25] Docker, *Docker*. [Online]. Available: https://www.docker.com/ (retrieved: 10/25/2022).

[26] GitHub, *Socketcand*. [Online]. Available: https://github.com/linux-can/socketcand (retrieved: 10/25/2022).

[27] Armstrong, Patricia, *Bloom's taxonomy*. [Online]. Available: https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/ (retrieved: 10/25/2022).

[28] G. Macke, U. Hanke, W. Raether, and P. Viehmann-Schweizer, *Kompetenzorientierte Hochschuldidaktik*, ISBN: 9783407294852.

[29] A. Gazdag, C. Ferenczi, and L. Buttyán, *Development of a Man-in-the-Middle Attack Device for the CAN Bus*, 2020. [Online]. Available: http://www.hit.bme.hu/~buttyan/publications/GazdagFB2020citds.pdf (retrieved: 10/25/2022).

[30] Dan Maloney, *Dashboard Dongle Teardown Reveals Hardware Needed To Bust Miles*, 2019. [Online]. Available: https://dangerouspayload.com/2020/03/10/hacking-a-mileage-manipulator-can-bus-filter-device/ (retrieved: 10/25/2022).

[31] GitHub, *Scapy*. [Online]. Available: https://github.com/secev/scapy/tree/master/scapy (retrieved: 10/25/2022).

[32] OWASP, *Education & guidance*. [Online]. Available: https://owaspsamm.org/model/governance/education-and-guidance/ (retrieved: 10/25/2022).

[33] OWASP, *Security testing*. [Online]. Available: https://owaspsamm.org/model/verification/security-testing/ (retrieved: 10/25/2022).