Work-in-Progress: Emerging E/E-Architectures as Enabler for Automotive Honeypots

Niclas Ilg[*‡], Dominik Germek[*], Paul Duplys[†], and Michael Menth[‡]
[*] *Corporate Research, Robert Bosch GmbH, Germany, {niclas.ilg, dominik.germek}@bosch.com*
[†]*Sector Mobility, Robert Bosch GmbH, Germany, paul.duplys@bosch.com*
[‡]*University of Tuebingen, Chair of Communication Networks, Germany, menth@uni-tuebingen.de*

Abstract—**The surge towards autonomous driving causes a growing need for software and connectivity inside the vehicle. This growth in complexity poses a substantial challenge to the security of vehicles as it expands the cyber attack surface. Moreover, to cope with computational demands, the automotive industry currently develops new in-vehicle architectures. In this context of emerging technologies and escalating complexity, ensuring cybersecurity is an increasingly demanding task. Honeypots are a well-established tool for threat intelligence and intrusion detection in traditional IT and the Internet of Things. In the automotive domain, however, honeypots have never found their way into practical application. In this work, we highlight how emerging in-vehicle architectures present opportunities for honeypot deployments inside the vehicle and threat landscape monitoring on the Internet. In contrast to existing research, we consider emerging in-vehicle architectures and how functional limitations from the automotive industry have prevented the widespread use of honeypots in the past.**

1. Introduction

Advanced driver assistance systems, the surge towards Autonomous Driving (AD), and the integration of cloud services demand enhanced connectivity between vehicles, back-ends, and the Internet. Moreover, due to the amplified computational demands of these features, Original Equipment Manufacturers (OEMs) have initiated a transformation of in-vehicle architecture; once static and deeply integrated vehicles will gradually evolve to resemble a smartphone on wheels in the future. Unfortunately, pervasive connectivity and the increasing software complexity—approximately 300 million lines of code are projected for the vehicle of 2030 [1]—go hand in hand with an expanding cyber attack surface. Even in the past, white-hat researchers have repeatedly exploited infotainment and telemetry units remotely [2]–[4], showcasing the possibility of causing fatal accidents. For this reason, it is vital to continuously monitor the automotive threat landscape and detect new approaches to vulnerabilities, zero-day exploits, and potential malware targeting vehicle computers [5].

In traditional IT as well as in the Internet of Things (IoT), honeypots are an established tool to support threat landscape monitoring. These decoy systems mimic valuable services, devices, or entire networks to attract adversaries. Adversarial activity on the honeypot is captured and analyzed to subsequently update existing security measures. Even in the automotive sector, honeypots have been a subject of research for some time; back in 2008, Verendel et al. published a now well-known approach to in-vehicle honeypots and their simulation [6].

However, honeypots never found widespread application in the automotive industry. Therefore, we first discuss the limitations that currently prevent the success of automotive honeypots—a honeypot that is either placed in a vehicle or mimics a vehicle unit. Further, we consider whether technological advances and the development of new in-vehicle architectures can enable a widespread deployment of honeypots in the automotive domain. Thus, our contribution is as follows:

- We discuss the current state of automotive honeypots as well as the limitations that prevent their actual use.
- We illustrate how emerging vehicle Electrical/Electronic (E/E)-architectures enable automotive honeypots, especially, in-vehicle solutions.
- We discuss the different requirements of production and research honeypots for automotive purposes.
- We highlight how cloud-deployed honeypots can support threat landscape monitoring in the automotive domain.

The remainder of this paper is structured as follows. In Section 2, we discuss related work on automotive honeypots. Section 3 provides the reader with a background on honeypots and automotive E/E-architectures. In addition, we discuss the factors that previously hindered the deployment of automotive honeypots. Section 4 highlights the opportunities that emerging E/E-architectures hold for in-vehicle honeypots. Furthermore, we discuss in Section 5, how cloud-native honeypot deployments aid automotive threat landscape monitoring and the research of attacker behavior. Section 6 concludes the paper.

2. Related Work

The authors of [6] propose an approach to in-vehicle honeypots that fully simulates the in-vehicle network. The honeypot is deployed on hardware separate from the in-vehicle network including its own wireless gateway to ensure the vehicle's secure operation. Furthermore, the authors present three strategies to simulate the in-vehicle network to provide a realistic honeypot environment. They also discuss the limitations of their approach which we will cover in Section 3.5.

In [7] the authors discuss how honeypots are an important aspect of a defense-in-depth strategy. Defense-

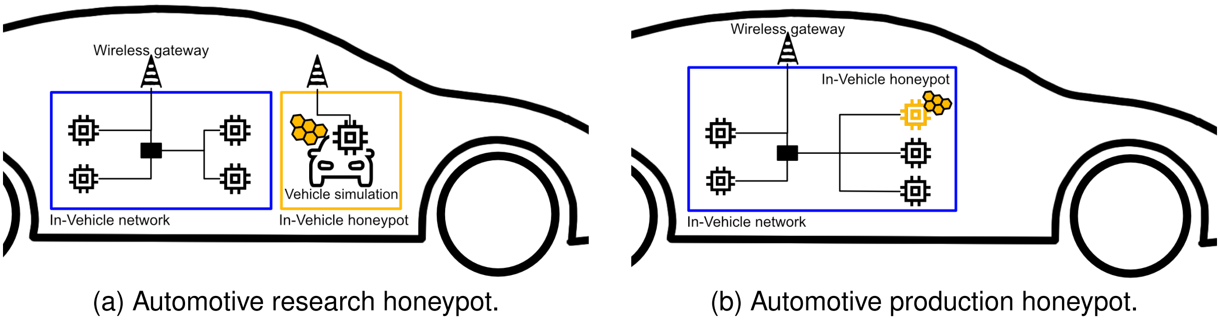


Figure 1. Research and production honeypot concepts transferred to the automotive domain.

in-depth is an approach to security that utilizes security measures at multiple layers of a system or network, not only at the entrance point. In this multi-layer approach honeypots are used to gather intelligence and defenders adjust other security measures with the acquired knowledge. The authors also highlight their central challenge of implementing an in-vehicle honeypot: providing a credible simulation while separating it from the genuine in-vehicle network.

The authors of [8] discuss the deployment of honeypots in vehicular ad-hoc networks which are used for vehicle-to-vehicle and vehicle-to-infrastructure communication. In particular, they expect roadside infrastructure honeypots to be more accessible to adversaries as these are stationary in contrast to moving in-vehicle honeypots.

In [9] the author discusses approaches for vehicle honeypots. They propose honeypot deployment in the form of OBD-II dongles (remote diagnostic) and honeypot nodes in vehicular ad-hoc networks. Furthermore, they touch on Verendel et al.'s simulation strategies; simulation quality could be improved by using genuine in-vehicle data until the adversary performs malicious tasks at which point a simulation is required.

In [10] the authors propose a framework that uses machine learning to determine honeypot configurations for application in the Internet of Vehicles. They use the Common Vulnerability Scoring System score of Common Vulnerabilities and Exposures that would be deployed in the automotive honeypot as a metric for a defender's insights and an adversary's time investment.

The authors of [11] investigate a feasible way of utilizing honeypots inside the vehicle. They outline requirements for in-vehicle honeypots and give recommendations on, among others, the placement and interaction level of the honeypot. During their work, the authors identify a fundamental difficulty: honeypots are difficult to integrate into the car while keeping hardware costs and risk to the rest of the network low. Both these limitations are part of our discussion in Section 3.5, and we highlight how future E/E-architectures solve these issues in the remainder of this paper.

In contrast to existing work, we present approaches to honeypot deployment that not only consider but are enabled by emerging vehicle architectures. We also highlight the limitations of existing approaches. Some of these constraints have originated within the automotive industry, making them challenging to anticipate for those outside the domain. Finally, we present honeypot strategies to aid both, threat intelligence and intrusion detection.

3. Background

In this section, we provide the necessary background regarding honeypots and, especially, honeypots in the automotive domain. In addition, we highlight the transformation of automotive E/E-architectures, which are designed to enable ubiquitous connectivity and AD.

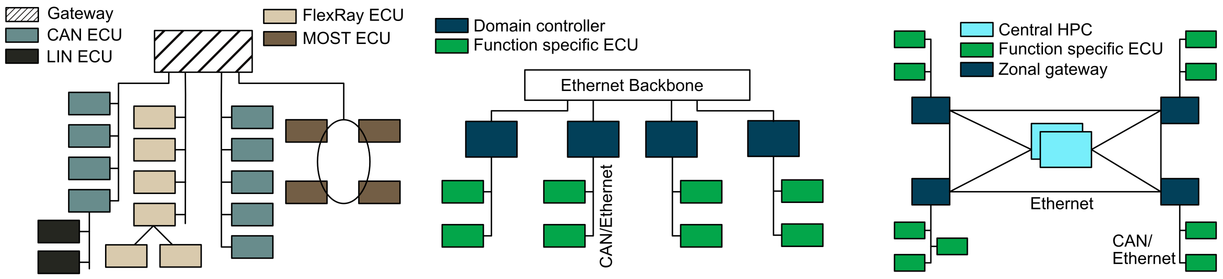
3.1. Honeypots

Honeypots are decoy resources that are deployed to mimic a valuable target system. They can mimic simple network-facing services as well as entire systems. Activities of enticed adversaries are logged for subsequent analysis of the attacker's approach. The quality of the received data depends on how convincing the honeypot appears to the attacker and how many interaction possibilities it allows [15]. Since honeypots do not serve any purpose to legitimate clients, it can be assumed that all incoming connection attempts are hostile.

Honeypots are generally categorized by their interaction level as follows [16]. *Low-interaction honeypots* usually just mimic network-facing services, e.g., Secure Shell or Network Time Protocol. They only provide very basic interaction possibilities like a log-in shell or request-response communication. *Medium-interaction honeypots* are more refined decoys as they additionally simulate parts of the internal functionality; simulating the operating system of a valuable target can get the adversary to further interact with the honeypot after a successful login and reveal more sophisticated attacks. While low- and medium-interaction honeypots are only computer programs, *high-interaction honeypots* are genuine systems that are left to the adversary's hands to provide a realistic playground. Although high-interaction honeypots promise the best insights into the adversary's attack path, they also require the highest maintenance effort as genuine systems are often used for subsequent attacks.

3.2. In-Vehicle Honeypots

There is an important distinction between two types of in-vehicle honeypots: on the one hand, a honeypot that is entirely *separated from the in-vehicle network* as illustrated in Figure 1a. This serves the purpose of learning about an attacker's behavior inside an automotive unit or



(a) Gateway architecture [12]. (b) Domain controller architecture [13]. (c) Zonal architecture [14].

Figure 2. Logical illustration of current and future E/E-architectures.

network [6]—a research honeypot. On the other hand, true in-vehicle honeypots are *part of the in-vehicle network* and mimic, for example, a valuable Electronic Control Unit (ECU). Figure 1b illustrates such a setup. In this way, defenders get alerted as soon as an adversary breaches into the network and investigates the systems inside. In traditional IT, the latter is called a production honeypot.

Placing the research honeypot inside the vehicle, despite being completely separate from the in-vehicle network, is primarily motivated by simulation quality and credibility. Real vehicle data ranging from a driver's input to environment perception can be used to enhance the simulation quality [6], [9]. Furthermore, to provide credible simulation and device behavior, the honeypot should be mobile [6] and reachable only at realistic cycles; a 24-hours-a-day vehicle up-time triggers suspicion. Nevertheless, as discussed in Section 3.5, there are limitations to in-vehicle solutions for research purposes. Thus, we propose a cloud-native approach to support automotive threat intelligence with honeypots in Section 5.

3.3. In-Vehicle Architectures

The in-vehicle network consists of ECUs that are connected with a central gateway by buses, e.g., Controller Area Network (CAN), Media Oriented Systems Transport (MOST), Local Interconnect Network (LIN), or FlexRay. Most current vehicles have a very distributed E/E-architecture. Figure 2a illustrates this so called gateway architecture. It has limitations in computing resources, complexity, and a distributed connectivity approach; individual ECUs like infotainment units or telemetry units handle connectivity to back-end services by themselves. Current architectures are not only incredibly complex but very heterogeneous as, for the most part, different ECUs are built by separate suppliers, introducing a lot of variance in hardware components and software solutions even inside a single vehicle.

As illustrated in Figure 2b and 2c, future architectures are moving towards more *centralized* approaches. The Domain Controller architecture is the first step towards centralized computing resources. Domain controllers handle the majority of computation for their respective domains. AD, however, will require even more powerful computing units. Thus, central High Performance Computers (HPCs) will eventually take over most computing and connectivity. Resulting in the Zonal architecture. This concept also separates hardware and software planes to allow for independent development and easier update mechanisms.

3.4. Development Towards Large-Scale Remote Attacks on Automotive Units

At present, knowledge of large-scale remote attacks on vehicle units is limited. Mostly white-hat researchers were able to remotely exploit infotainment systems and telemetry units. While the automotive industry as a whole is already a target of large-scale remote cyber attacks, attackers currently focus on the back-end and IT infrastructure of OEMs and suppliers [17]. Vehicle units are more so targeted in the event of theft. However, it stands to reason that the increasing connectivity and computing power in future vehicles will also lead to large-scale remote attacks against vehicle computers.

Especially the heterogeneous environment of hardware and software components is a major contributor to the current sparse attack landscape. However, in other domains, e.g., traditional IT and the smartphone market, we have seen a strong consolidation of software solutions over time. As more and more technology transfers from those realms into future E/E-architectures, we expect a similar outcome in the automotive industry. The limited number of software components allows adversaries to focus on a few Operating Systems (OSs) to target entire fleets of vehicles.

3.5. Limitations of Prior Approaches

In traditional IT, honeypots are a well-established tool for threat intelligence, collecting malware samples, and getting early warnings on novel exploits as well as zero-day attacks. As discussed in Section 2, even in the automotive domain there is research on honeypots that dates to 2008. However, automotive honeypots have faced practical issues that prevent actual use:

Accessibility: For a honeypot to collect useful data, it must be accessible to a large number of adversaries. Especially low-range wireless interfaces like Bluetooth or WiFi are generally not exposed to enough attackers. Furthermore, observing large-scale remote attacks is only possible if the honeypot is reachable from the public Internet. However, since vehicles should operate within a private network, the exposure to attacks over cellular networks is close to none. This limited exposure to attackers does not justify the expense of fully simulating an in-vehicle network.

Costs: For obvious reasons, OEMs are very strict about additional expenses. As automotive honeypots have not yet proven that their insights are worth the additional manufacturing, maintenance, and especially hardware costs, there is currently no incentive to spend large amounts of money for an automotive honeypot.

Data acquisition: Although it seems promising to use real vehicle data to improve simulation quality, the source of such data is an important factor. Using the data of customers is not only concerning with regard to privacy but also difficult to impose by OEMs and suppliers.

Prior approaches also have functional limitations caused by restrictions of the honeypot technology and the E/E-architecture:

Simulation quality: Simulating an entire in-vehicle network is a challenging task. Providing a simulation that is able to react to an attacker's input is even more difficult: properly reacting to input that should provoke driver reaction is the main concern. As an attacker, for example, issues acceleration commands it is expected that the driver hits the brakes; a simulation should follow this behavior.

In-vehicle technology: Microcontrollers are strictly tailored to their use case and do not offer additional computing resources for a honeypot. In addition to hardware limitations, current E/E-architectures do not provide the separation between resources nor the possibility to remotely or dynamically react to incidences to justify the use of honeypots.

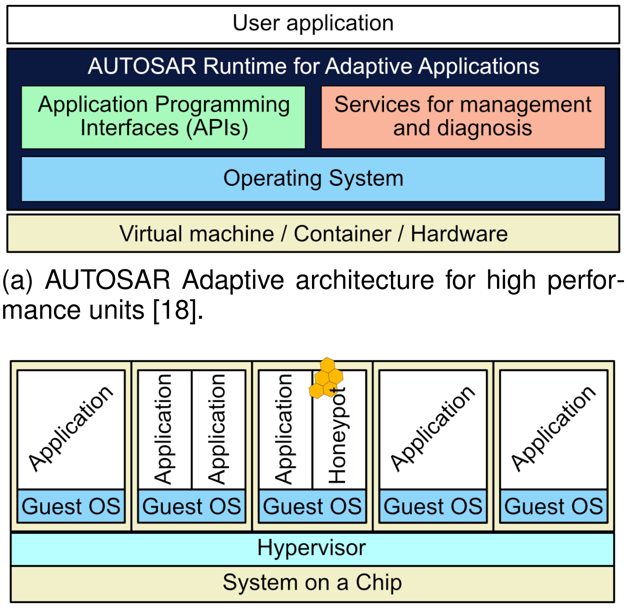
In the following sections, we discuss how upcoming in-vehicle architectures and enhanced connectivity are capable of, at least partially, solving these issues.

4. Opportunities for In-Vehicle Honeypots

As discussed in Section 3.3, vehicle architectures evolve towards a centralized computing approach. The software-defined vehicle is a term often associated with this transformation: the higher computing power, the integration of cloud services, and, especially, the separation of software and hardware layers lay the foundation for AD and an overall improved driver experience. In this section, we highlight how the emerging vehicle architecture also provides opportunities for in-vehicle production honeypots. However, we also discuss remaining limitations with regard to research honeypots that explain why automotive threat intelligence requires honeypot deployments on the public Internet.

4.1. High Performance Computing

High performance units will centralize computing and bring additional performance compared to current architectures as resources are no longer bound to a specific ECU. Furthermore, the AUTOSAR architecture standard for automotive units was overhauled and aims for a software-defined implementation approach for HPC units. As depicted in Figure 3a, the application is no longer tightly coupled with specific hardware but can be deployed on top of hypervisors as a virtual machine or container. This approach allows for more hardware-independent development, easier update mechanisms, and more dynamic deployment.



(a) AUTOSAR Adaptive architecture for high performance units [18].

(b) Hypervised high performance unit that deploys a honeypot application.

Figure 3. Architecture and concept for high performance computers.

Honeypots benefit from all the improvements mentioned above. First, the additional computing resources can power a honeypot application without interfering with the regular operation. Not only do honeypots benefit from computing resources but also from the possibility of container deployments. Resource-efficient base images and compiled programming languages allow the development of honeypots that can be packaged in a container image as small as a few megabytes. Second, the separation of software from hardware layers introduces additional defensive mechanisms. No longer does a corrupted application imply that the entire device is lost to the attacker. Instead, an application that interacts with a honeypot can be isolated from safety-critical functionality, ensuring the integrity and reliability of critical systems. Third, given the limited number of expected adversaries within the in-vehicle network, honeypots benefit from the dynamic deployment possibilities of HPC units. While in day-to-day operation not every vehicle must deploy a honeypot, OEMs can activate additional honeypots dynamically. For example, in the event of heightened security requirements due to published attacks, real-world attacks on other vehicles, or threat intelligence indicating an increased interest in vehicle computers. Figure 3b illustrates such a honeypot setup. Guest OSs are selected depending on the application. Thus, honeypots can be deployed in non-critical environments to not interfere with safety-critical operations.

4.2. Dynamic Networking

In traditional IT, the paradigm of Software-defined Networking (SDN) has not only been a subject of academic research but has also made its way into real networks. It is foreseeable that this technology will eventually be transferred to the automotive industry. SDN offers concepts to dynamically re-configure network participants [19]. It decouples the control plane from the data plane, allowing for a more flexible network architecture. In SDN, a central controller configures the routing of network traffic (control plane), while the switches solely focus on forwarding the actual network packets (data plane). Hence, the SDN controller can define routing policies, prioritize traffic classes, or even redirect network packets. The data plane then implements these instructions.

Logical network separation is a fundamental capability of SDN, eliminating the need for additional technologies such as virtual LAN. This feature is particularly crucial in the automotive domain, where it is essential to separate safety or security-critical components from non-critical ones. While physical separation inside the vehicle is considered best practice, it is not always the norm [2]. Logical network separation is achieved through network flows. The controller, for example, defines flow rules based on the packet's source, destination, or traffic class—thus, a headlight ECU could no longer send CAN messages to the motor control unit [20].

Using Honeypot Findings. Dynamic re-configuration allows to separate devices that are suspected compromised. If an adversary investigates a honeypot system or application, a finding is issued for the corresponding devices. The SDN controller can respond to the report by isolating the suspected rogue component. However, in the case of safety-critical components, a separation from the core network during high-speed driving is not feasible. Thus, in the event of a honeypot finding, we propose to discontinue non-safety related network flows while allowing safety-critical network flows to continue until the vehicle comes to a stop. Depending on the incident, the driver should of course be notified to halt. By utilizing network flows, it becomes possible to mitigate the consequences of compromised applications while ensuring that the vehicle's basic functionality is still maintained and guaranteed.

Further Honeypot Opportunities. SDN not only enables the integration of honeypots and the processing of their findings, but it also presents an intriguing opportunity for honeypot application within the SDN controller. As the central intelligence of the network, SDN controllers are attractive targets for adversaries. Consequently, deploying a honeypot that mimics an SDN controller is a promising application.

4.3. Utilizing Connectivity

The utilization of Over-the-Air (OTA) updates presents a significant opportunity for automotive honeypots. In the past, when automotive honeypots were first introduced in 2008, OEMs had to recall millions of vehicles to address vulnerabilities that may or may not have been exploitable by attackers. However, with the advent of new architectures and increased connectivity, OEMs can now leverage honeypot findings more efficiently by incorporating most security fixes into the next OTA software update. Furthermore, OTA services themselves offer a promising application for honeypots, as these critical interfaces are likely to be targeted by attackers.

4.4. Remaining Limitations

HPC and Ethernet networks provide realistic conditions for in-vehicle production honeypots to add to existing intrusion detection methods. However, we do not see a feasible deployment scenario for in-vehicle research honeypots, as hardware separation, resulting costs, and, especially, accessibility limit the number of deployable honeypots and, especially, their exposure to adversaries. Furthermore, a research laboratory—that deliberately invites adversarial parties—in customer property also has legal limits. In the next section, we will therefore explain how honeypots on the Internet can contribute to automotive threat intelligence.

5. Proposing a Cloud Deployment Strategy

The concept of in-vehicle research honeypots is not practical due to additional hardware costs and customers unwilling to carry a honeypot that would deliberately invite an adversary into their vehicle. For this reason, we propose a *cloud simulation strategy* that moves research honeypots from the vehicle into the cloud. Cloud environments drastically increase attacker exposure. In addition, they enable a versatile use of honeypots with different simulation depths; in this section, we highlight how various types of honeypots can be deployed realistically on the Internet to support automotive threat intelligence. Furthermore, we discuss how honeypots can contribute to threat intelligence even now, with the software-defined vehicle still a considerable distance away.

5.1. Utilizing Low-Interaction Honeypots

Considering the current state of large-scale remote attacks (discussed in Section 3.4), automotive research honeypots should monitor the general interest of attackers in automotive operating systems and services. Due to the unknown number of attackers against vehicle units, the cost and effort required for a credible vehicle simulation are very high. Thus, we propose the deployment of low-interaction honeypots as a first step.

Low-interaction honeypots that mimic automotive operating systems and services offer several advantages over complex simulations. They are proficient in monitoring general interest in a given system with low resource requirements and development costs. Since the limited depth of adversarial interaction requires lesser monitoring and maintenance by defenders, low-interaction honeypots allow for automated deployment. Furthermore, mimicking only operating systems and services offers the advantage of credible Internet deployments; many of them can be found in more Internet-native domains (e.g., QNX in IIoT, MQTT in IoT, or Android in the mobile market). Monitoring such honeypot systems is valuable for automotive threat intelligence, as attacks on industrial QNX systems, for example, are equally dangerous to vehicle units. In addition, we expect adversaries to test new exploits or attacks on devices on the Internet, as automotive units, especially modern ones, are difficult to acquire and deploy on their own.

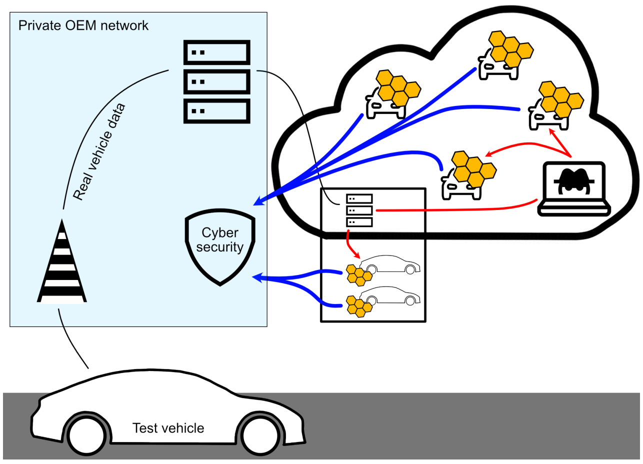


Figure 4. Automotive research honeypot setup with low- and medium-interaction honeypots.

5.2. Vehicle Honeypots on the Internet

As soon as low-interaction honeypots suggest a rising interest in automotive systems, the development of a more in-depth honeypot system is feasible. However, the most difficult part is a credible and accessible placement of full in-vehicle simulations. One way to build a realistic attack surface is a web portal that hides automotive honeypots behind them: automated parking services, remote telemetry services, remote control services, and test vehicle portals can be credible hurdles to deceive adversaries. White-hat researchers have exploited a number of such portals and have thus shown that this is a credible path to automotive devices [21].

As a result, we propose a twofold cloud deployment strategy (illustrated in Figure 4). First, widespread deployments of low-interaction honeypots on the public Internet monitor general attacker interest. Second, sophisticated honeypots simulating entire in-vehicle networks are placed more sparingly behind company or service portals. Potential real vehicle data that is necessary to enhance the honeypot's vehicle simulation is obtained through test vehicles of OEMs and suppliers. While advances in machine learning make full simulation feasible, the technology of digital twins is another promising route. Since remote diagnostics and data analysis in digital twins require real vehicle data, the same data flow can be used to create a realistic honeypot environment—*a digital twin honeypot*.

6. Conclusion

Although honeypots are an established tool for threat intelligence, the automotive industry has not been able to capitalize on this in the past; functional and practical limitations have hindered the widespread deployment of honeypots that mimic in-vehicle components, both on the Internet and within vehicles. However, the necessity of additional computational resources and software-defined approaches proves to be an opportunity for honeypot deployments.

Emerging E/E-architectures lift most of the limitations that the automotive realm currently imposes on the use of honeypots. Additional computing resources, dynamic network approaches, and pervasive connectivity support the deployment of honeypots and the use of their findings. Nevertheless, the honeypot approach can be simplified. Low-interaction honeypots can solve fundamental limitations in many areas, and even simplify deployment through automation options. As an increasing attacker interest in automotive units becomes apparent, however, properly deployed honeypots with higher interaction still have a legitimate place in the threat intelligence strategy.

References

[1] O. Burkacky *et al.*, "Cybersecurity in automotive: Mastering the challenge," Tech. Rep., 2020.

[2] C. Miller *et al.*, "Remote Exploitation of an Unaltered Passenger Vehicle," DARPA, Tech. Rep., 2015.

[3] I. Foster *et al.*, "Fast and Vulnerable: A Story of Telematic Failures," in *9th USENIX WOOT*, 2015.

[4] M. Yan *et al.*, "Security Research on Mercedes-Benz: From Hardware to Car Control," 360 Group, Tech. Rep., 2020.

[5] M. Wolf *et al.*, "WANNADRIVE? Feasible attack paths and effective protection against ransomware in modern vehicles," in *escar Europe*, 2017.

[6] V. Verendel *et al.*, "An Approach to using Honeypots in In-Vehicle Networks," in *IEEE Semiannual Vehicular Technology Conference (VTC)*, 2008, pp. 1–5.

[7] P. Kleberger *et al.*, "Security Aspects of the In-Vehicle Network in the Connected Car," in *IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 528–533.

[8] D. Gantsou *et al.*, "Toward a Honeypot Solution for Proactive Security in Vehicular Ad Hoc Networks," in *Future Information Technology*. Springer Berlin Heidelberg, 2014, pp. 145–150.

[9] Y. Maria Schmitz Née Nestler, "A Strategy for Vehicular Honeypots," 2019.

[10] S. Panda *et al.*, "HoneyCar: A Framework to Configure Honeypot Vulnerabilities on the Internet of Vehicles," 2021.

[11] E. Eriksson *et al.*, "Investigating the Use of Honeypots in Vehicles," Master's thesis, Chalmers University of Technology and University of Gothenburg, 2022.

[12] A. Diarra, "OSI Layers in Automotive Networks," *IEEE 802.1 Plenary Meeting - Orlando*, 2013.

[13] A. Lock *et al.*, "Entering New Worlds: New E/E Architectures With Vehicle Computers Offer New Opportunities," Tech. Rep., 2020.

[14] D. Pannell *et al.*, "Use Cases-IEEE P802. 1DG V0. 4," Jul. 2019.

[15] C. Sanders, *Intrusion Detection Honeypots: Detection Through Deception*. Applied Network Defense, 2020.

[16] N. Ilg *et al.*, "A survey of contemporary open-source honeypots, frameworks, and tools," *Journal of Network and Computer Applications*, vol. 220, p. 103737, 2023.

[17] Upstream Security, "2023 Global Automotive Cybersecurity Report," Last Accessed: 2023-10-06. [Online]. Available: https://upstream.auto/reports/

[18] AUTOSAR GbR, "Adaptive Platform," Last Accessed: 2024-02-07. [Online]. Available: https://www.autosar.org/standards/adaptive-platform

[19] M. Haeberle *et al.*, "Softwarization of Automotive E/E Architectures: A Software-Defined Networking Approach," in *IEEE Vehicular Networking Conference (VNC)*, 2020, pp. 1–8.

[20] K. Tindell, "Canis Automotive Labs CTO Blog - CAN Injection: keyless car theft," Tech. Rep., 2023, Last Accessed: 2024-02-21. [Online]. Available: https://kentindell.github.io/2023/04/03/can-injection/

[21] S. Curry, "Web Hackers vs. The Auto Industry: Critical Vulnerabilities in Ferrari, BMW, Rolls Royce, Porsche, and More," Tech. Rep., 2023, Last Accessed: 2024-01-12. [Online]. Available: https://samcurry.net/web-hackers-vs-the-auto-industry/