The published version is available here: 10.1109/MOCAST61810.2024.10615785

SiHoneypot: a Digital Twin-based honeypot for Autonomous Vehicles

Athanasios Liatifis*, Charis Eleftheriadis†, Zisis Mpatzos†,
Ioannis Nanos†, Thomas Lagkas‡, Sotirios Goudos§, Vasileios Argyriou**,
Konstantinos E. Psannis¶, Ioannis D. Moscholios‖ Panagiotis Sarigiannidis*,

***Abstract*—Autonomous Vehicles (AVs) stand as the vanguard of the automotive industry's evolution, offering a multitude of advantages in terms of transportation efficiency and applications of critical importance. Notably, their interconnection with various smart devices, such as smartphones and associated services, is achieved effortlessly. However, these merits are counterbalanced by significant security risks pertaining to human safety and the potential exposure of personal data. This work introduces SiHoneypot, an innovative honeypot system rigorously crafted to address security challenges intrinsic to AVs. SiHoneypot leverages Digital Twins and incorporates state-of-the-art trends in software deployment, providing a faithful emulation of Autonomous Vehicle systems. Demonstrating its efficacy as a strategic decoy, SiHoneypot affords sufficient time for other security systems to enact responsive measures. Experimental results underscore the minimal resources required for the deployment of SiHoneypot, emphasizing its operational efficiency and resource optimization. Moreover, the inherent extensibility and versatility of SiHoneypot's architecture are showcased, illustrating its adaptability to evolving security challenges within the dynamic landscape of autonomous vehicular technologies.**

***Index Terms*—Honeypots, Autonomous Vehicles, Digital Twin, LiDAR Honeypot**

I. Introduction

Internet of Autonomous Vehicles (IoAV) is the next evolution in the transportation domain, ensuring advanced and dynamic traffic management of vehicles with little-to-none human intervention, eliminating road accidents and improving the overall experience of vehicle drivers and passengers [1]. To achieve the aforementioned goals, Autonomous Vehicles (AVs) leverage several technological advancements such as hardware acceleration, Machine Learning (ML) and 5G technology to make rapid and accurate decisions in constantly evolving environments such as highways and towns with little communication delay [2]. Despite the evident benefits of AVs to society, these systems are prone to a multitude of attacks stemming from the underlying technologies they rely upon, but also to new threats specific to their operational characteristics [3]. Such threats include attacks on the navigation and control systems, attacks on the sensors of the vehicle and attacks related to the protocols and subjacent communication technologies [4].

Honeypots are software components that emulate the behaviour of real assets or services, with the goal to present themselves as vulnerable entities to attract attackers [5]. Depending on the level of interaction with another entity (usually an attacker), a honeypot may be categorised as low-interaction, medium-interaction or high-interaction. As the interaction level increases, so does the complexity in terms of development effort and the emulation capabilities of the honeypot with respect to the device or service.

Low-interaction honeypots can emulate simple protocols and services, whereas high-interaction ones can emulate whole systems making them indistinguishable. Evidently, a tradeoff exists in the degree of interaction between a honeypot and the host system, thereby impacting the potential data acquired from malicious actors. A higher interaction level allows for more extensive engagement, resulting in a larger volume of collected data. Conversely, a lower interaction level constrains engagement, diminishing critical interactions and limiting the amount of collected data. Throughout the interaction process, valuable logs are recorded together with traffic traces. The data can then be analysed to get valuable insight related to attacker motives and possible misconfiguration of security tools.

In this work SiHoneypot framework is presented, a novel high-interaction honeypot that leverages the latest trends in microservices and Digital Twin technology to emulate AV sensors. To the best of our knowledge this is the first work to present an application of a LiDAR sensor Honeypot. The structure of the paper is as follows: Section I introduces honeypots as a possible mitigation method against security concerns regarding AVs. A short survey review of related honeypot developments in past years is presented in Section II. Section III presents the SiHoneypot framework, a Digital Twin high interaction Honeypot that follows state-of-the-art practices. Section IV presents evaluation results and performance metrics of the SiHoneypot. Finally, Section V concludes our findings, while also outlines potential future research endeavors.

*A. Liatifis and P. Sarigiannidis are with the Department of Electrical and Computer Engineering, University of Western Macedonia, Kozani, Greece - E-mail: {aliatifis, psarigiannidis}@uowm.gr

†C. Eleftheriadis, Z. Mpatzos and I. Nanos are with Sidroco Holdings Ltd, Nicosia, Cyprus - E-mail: {celeftheriadis, zmpatzos, inanos}@sidroco.com

‡T. Lagkas is with the Department of Computer Science, International Hellenic University, Kavala, Greece - E-mail: tlagkas@cs.ihu.gr

§S. Goudos is with the Department of Physics, Aristotle University of Thessaloniki, Thessaloniki, Greece - E-mail: sgoudo@physics.auth.gr

**V. Argyriou is with the Department of Networks and Digital Media, Kingston University London, Penrhyn Road, Kingston upon Thames, Surrey KT1 2EE, UK - E-Mail: vasileios.argyriou@kingston.ac.uk

¶K. E. Psannis is with the Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece - E-mail: kpsannis@uom.edu.gr

‖I. D. Moscholios is with the Department of Informatics and Telecommunications, University of Peloponnese, Tripolis, Greece - E-mail: idm@uop.gr

II. Related Work

Honeypots differ fundamentally from conventional security solutions, such as firewalls, as their primary purpose is not to obstruct attackers. Instead, they purposefully present themselves as vulnerable targets, diverting the focus of potential threats and introducing delays sufficient for other security systems to promptly detect and forestall any malicious activities. This proactive approach ensures the timely protection of authentic assets. Given their inherent attributes, honeypots find applicability across diverse domains, including but not limited to IoT [6] and web-servers [7].

Siniosoglou et al. developed NeuralPot [8], a high-interaction Industrial Control Systems honeypot that leverages Deep Neural Networks (DNNs) to emulate the behaviour of Modbus devices. The honeypot uses a collection of past measurements to train the model and emulate Modbus device readings. Suratkar et al. [9] also leverage DNNs to develop a Reinforcement Learning honeypot that can perform severity analysis on adversaries and hide its presence. The proposed system was evaluated using Cowrie [1] honeypot.

HoneyBot [10] is a hybrid robotic system honeypot that incorporates software and hardware components to emulate a robotic system. HoneyPhy honeypot [11] includes real sensors that are queried in real time upon request. Malicious commands are analysed and executed in a safe emulated environment to protect the real devices. HoneyBoT is the first honeypot that attempts to combine the benefits of real systems with the safety of emulated environments.

HoneyCar [12] is a framework designed specifically for honeypot deception in Internet of Vehicle environments. HoneyCar offers dynamic honeypot configuration of honeypot instances by modelling the honeypots and the attacker's interactions as an imperfect zero-sum game theoretic model. The framework can efficiently generate optimal strategies Common Vulnerability Scoring System (CVSS) [13] values of the Common Vulnerabilities and Exposure (CVE) [2] list. However, Honeycar does not propose any new AV-specific honeypot.

Anastasiadis et al. [14] propose a high-interaction honeypot framework that emulates IoV sensors and leverages Markov Chain Models to analyse honeypot logs to describe attack propagation patterns.

III. Proposed System Architecture

At this juncture, the proposed architecture of SiHoneypot introduced, exhibiting the details of its integral components and their respective functionalities. The tool comprises three primary components: 1) The Honeypot component, tasked with emulating the behavior of authentic sensors and/or embedded devices within complex cyber-physical systems. In our case, the Honeypot implementation focuses on replicating the operations of a LiDAR sensor integrated into an AV. 2) The Digital Twin which mirrors the functionalities of the Honeypot, accurately replicating its entire operational cycle. 3) The Dashboard which serves as a user-friendly interface equipped with visualizations, aiding interested individuals in making informed decisions.

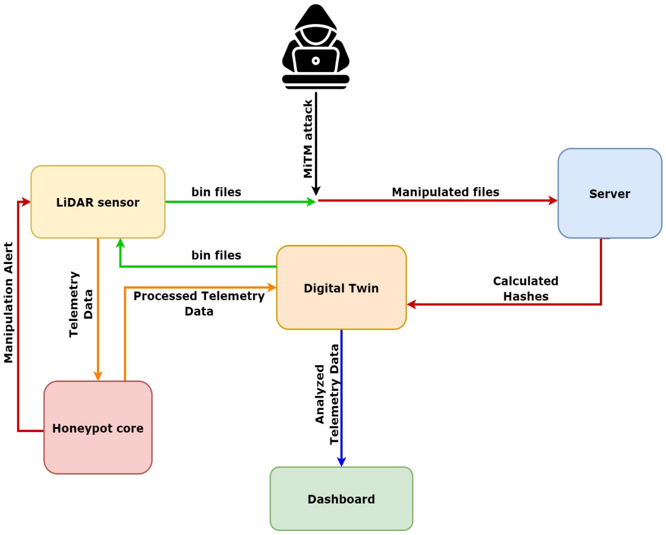


Fig. 1: SiHoneypot architecture

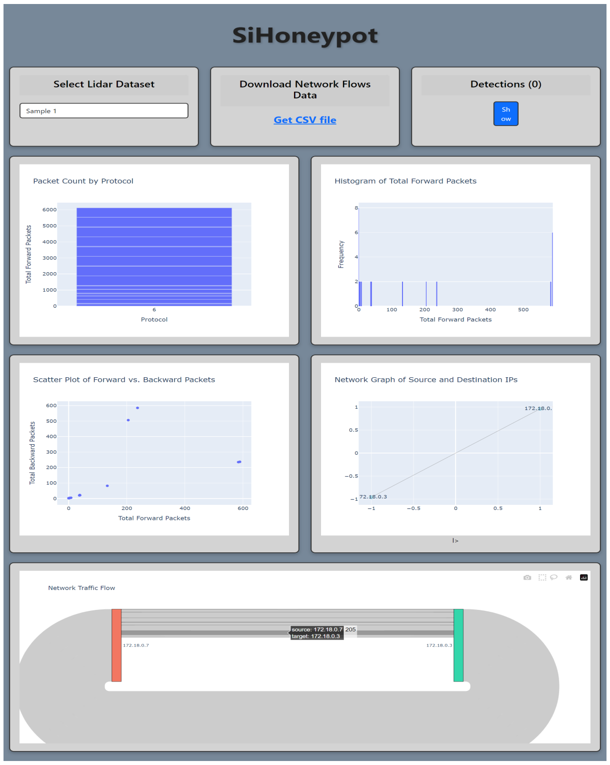
Figure 1 outlines a detailed depiction of SiHoneypot's architecture. The LiDAR Honeypot simulates a LiDAR sensor, generating binary (.bin) files with unique hashes. These files are transmitted to a designated location, posing a potential cybersecurity risk, such as a Man-in-the-Middle attack. Upon arrival, the .bin files undergo Merkle tree hashing [15], and any inconsistency triggers a manipulation alert sent to the LiDAR Honeypot. In response, the Honeypot generates a Telemetry data JSON file, converted to STIX 2.1 [3] format and sent to the Dashboard. The Dashboard transforms the analyzed data into insightful plots for decision-making. Detailed descriptions of each component are provided in subsequent subsections.

A. LiDAR Honeypot

LiDAR Honeypot serves as a protective measure against cyber threats targeting LiDAR Sensors crucial for modern AI systems and AVs. The proposed honeypot component consists of two main facets: the emulated sensor and the core part that is responsible for all the related operations and message exchanges. This digital decoy accurately emulates a real LiDAR Sensor, streaming LiDAR-generated point clouds as .bin files and transmitting telemetry data in JSON format to the Digital Twin. Leveraging Apache Kafka [4] for scalability, fault tolerance, and high throughput, the Honeypot ensures robust data pipelines.

The Honeypot core serves as the central orchestrator for all LiDAR data operations, embodying principles of modularity and extendibility in its design patterns. This fact ensures the seamless adaptation of its operations and functionalities, enabling the future simulation of a wide range of devices and

[1] https://github.com/cowrie/cowrie/
[2] https://www.cve.org/
[3] https://oasis-open.github.io/cti-documentation/stix/intro
[4] https://kafka.apache.org



Sample 1

Fig. 2: Graphical Interface of SiHoneypot

sensors. This adaptability is crucial for accommodating evolving cybersecurity landscapes and technological advancements.

The Honeypot core is responsible for the alert exchange with the emulated LiDAR sensor upon detecting malicious activities, transforming telemetry data for anomaly detection using CICFlowmeter-V4.0 [5] format. To convey attack details, Honeypot core prepares informative messages in compliance with the STIX 2.1 cyber threat intelligence standard.

B. Digital Twin

A Digital Twin [16] is a virtual representation of a physical entity, utilized for simulation, integration, testing, monitoring, and maintenance. In the proposed system, the Digital Twin acts as a moderator between the LiDAR Honeypot and the Dashboard. It pushes .bin files to the Honeypot, which assimilates them through Kafka streaming and assigns unique hashes to them facilitating future verification steps. The Digital Twin can serve data from various sources such as well-known atasetss [17], organisational data or anonymised data incorporating LiDAR measurements, represented as point clouds, from various landscapes.

The verification procedure unfolds as follows: upon receiving the .bin files transmitted by the LiDAR Honeypot, the Server utilizes these files as input for a hashing function, specifically utilizing the Merkle tree signature (MTS) algorithm, yielding a corresponding hash. The Server assumes the role of a database management system, facilitating communication between the SiHoneypot and the network. Additionally, it undertakes crucial operations including hash calculation, in order to optimize the verification process, thereby enhancing the overall dependability of the system architecture. Consequently, these hashes are transmitted back to the Digital Twin for comparison against the pre-existing list of known hashes. Upon a non-matching scenario, a flag is raised, triggering the transmission of a message to the LiDAR Honeypot to commence the generation of Telemetry data, encompassing detailed information about the deployed attack.

As a digital twin, it faithfully mirrors Honeypot operations and functionalities, ensuring synchronized and congruent behaviour. Distinctively, the Digital Twin interacts with the Dashboard pushing metrics and results related to data integrity. In this interaction, the Digital Twin communicates several file types to the Dashboard, encompassing: 1) .bin files linked to the LiDAR Honeypot output, and 2) thoroughly analyzed Telemetry data packaged in CSV files, offering a comprehensive overview of malicious network flows.

C. Dashboard

The architecture's final component is a dynamic dashboard showcasing SiHoneypot outcomes and emulating LiDAR sensor data, as illustrated in Figure 2. The Digital Twin connects these elements and launches the Dashboard online. Given the containerised environment of SiHoneypot, the dashboard offers a simple and efficient interface to communicate with container instances.

The Dashboard has the capacity to accommodate various output file types from different devices/sensors, offering a comprehensive view of the network's status through diverse visualization techniques like charts, histograms, and other methods. Key functionalities include selecting LiDAR data for which the attack report has been generated, downloading reports in CSV format, and displaying real-time detections of manipulated data. Histograms depict network packet details, while scatter plots and spider diagrams allow in-depth analysis. In summary, the Dashboard's visualizations provide a clear representation of network traffic, improving usability and comprehension for end-users.

IV. Evaluation and Results

To evaluate SiHoneypot, a high-end computer with an i7-12700K, 32GB of RAM and 1 TB of solid-state drive was used. Table I summarises our findings regarding resource consumption when idle and under load (i.e. when the honeypot is attacked). The tests were conducted to measure the overall resource consumption of SiHneypot. To attack the honeypot, *hping3* tool was used to perform denial of service attacks, a set of attacks well-known for their resource consumption on the victim's side. The attacks included packets with payload and random ports to stress the LiDAR Sensor. The test was designed to measure the worst-case scenario of resource consumption for each component, especially on the LiDAR Sensor responsible for interaction with any potential attacker.

[5] https://github.com/ahlashkari/CICFlowMeter

All components consume little resources when in an idle state and under load with the LiDAR sensor being the only exception. The percentage values measure the utilization of all system cores (i.e. 100% translates to all system cores utilised).

TABLE I: SiHoneypot idle and under load scnario

		Idle	Under Load
CPU	LiDAR Sensor	0.6%	7.7%
	Digital Twin	0.5%	2.6%
	Honeypot Core	0.5%	4%
Memory	LiDAR Sensor	400MB	1.2GB
	Digital Twin	400MB	450MB
	Honeypot Core	300MB	950MB
Network bandwidth	LiDAR Sensor	2Mpbs	3Mbps

Based on the analysis above Table II summarises the recommended system requirements for all components to properly function. In terms of necessary bandwidth, the LiDAR Sensor uses 2 Mbps to send the security incidents in STIX 2.0 format. As a result, SiHoneypot can operate on a wide range of infrastructure sites, ranging from a couple of computers to large-scale computing nodes. The microservices approach of SiHoneypot simplifies the deployment of its components with little effort for infrastructure operators or administrators.

TABLE II: System Requirements for SiHoneypot and Digital Twin Core

	LiDAR Sensor	Digital Twin	Honeypot Core
CPU	2 Cores	1 Core	2 Cores
RAM	2 GBs	1 GB	1 GB

V. Conclusion

Honeypots are a promising means of gathering and analyzing data pertaining to attackers, yielding valuable insights into their motives and uncovering potential security misconfigurations. A lack of IoV-specific honeypots has led to the development of SiHoneypot, a high-interaction honeypot that leverages Digital Twin technology to emulate AV sensors. The first version of Sihoneypot set the foundation for a highly configurable and extensible honeypot system. In the future, we plan to include additional sensors and AV components, increasing support for diverse AV profiles. Additionally, we aim to follow recent trends in honeypot development by incorporating AI into the process of system behaviour emulation. In conclusion, prioritizing the establishment of metrics to assess the performance and emulation accuracy of honeypots used in AVs and related domains is imperative, given the limited progress in this research area.

Acknowledgment

This work is a part of the IRIS project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101021727. This content reflects only the author's view, and the European Commission is not responsible for any use that may be made of the information this publication contains.

References

[1] O. Kaiwartya, A. H. Abdullah, Y. Cao, A. Altameem, M. Prasad, C.-T. Lin, and X. Liu, "Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects," *IEEE Access*, vol. 4, pp. 5356–5373, 2016.

[2] M. N. Ahangar, Q. Z. Ahmed, F. A. Khan, and M. Hafeez, "A survey of autonomous vehicles: Enabling communication technologies and challenges," *Sensors*, vol. 21, no. 3, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/3/706

[3] K. Kim, J. S. Kim, S. Jeong, J.-H. Park, and H. K. Kim, "Cybersecurity for autonomous vehicles: Review of attacks and defense," *Computers & Security*, vol. 103, p. 102150, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404820304235

[4] A. Nanda, D. Puthal, J. J. P. C. Rodrigues, and S. A. Kozlov, "Internet of autonomous vehicles communications security: Overview, issues, and directions," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 60–65, 2019.

[5] J. Franco, A. Aris, B. Canberk, and A. S. Uluagac, "A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2351–2383, 2021.

[6] D. Pliatsios, P. Sarigiannidis, T. Liatifis, K. Rompolos, and I. Siniosoglou, "A novel and interactive industrial control system honeypot for critical smart grid infrastructure," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2019, pp. 1–6.

[7] D. K. Rahmatullah, S. M. Nasution, and F. Azmi, "Implementation of low interaction web server honeypot using cubieboard," in *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, 2016, pp. 127–131.

[8] I. Siniosoglou, G. Efstathopoulos, D. Pliatsios, I. D. Moscholios, A. Sarigiannidis, G. Sakellari, G. Loukas, and P. Sarigiannidis, "Neuralpot: An industrial honeypot implementation based on deep neural networks," in *2020 IEEE Symposium on Computers and Communications (ISCC)*, 2020, pp. 1–7.

[9] S. Suratkar, K. Shah, A. Sood, A. Loya, D. Bisure, U. Patil, and F. Kazi, "An adaptive honeypot using q-learning with severity analyzer," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 10, p. 4865–4876, Oct. 2022. [Online]. Available: https://link.springer.com/10.1007/s12652-021-03229-2

[10] C. Irvene, D. Formby, S. Litchfield, and R. Beyah, "Honeybot: A honeypot for robotic systems," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 61–70, 2018.

[11] S. Litchfield, D. Formby, J. Rogers, S. Meliopoulos, and R. Beyah, "Rethinking the honeypot for cyber-physical systems," *IEEE Internet Computing*, vol. 20, no. 5, pp. 9–17, 2016.

[12] S. Panda, S. Rass, S. Moschoyiannis, K. Liang, G. Loukas, and E. Panaousis, "Honeycar: A framework to configure honeypot vulnerabilities on the internet of vehicles," *IEEE Access*, vol. 10, pp. 104 671–104 685, 2022.

[13] P. Mell, K. Scarfone, and S. Romanosky, "Common vulnerability scoring system," *IEEE Security & Privacy*, vol. 4, no. 6, pp. 85–89, 2006.

[14] M. Anastasiadis, K. Moschou, K. Livitckaia, K. Votis, and D. Tzovaras, "A novel high-interaction honeypot network for internet of vehicles," in *2023 31st Mediterranean Conference on Control and Automation (MED)*, 2023, pp. 281–286.

[15] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Advances in Cryptology — CRYPTO '87*, C. Pomerance, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 369–378.

[16] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.

[17] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013. [Online]. Available: https://doi.org/10.1177/0278364913491297