Side Channel Monitoring for Fuzz Testing of Future Mobility Systems

Philipp Fuxen
Dept. Informatics and Mathematics
OTH Regensburg
Regensburg, Germany
Email: Philipp.Fuxen@oth-regensburg.de

Murad Hachani
Dept. Informatics and Mathematics
OTH Regensburg
Regensburg, Germany
Email: Murad.Hachani@oth-regensburg.de

Jonas Schmidt
Dept. Informatics and Mathematics
OTH Regensburg
Regensburg, Germany
Email: Jonas.Schmidt.lth@t-online.de

Philipp Zaumseil
Dept. Informatics and Mathematics
OTH Regensburg
Regensburg, Germany
Email: Philipp.Zaumseil@st.oth-regensburg.de

Rudolf Hackenberg
Dept. Informatics and Mathematics
OTH Regensburg
Regensburg, Germany
Email: Rudolf.Hackenberg@oth-regensburg.de

***Abstract*—The current transformation in the automotive industry is leading to new technologies with a higher software content, a higher degree of networking, and connections to cloud services. This development leads to an increase in the attack surface and the potential extent of damage. ISO/SAE 21434 and UNECE WP.29/R155 were published to address this development. The ISO/SAE 21434 proposes fuzz testing as a measure. In fuzzing, so-called fuzz data is generated and transmitted to a device under test to identify previously unknown and known vulnerabilities. This approach is already being used very successfully in other industries. But in the automotive sector, some challenges arise when testing hardware-related electronic control units. These include the fact that the internal system structures are often poorly known or not known, as well as the severely restricted access and hardware limitations for monitoring. One way to solve these challenges is to use side-channel information to monitor the device under test. Such information includes power consumption, temperature, and noise levels, for example. In this paper, we present a fuzz testing experiment to determine anomalies, data, and requirements for analyzing various side channels. Basic procedures were used to generate the fuzz data. Monitoring of the device under test was performed manually at the beginning. In addition, a side-channel measurement system with various measurement devices and a test setup are presented. Based on the identified fuzz messages, the behavior of the respective side channels during the abnormal behavior is analyzed and described.**

Keywords*—*Fuzzing; Fuzz Testing; Automotive; Cybersecurity; Side Channel Information; Measurement System.

I. INTRODUCTION

The four major themes of future mobility - Connected, Autonomous, Shared, and Electric - have brought a transformation in the automotive industry [1]. New technologies with high software content and a high degree of networking have become established. In addition to the added value, however, this also leads to new risks. The vehicle has evolved into a highly networked system which is connected with multiple cloud services, and whose attack surface has grown significantly. This has increased the probability of becoming the target of an attack on cybersecurity. In the past, attackers had to gain physical access to a vehicle to manipulate it. Today, remote access to a vehicle can be carried out through a communication channel or a cloud backend. In addition to increasing the probability of occurrence, this also means that the extent of damage is significantly greater because attacks could be extended to fleets of vehicles.

In recent years, the international standard ISO/SAE 21434 and the european standard UNECE WP.29/R155 have been developed to address this issue [2][3]. The ISO/SAE 21434 defines a framework of technical requirements for cybersecurity and risk management to ensure the cybersecurity of motor vehicles throughout their life cycle. It covers concepts, product development, production, operation, maintenance, and decommissioning of Electric / Electronic (E/E) vehicle systems. A test method proposed by ISO/SAE 21434 that is suitable for automated execution is the so-called fuzz testing. It is already used successfully in other industries and makes it possible to identify even previously unknown weaknesses and vulnerabilities [4]. A fuzz tester generates so-called fuzz data and transmits it to the System Under Test (SUT) or Device Under Test (DUT). Some fuzzers look for faults and anomalies while the SUT/DUT processes the fuzz data. The goal is to find out what fuzz data causes unwanted system behavior. The data is then analyzed to see if there is a vulnerability. To use fuzz tests automatically and efficiently for automotive systems, it is necessary to detect abnormal behavior of the DUT. This is particularly difficult for automotive Electronic Control Unit (ECU) because there is often little or no knowledge of the internal processes during testing. In addition, their monitoring is a challenge due to highly restricted access and hardware limitations. So-called black box methods are therefore particularly relevant in the automotive sector. Compared to white box or grey box methods, no initial information about the DUT is required.

The main goal of the paper is to improve black box protocol

 ISBN: 978-1-68558-044-5

fuzz testing for hardware-based automotive systems using side channel information. For this reason, the following research questions are addressed:

RQ1: How can fuzz messages be found which have led to abnormal behavior?

RQ2: Which side channels are suitable to use for automotive ECUs?

RQ3: How can this side channel information be measured and used?

The structure of the paper begins with related work in Section II. Section III presents the general conditions of an experiment and its setup as well as its results. In Section IV, a side channel measurement system is described that is designed to solve the challenges of hardware-based fuzzing. The paper ends with a conclusion and future work in Section V and Section VI.

II. Related Work

During fuzz testing of complex and networked vehicle systems, obtaining information about the state of the ECU is difficult. The main reason for this is that access to the ECU is limited and very few information channels are open or available. One method for monitoring during fuzz testing is the so-called side channel information, which has already been successfully used in other areas.

A. Side Channel Analysis

Side channel analysis is a technique for detecting vulnerabilities in a system by analyzing information that can be measured through side channels. D. Agrawal et al. [5] present multichannel attacks, i.e., attacks that use multiple side channels. These attack types use more than one side channel, e.g., energy and Electromagnetic Fields (EMF), in parallel. Based on their analysis, they show that using multiple channels is better for template attacks by experimentally demonstrating a threefold reduction in error probability. In this work, the transfer to ECUs was performed by connecting a large number of side channels. In particular, the analysis of the temperature and electromagnetic radiation of the power showed clear results for reverse engineering cryptographic functions [6][7]. Therefore, the application of these side channels found use in our setup right at the beginning.

B. General Fuzzing

As stated in Section I, fuzzers can be classified into three test procedures based on their knowledge about the system: black box, grey box, and white box fuzzing [8][9]. The following papers present various fuzzing approches from the diffrent test procedures.

M. Böhme et al. [10] present a comprehensive synthesis of the open challenges and opportunities associated with fuzzing and symbolic execution techniques. These problems were identified through a discourse between researchers and practitioners during a Shonan meeting and confirmed by a follow-up survey. They used the term human-in-the-loop for an issue, defined as the work effort that an test auditor has to perform within a semi-automatic fuzzing loop. This was also confirmed by their survey, where 71% of the participants indicated that there is potential to improve the automation of such fuzzing mechanisms. As we move forward, we focus on addressing this issue by automating most of the evaluation steps with Artificial Intelligence (AI) in our future work.

L. McDonald et al. [9] synthesize the current state of the art in fuzzing approaches, classify these approaches, and highlight key insights into the current state of research as well as current challenges. After comparing the current state of the art in fuzzing methods, including hybrid fuzzing, which combines a static analysis of the program and the discovering of bugs during runtime, symbolic execution, which discovers new execution paths by tracking symbolic inputs and machine learning approaches. They continued their future work by highlighting the threats associated with the transition to cyber-physical systems, such as fully automated cars and smart power grids. They presented several options that extend fuzzing as a useful test technique. In the context of embedded systems, they explained that fuzzing using side channels is a suitable mechanism to make black box testing more efficient. In Section II-C we will focus on these approaches and afterward try to research adaptions to the automotive sector in Section II-D.

C. Side Channel Fuzzing

The increasing popularity of efficient fuzzing methods in the embedded systems domain is leading to the identification of new barriers to test automation. These often include a lack of Input / Output (I/O) capabilities, limited computational capacity, and the lack of an operating system in most cases [11][12]. In combination, this results mainly in a black box view of the system. However, in order to implement the three-stage process of fuzzing and thus guarantee a higher level of fault detection, feedback information in the form of side channels has to be applied. These can subsequently be supplied to the fuzzer as input in order to be able to continuously adapt to subsequent test cycles. P. Sperl et al. [11] present a new approach to extract feedback for fuzzing on embedded devices using the information on the power consumption leaks. They carried out their proof of concept by fuzzing synthetic software and a lightweight Advanced Encryption Standard (AES) implementation running on an ARM Cortex-M4 microcontroller. Focusing on detecting various vulnerabilities in an ECU and combining different side channels, less cryptographic analysis of side channel information was completed and more emphasis was on detecting unexpected anomalous behavior.

D. Fuzzing of Automotive ECUs

The implementation of fuzzing within the Controller Area Network (CAN) protocol is an area that has already been widely represented. The publication by P. Patki et al. [13] discusses the importance of penetration testing (pen testing) in finding vulnerabilities in enterprise and automotive networks. The proposed fuzzing tool uses a mutation-based approach for invalid input creation for CAN. The mutation-based approaches use seeds, which are initial inputs, and then

 ISBN: 978-1-68558-044-5

modify them according to a mutation algorithm. According to H. Lee et al. [14], no form of prior knowledge was necessary for fuzzing the CAN protocol. They describe a two-step process that involved, on the one hand, scanning and analyzing the CAN traffic on the CAN bus and, on the other hand, forming completely randomized messages. The packets were injected into the CAN bus via a Bluetooth interface. They were able to attack sophisticated ECUs and change the behaviors of the vehicle. Without intelligent mechanisms for automated evaluation, these procedures require a high degree of manual observation and analysis. We are trying to improve process capability by introducing a fully automated cycle for test automation. M. Dunne et al. [15] introduced a so-called hardware-in-the-loop system for fuzzing a CAN-connected system. For this purpose, monitoring of the power trace was implemented as a side channel. They demonstrate that this black box approach can be used to detect responses to messages.

In summary, through our extension of the side channels, which are specified in Section IV, and the resulting enrichment of the system's feedback information, we aim to increase the coverage of detectable error sources and vulnerabilities. Through the automated cycle, we take the approach of reducing manual analysis to increase data throughput and speed.

III. Experiment

In this Section, we describe the implementation of the fuzzing experiment and its results. It was conducted to collect anomalies and data for later evaluation of the side channels. Furthermore, requirements for the implementation of a fuzzer are collected during the execution. The fuzzing of the CAN channels of an automotive ECU is started. In this process, the ECU was observed manually and with basic analysis methods. An anomaly was detected when the ECU behaved in a way that deviated from the normal operating state.

A. Hardware Setup

In order for the experiment to be carried out, the hardware must first be connected to the automotive ECU. Therefore, it must be supplied with voltage on the one hand and the CAN bus must be connected to a computer on the other. For the connection between the computer and the CAN bus of the ECU, a so-called CAN-to-Universal Serial Bus (USB) interface was used. Specifically, the OWASP Automotive EMB 60 was used, which is available as an open-source project [16]. Therefore, hardware and software can be accessed. It provides two Controller Area Network Flexible Data Rate (CAN FD) channels connected via a single D-sub 9 connector. Connection to the computer is via a USB 1.0/2.0 type B connector.

As shown in Figure 1, in addition to the computer and the CAN interface, a laboratory power supply is also needed to ensure the supply voltage of the control unit. To connect these, the corresponding connection pins on the control unit for Voltage Common Collector (VCC) and Ground (GND) were measured with a multimeter. These were then wired to the laboratory power supply with VCC and GND. A residual

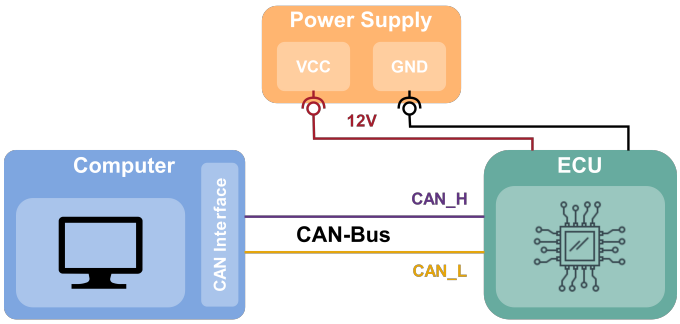

Figure 1. Hardware Setup.

bus simulation or similar is not required for the control unit, as it is integrated into a dedicated experimental setup.

B. Fuzzing Test

For carrying out the fuzzing test, some software is also required on the computer. A Linux distribution serves as the operating system. To use the CAN-to-USB interface, the kernel interface SocketCAN is utilized. To be able to run SocketCAN via the Linux console for the CAN functionality, the package can-utils is installed. The Python programming language is suitable for programming fuzzing scripts. The Python package python-can enables the CAN functionality.

At the beginning of the experiment, so-called random fuzzing was implemented with the Python library python-can. Randomly generated CAN frames were sent to the ECU. In addition to the basic random fuzz tests, protocol-specific areas were analyzed and tested with random values according to the limits. The random analysis of individual features and communication paths of the CAN protocol already provided initial results. The monitoring of the system behavior was carried out manually during the experiment. For this purpose, the dedicated experimental setup of the ECU was observed, equipped with several infotainment displays and an instrument cluster. These indicate changes when the fuzzy CAN frames are sent. Moreover, the breakdown of the displays indicates a potential crash of the system.

In addition to the self-implemented random fuzzing, tests were also carried out with Scapy. Scapy is a python library, which is utilized for the manipulation and analysis of network packages. Caring Caribou, which is an open source fuzzer, was also used to identify anomalies. Its fuzzer module has three modes for generating fuzz data: Random, brute, and mutate. Besides generating fuzz data, Caring Caribou also has a function to identify messages that have triggered anomalous behavior. For identification, the transmitted fuzz messages are played back block by block. The user can indicate whether the observed behavior occurred in a block. If this is the case, the block is divided into smaller blocks and replayed. This process continues until the message is identified [17][18].

The fuzz messages identified and classified as anomalies form the basis for further work. With the help of the detected abnormal behaviors, the side channel information can be

 ISBN: 978-1-68558-044-5

analyzed and training data for AI models can be generated. Identifying the relationship between a fuzz message and the associated abnormal behavior proved to be very tedious without a monitoring system. Therefore, an improvement is necessary through the monitoring of side channels.

IV. Side Channel Measurement System

The optimization of new application-oriented fuzzing mechanisms for automotive ECUs, as well as the consideration of these hardware-related systems as a black box, requires the extraction of information through the connection of side channels. Previous methods and applications of fuzzing often involved the self-performed observation and evaluation of system responses to classify a vulnerability based on the results. To ensure a standardized and secure development process according to ISO/SAE 21434, an automated approach is essential. Only an automated evaluation and adaptation of the fuzzing and its results enables a high coverage of the detectable vulnerabilities. Based on the assumption that the efficiency and correctness of our fuzzing unit increase proportionally with the amount of information that can be extracted from the system in the form of side channels, we used further side channels for the analysis of the DUT in addition to the established analysis of the power traces.

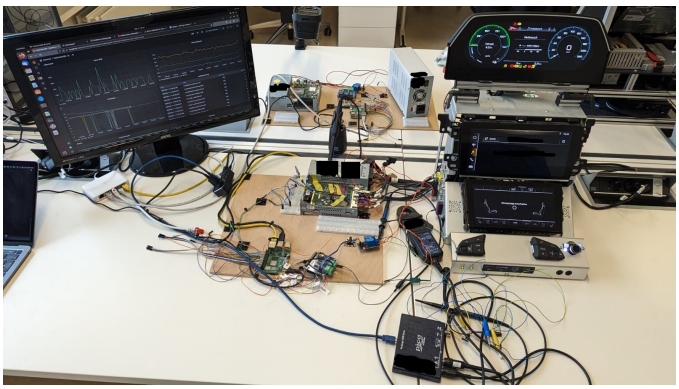

Figure 2. Side Channel Measurement.

Figure 2 shows our setup with the measuring devices included. The ECU is visible in the center, which is connected to the rest infotainment system and to the side channel measuring devices. These are described in the following sections. In addition to the measurement setup, the Grafana dashboard can also be seen, as it is described in more detail in Section IV-G.

A. Power

The analysis of power traces as a side channel achieved remarkable results, especially in the field of cryptography. For reverse engineering applications, the analysis of this side channel proved to be a useful instrument to draw conclusions about program structures and functional flows. In the context of fuzzing and the consideration of the DUT as a black box, the detection of anomalies in the power trace and the interpretation of the fuzzer's input serve to uncover vulnerabilities.

The acquisition of the power trace is done by connecting an oscilloscope. The current intensity is measured, which the control unit requires during the input phase of data generated by the fuzzer. With randomized fuzzing, it was already possible to identify a bug by the history of the measurement of the power consumption and the voltage. The visualization in Grafana shows a drop in voltage, which was related to the simultaneous crash of the infotainment displays. After a few seconds, an automatic reset of the ECU could be detected in the measurement. After the restart, the voltage, as well as the power consumption, fluctuated from the normal state.

B. CAN

In addition to the physical data, software, and protocol-specific data can also be recorded. This data is not measured via a sensor but is directly acquired by the system or the communication protocol. In the case of the CAN protocol, several side channel features can be calculated. These include bus load, message frequency per ID, or bit flip rate. These metrics are added to the data stream as measured values.

When monitoring the DUT during fuzz testing, the side channel features of the CAN are used to detect changes in communication behavior. For example, when analyzing an anomaly, the bus load was found to dip and then return to normal shortly thereafter.

C. Thermal Image

The use of infrared images is increasingly applied in a wide range of applications. Due to the rich information content paired with the wide range of areas of application, the use of infrared images proved to be a good tool for the detection of anomalies. Usually, reference values in the form of patterns within images or plain temperature values are used to successfully detect an anomaly. For the adaptation to embedded systems and especially ECUs in the automotive domain, preparations have to take place in the way of exposing components of interest. By exposing components and so-called regions of interest, the focus of anomaly detection can be explicitly set on CAN-related components, for example.

Using direct fuzzing on specific areas of the CAN protocol, significant differences in the heat signatures could already be detected manually. Unusual temperature patterns were measured in the area of the CAN controller and the Central Processing Unit (CPU). The temperature increased significantly compared to the normal state.

D. Temperature

The need to expose components leads to a reduction of the potentially measurable area. In order to compensate for this limitation of the thermal camera, temperature sensors offer a remedy. By subdividing the DUT into measuring ranges and placing individual sensors in a controlled manner, maximum measurement coverage of relevant areas can be achieved. Combined with visual patterns, a larger data space for the temperature side channel is created, facilitating the contribution and application of AI techniques.

 ISBN: 978-1-68558-044-5

Like the detection by infrared images described previously, the same effect could be measured by the stationarily installed temperature sensors. A clear increase could be analyzed, which also deviates considerably from the normal state.

E. Visual Image

The input to an ECU by fuzzer-generated messages produces different reactions depending on the communication protocol and the task of the ECU. Manual examination of the ECU's response and reaction to fuzzer-generated messages allows an anomaly to be accurately detected, yet the effort involved is too high. Visual and automated examination by connecting a camera that monitors the system can substitute this process. The functionality for anomaly detection is limited to the detection of movements, changes in structure, and other visual features. Relevant values for recording are the image sequence, which was identified as an anomaly, and a value that indicates an anomaly.

As described in Section IV-A, it was possible to see a parallel to the crash of the displays and an anomaly within the measurements. This connection could be created by performing an observation of the outputs on the displays of the test bench. Through automated observation and anomaly detection, the following anomalies, brought about by random fuzzing, could be detected. Firstly, flickering and abnormal changes in driving modes could be detected. Secondly, repeated crashes could be detected by the camera.

F. Acoustic

Besides the already mentioned side channels, there are also indicators that can be recorded via the acoustic channel. The rotation of the ECU fan can indicate the processor load. After all, as computing power increases, so does the temperature of the CPU, which is cooled by the controller's fan. Consequently, an unexpectedly high fan speed is synonymous with a code execution anomaly. Since the fan produces a certain noise depending on the workload, measuring this noise is a way to draw conclusions about the fan's speed. Noise and other irritations can be avoided by placing a microphone next to the fan. The rotation and resulting noise of the fan thus represent the dominant frequency in the measurement. This can then be isolated and analyzed without adding other noise. Performing fuzzing randomly also led to the detection of an anomaly in the fan's measurement. When the anomaly occurred, the fuzzer gradually increased the speed of the fan. The increase in speed from normal was not stopped when the anomaly occurred. Accordingly, the fan remained in this mode even after several hours without resetting.

G. Visualization

The connection of a wide variety of side channels led to increased complexity in the evaluation of the combined side channels. The uniform and central data collection as a data lake is the basic building block for subsequent analyses. The data lake was implemented in the form of an Influx database, which enables the transfer of measured values in near real time. Based on the visualization of the data with Grafana, as shown in Figure 3, first explorative analyses can be performed on the measurement data. In this way, initial findings could be obtained. In addition, the centralized data storage achieves preparation for further analysis methods for anomaly detection (Machine learning and deep learning).

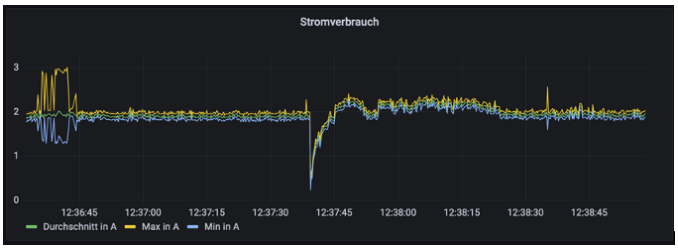

Figure 3. Power Consumption Widget of Grafana Dashboard.

Figure 3 represents exemplary the visual processing of the measured power consumption as a time series. A visually identifiable anomaly can be seen within the depicted time frame. The fuzzing of the ECU started at time 12:36:30 and caused a shutdown, which triggered at time 12:37:40. After rebooting the ECU, a normal condition could not be established. This anomaly can already be detected by applying simple algorithms within the measurement system and offers the possibility to generate a dedicated feedback for the fuzzer. In addition to the various physical side channel data, CAN messages are also recorded to find correlations between anomalies and received messages.

V. Conclusion

The current turnaround in the automotive sector is leading to the introduction of new technologies with significantly more software and connectivity. This increases the attack surface and the damage potential. One standard that counteracts this development is ISO/SAE 21434, which regulates the cybersecurity of vehicles over their entire life cycle. One of the measures it proposes is fuzz testing. In other industries, fuzz tests are already being used very successfully. However, in the automotive sector, some challenges arise due to hardware-related ECUs.

The approach taken in this paper aims to solve these problems using side channel information processing. These are already being used successfully in several areas. Therefore, the Section on related work has been divided into the following structure: side channel analysis, general fuzzing, side channel fuzzing, and fuzzing of automotive ECUs. In the subsections, various relevant approaches have been discussed, which are in the context of this paper.

To collect anomalies, data, and requirements for evaluating the different side channels, a fuzzing experiment was conducted.CAN was used as the communication protocol to be fuzzed. Fuzz data generation was performed using a self-programmed random fuzzer and the two frameworks Caring Caribou and Scapy. The monitoring of system behavior was performed manually for the time being. Detect the relationship

 ISBN: 978-1-68558-044-5

between a fuzz message and an anomaly is tedious. The discovered fuzz messages and the corresponding anomalies are used for the later analysis of the side channels and for the dataset creation.

A measurement system with several sensors and interfaces was built to measure side channel data. This was connected to a test setup. Methods, such as the analysis of power and temperature were used. To achieve a high degree of coverage of temperature information, temperature sensors were used in addition to a thermal imaging camera. These were placed in areas that could not be detected by the thermal imaging camera. In addition, a microphone was installed in such a way that the frequency of the control unit fan was recorded. A camera was used to record the multimedia displays and the instrument cluster of the test bench. With a CAN interface, various bus-specific side channel information could be obtained. By combining the different side channels, the information content is increased because not every abnormality is noticed on every side channel. The detection of anomalies during monitoring thus has a broad database.

VI. Future Work

Since the current measurement system implements only part of the fuzzing cycle, further process steps must be performed to complete the fuzzer. Based on the data from the measurement system, static analyses are first performed. After these analyses, more intelligent methods (Machine learning and deep learning) for monitoring the fuzzed DUT will be investigated and implemented.

In the next phase, fuzz data generation will be extended from random generation and block-based generation to feedback-based generation. To this end, the fuzz data generation will be adjusted according to the feedback from the monitoring system to find anomalies more efficiently. This is to achieve deeper program structures and the system architecture.

References

[1] U. Z. Abdul Hamid, *Autonomous, Connected, Electric and Shared Vehicles: Disrupting the Automotive and Mobility Sectors*. Warrendale, Pennsylvania, USA: SAE International, Oct. 2022, ISBN: 978-1-4686-0347-7.

[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] N. Besic, *Fuzzing: The Next Big Thing in Cybersecurity? - Bright Security*, May 2022. [Online]. Available: https://brightsec.com/blog/fuzzing/ (retrieved: 2023-06-08).

[5] D. Agrawal, J. R. Rao, and P. Rohatgi, "Multi-channel attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2003*, ser. Lecture Notes in Computer Science, vol. 2779, Springer, Sep. 2003, pp. 2–16. DOI: 10.1007/978-3-540-45238-6_2.

[6] M. Hutter and J.-M. Schmidt, "The temperature side channel and heating fault attacks," in *Smart Card Research and Advanced Applications - CARDIS 2013*, Nov. 2014, pp. 219–235, ISBN: 978-3-319-08301-8. DOI: 10.1007/978-3-319-08302-5_15.

[7] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *Cryptographic Hardware and Embedded Systems - CHES 2001*, ser. Lecture Notes in Computer Science, vol. 2162, Springer, May 2001, pp. 251–261. DOI: 10.1007/3-540-44709-1_21.

[8] J. Li, B. Zhao, and C. Zhang, "Fuzzing: A survey," *Cybersecurity*, vol. 1, no. 6, Jun. 2018. DOI: 10.1186/s42400-018-0002-y.

[9] A. Barkworth, L. Mcdonald, and M. Ijaz Ul Haq, "Survey of software fuzzing techniques," Dec. 2021.

[10] M. Böhme, C. Cadar, and A. Roychoudhury, "Fuzzing: Challenges and reflections," *IEEE Software*, vol. 38, no. 3, pp. 79–86, 2021.

[11] P. Sperl and K. Böttinger, "Side-channel aware fuzzing," in *Computer Security–ESORICS 2019: European Symposium on Research in Computer Security*, Springer, vol. 24, Sep. 2019, pp. 259–278, ISBN: 978-3-030-29958-3. DOI: 10.1007/978-3-030-29959-0_13.

[12] M. Muench, J. Stijohann, F. Kargl, A. Francillon, and D. Balzarotti, "What you corrupt is not what you crash: Challenges in fuzzing embedded devices," Jan. 2018. DOI: 10.14722/ndss.2018.23176.

[13] P. Patki, A. Gotkhindikar, and S. Mane, "Intelligent fuzz testing framework for finding hidden vulnerabilities in automotive environment," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, vol. 4, IEEE, 2018, pp. 1–4.

[14] H. Lee, K. Choi, K. Chung, J. Kim, and K. Yim, "Fuzzing can packets into automobiles," in *IEEE International Conference on Advanced Information Networking and Applications*, vol. 29, IEEE, 2015, pp. 817–821.

[15] M. Dunne and S. Fischmeister, "Powertrace-based fuzzing of can connected hardware," in *2022 IEEE International Conference on Cyber Security and Resilience (CSR)*, IEEE, 2022, pp. 239–244.

[16] A. Meisel, *Owasp automotive emb 60 — owasp foundation*. [Online]. Available: https://owasp.org/www-project-automotive-emb-60/ (retrieved: 2023-06-08).

[17] mjidhage, kasperkarlsson, TobLans, *et al.*, *Documentation for caring caribou*. [Online]. Available: https://github.com/CaringCaribou/caringcaribou/blob/master/README.md (retrieved: 2023-06-08).

[18] P. Biondi, *Scapy: The python-based interactive packet manipulation program & library*. [Online]. Available: https://scapy.readthedocs.io/en/latest/index.html (retrieved: 2023-06-08).

 ISBN: 978-1-68558-044-5