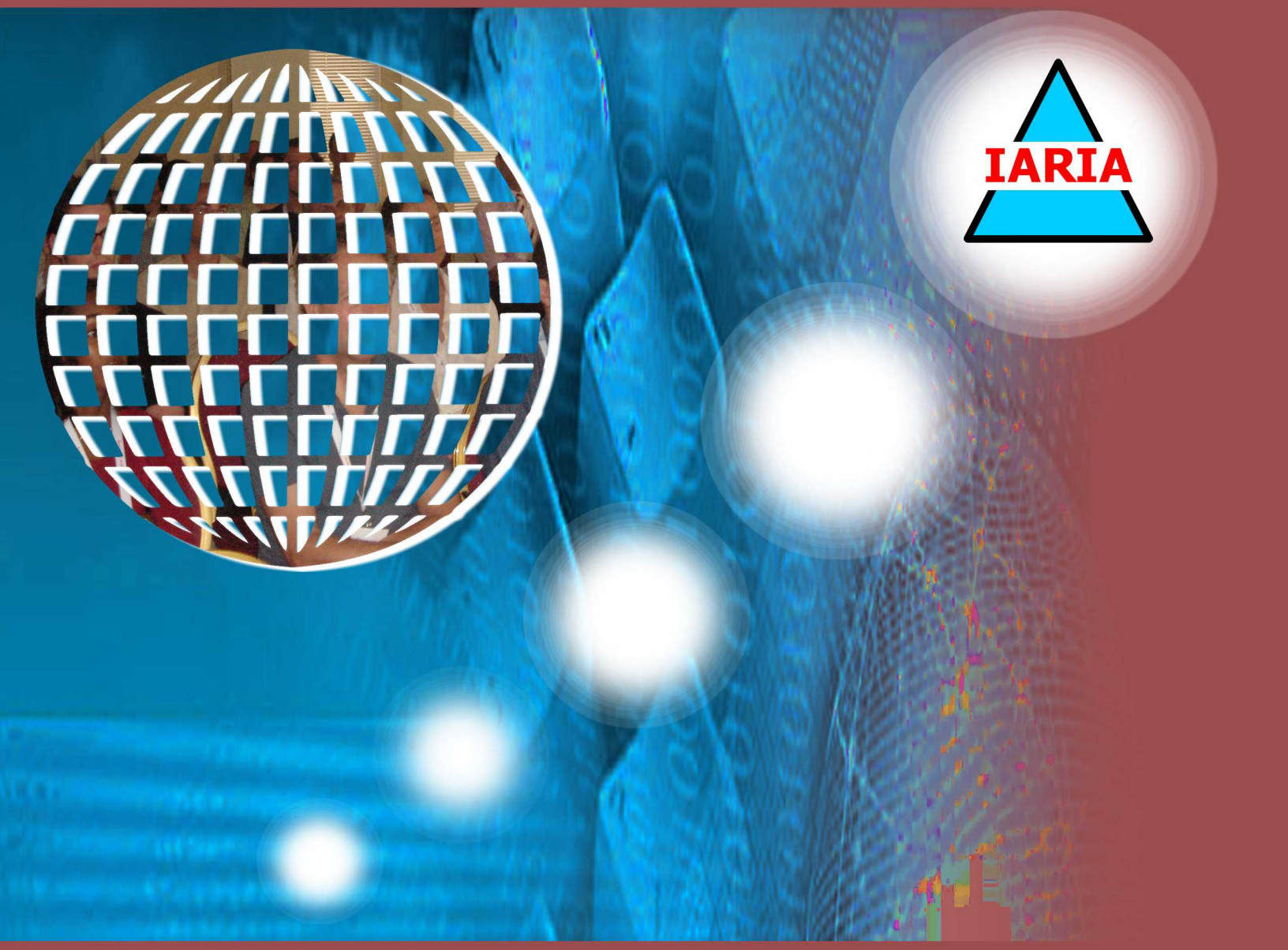
International Journal on Advances in Security



2022 vol. 15 nr. 3&4

The *International Journal on Advances in Security* is published by IARIA.
ISSN: 1942-2636
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Security
Volume 15, Number 3 & 4, 2022

Clelia Colombo Vilarrasa, Autonomous University of Barcelona, Spain
Peter Cruickshank, Edinburgh Napier University Edinburgh, UK
Nora Cuppens, Institut Telecom / Telecom Bretagne, France
Glenn S. Dardick, Longwood University, USA
Vincenzo De Florio, University of Antwerp & IBBT, Belgium
Paul De Hert, Vrije Universiteit Brussels (LSTS) - Tilburg University (TILT), Belgium
Pierre de Leusse, AGH-UST, Poland
William Dougherty, Secern Consulting - Charlotte, USA
Raimund K. Ege, Northern Illinois University, USA
Laila El Aimani, Technicolor, Security & Content Protection Labs., Germany
El-Sayed M. El-Alfy, King Fahd University of Petroleum and Minerals, Saudi Arabia
Rainer Falk, Siemens AG - Corporate Technology, Germany
Shao-Ming Fei, Capital Normal University, Beijing, China
Eduardo B. Fernandez, Florida Atlantic University, USA
Anders Fongen, Norwegian Defense Research Establishment, Norway
Somchart Fugkeaw, Thai Digital ID Co., Ltd., Thailand
Steven Furnell, University of Plymouth, UK
Clemente Galdi, Universita' di Napoli "Federico II", Italy
Birgit Gersbeck-Schierholz, Leibniz Universität Hannover, Germany
Manuel Gil Pérez, University of Murcia, Spain
Karl M. Goeschka, Vienna University of Technology, Austria
Stefanos Gritzalis, University of the Aegean, Greece
Michael Grottke, University of Erlangen-Nuremberg, Germany
Ehud Gudes, Ben-Gurion University - Beer-Sheva, Israel
Indira R. Guzman, Trident University International, USA
Huong Ha, University of Newcastle, Singapore
Petr Hanáček, Brno University of Technology, Czech Republic
Gerhard Hancke, Royal Holloway / University of London, UK
Sami Harari, Institut des Sciences de l'Ingénieur de Toulon et du Var / Université du Sud Toulon Var, France
Daniel Harkins , Hewlett Packard Enterprise, USA
Ragib Hasan, University of Alabama at Birmingham, USA
Masahito Hayashi, Nagoya University, Japan
Michael Hobbs, Deakin University, Australia
Hans-Joachim Hof, INSicherheit - Ingolstadt Research Group Applied IT Security, CARISSMA – Center of Automotive Research on Integrated Safety Systems, Germany
Neminath Hubballi, Infosys Labs Bangalore, India
Mariusz Jakubowski, Microsoft Research, USA
Ravi Jhawar, Università degli Studi di Milano, Italy
Dan Jiang, Philips Research Asia Shanghai, China
Georgios Kambourakis, University of the Aegean, Greece
Florian Kammueller, Middlesex University - London, UK
Sokratis K. Katsikas, University of Piraeus, Greece
Seah Boon Keong, MIMOS Berhad, Malaysia
Sylvia Kierkegaard, IAITL-International Association of IT Lawyers, Denmark
Hyunsung Kim, Kyungil University, Korea
Geir M. Køien, University of Agder, Norway
Ah-Lian Kor, Leeds Metropolitan University, UK
Evangelos Kranakis, Carleton University - Ottawa, Canada
Lam-for Kwok, City University of Hong Kong, Hong Kong
Jean-Francois Lalande, ENSI de Bourges, France
Gyungho Lee, Korea University, South Korea
Clement Leung, Hong Kong Baptist University, Kowloon, Hong Kong
Diego Liberati, Italian National Research Council, Italy

Giovanni Livraga, Università degli Studi di Milano, Italy
Gui Lu Long, Tsinghua University, China
Jia-Ning Luo, Ming Chuan University, Taiwan
Thomas Margoni, University of Western Ontario, Canada
Rivalino Matias Jr ., Federal University of Uberlandia, Brazil
Manuel Mazzara, UNU-IIST, Macau / Newcastle University, UK
Catherine Meadows, Naval Research Laboratory - Washington DC, USA
Carla Merkle Westphall, Federal University of Santa Catarina (UFSC), Brazil
Ajaz H. Mir, National Institute of Technology, Srinagar, India
Jose Manuel Moya, Technical University of Madrid, Spain
Leonardo Mostarda, Middlesex University, UK
Jogesh K. Muppala, The Hong Kong University of Science and Technology, Hong Kong
Syed Naqvi, CETIC (Centre d'Excellence en Technologies de l'Information et de la Communication),Belgium
Sarmistha Neogy, Jadavpur University, India
Mats Neovius, Åbo Akademi University, Finland
Jason R.C. Nurse, University of Oxford, UK
Peter Parycek, Donau-Universität Krems, Austria
Konstantinos Patsakis, Rovira i Virgili University, Spain
João Paulo Barraca, University of Aveiro, Portugal
Sergio Pozo Hidalgo, University of Seville, Spain
Yong Man Ro, KAIST (Korea advanced Institute of Science and Technology), Korea
Rodrigo Roman Castro, University of Malaga, Spain
Heiko Roßnagel, Fraunhofer Institute for Industrial Engineering IAO, Germany
Claus-Peter Rückemann, Leibniz Universität Hannover / Westfälische Wilhelms-Universität Münster / North-German Supercomputing Alliance, Germany
Antonio Ruiz Martinez, University of Murcia, Spain
Paul Sant, University of Bedfordshire, UK
Peter Schartner, University of Klagenfurt, Austria
Alireza Shameli Sendi, Ecole Polytechnique de Montreal, Canada
Dimitrios Serpanos, Univ. of Patras and ISI/RC ATHENA, Greece
Pedro Sousa, University of Minho, Portugal
George Spanoudakis, City University London, UK
Vladimir Stantchev, Institute of Information Systems, SRH University Berlin, Germany
Lars Strand, Nofas, Norway
Young-Joo Suh, Pohang University of Science and Technology (POSTECH), Korea
Jani Suomalainen, VTT Technical Research Centre of Finland, Finland
Enrico Thomae, Ruhr-University Bochum, Germany
Tony Thomas, Indian Institute of Information Technology and Management - Kerala, India
Panagiotis Trimintzios, ENISA, EU
Peter Tröger, Hasso Plattner Institute, University of Potsdam, Germany
Simon Tsang, Applied Communication Sciences, USA
Marco Vallini, Politecnico di Torino, Italy
Bruno Vavala, Carnegie Mellon University, USA
Mthulisi Velempini, North-West University, South Africa
Miroslav Velev, Aries Design Automation, USA
Salvador E. Venegas-Andraca, Tecnológico de Monterrey / Texia, SA de CV, Mexico
Szu-Chi Wang, National Cheng Kung University, Tainan City, Taiwan R.O.C.
Steffen Wendzel, Fraunhofer FKIE, Bonn, Germany
Piyi Yang, University of Shanghai for Science and Technology, P. R. China
Rong Yang, Western Kentucky University , USA
Hee Yong Youn, Sungkyunkwan University, Korea
Bruno Bogaz Zarpelao, State University of Londrina (UEL), Brazil
Wenbing Zhao, Cleveland State University, USA

International Journal on Advances in Security
Volume 15, Numbers 3 & 4, 2022

CONTENTS

Attack Surface Reduction to Minimize Private Data Loss from Breaches

George O. M. Yee
Computer Research Lab, Aptusinnova Inc., Ottawa, Canada
Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada
e-mail: george@aptusinnova.com, gmyee@sce.carleton.ca

Abstract— **Organizations are increasingly being victimized by breaches of private data, resulting in heavy losses to both the organizations and the owners of the data. For organizations, these losses include large expenses to resume normal operation and damages to its reputation. For data owners, the losses may include financial loss and identity theft. To defend themselves from such data breaches, organizations install security controls (e.g., encryption) to secure their vulnerabilities. While such controls help, they are far from being fool proof. Reducing the attack surface is a sound core approach for protecting valuable data. This paper applies this reduction to minimize the data loss from e-commerce data breaches. The paper first examines the behaviour of Business-to-Consumer (B2C) e-commerce companies in terms of why they collect and store personal data. It then applies attack surface reduction by limiting the amount of private data that the company stores in its computer system, while preserving the company's ability to accomplish its purposes for collecting the private data. The paper illustrates the approach by applying it to different types of B2C e-commerce companies.**

Keywords-attack surface reduction; minimizing data loss; data breach; private data loss; B2C e-commerce.

I. INTRODUCTION

This work extends Yee [1] by a) relating the approach to attack surface reduction, b) improving the explanations throughout the paper, as well as updating the examples of breaches in Section I, c) showing mathematically how the approach reduces the risk of data loss, d) adding application examples, and e) increasing the number of references.

Data breaches of personal data or personal information are appearing more and more often in the news, devastating the victim organizations. The losses have serious negative consequences both to the consumer (e.g., financial loss, identity theft) and to the organization (e.g., loss of reputation, loss of trust). Breaches of private data held by companies and other types of organizations have been occurring at an alarming rate. Each year has been accompanied by its assortment of data breaches. Consider the following sampling of breaches in 2022 [2], the year of this work:

- August, 2022: Up to 20 Million Plex Users Compromised. Plex offers streaming services for movies, music, and games, and hosts user-produced audio and visual content. Plex informed its customers on August 24 that it suffered a data breach impacting most of its user accounts. The private data loss included usernames, email addresses, and passwords of approximately 20 million users.
- July, 2022: 69 Million Accounts Exposed in Neopets Breach. Neopets is a virtual pet website where users can own virtual pets and buy virtual items for them. On July 19, 2022, a hacker posted data on 69 million Neopets users for sale on an online forum. The private data loss included name, email address, date of birth, zip code, and more, as well as 460 MB of compressed source code for the Neopets website.
- June, 2022: Up to 2 Million People Compromised in Shields Health Care Group Breach. The Massachusetts-based Shields Health Care Group disclosed in June, 2022, that they had detected a breach in March, 2022. The loss of private data included names, social security numbers, medical records, and other sensitive personal information.

In response to these attacks, organizations attempt to identify the vulnerabilities in their computer systems and secure these vulnerabilities using security controls. Example security controls are firewalls, intrusion detection systems, encryption, two-factor authentication, and social engineering awareness training for employees. Unfortunately, securing vulnerabilities with security controls is far from being foolproof. One major weakness is that it is impossible to find all the vulnerabilities in a computer system. This means that it is highly likely that a determined attacker will find an attack path into the organization's system that has been overlooked and cause a data breach, even though the organization believes that it has done due diligence and secured all its vulnerabilities. Nevertheless, security controls do help to prevent breaches, and we are not advocating that they be eliminated. Rather, the approach in this work can be considered as an addition to the existing arsenal of security controls.

In this work, we propose an approach in which most of the private data collected by an organization is stored on the user's device. Thus, a smaller quantity of private data remains on the company's computer system, reducing the system's attack surface and minimizing the loss of private data should the company-stored data ever be breached. The approach also ensures that the needs of the company to carry out its

purposes for collecting the private data are satisfied. The user's device could be a desktop computer, a laptop, or a smart phone. The approach is intended for Business-to-Consumer (B2C) e-commerce companies, since B2C companies appear to collect large quantities of personal data and are often victimized by data breaches. Note that in this work when we write about data storage on or in the "company's computer system", we mean that the data is stored on company premises or in the cloud.

This paper is organized as follows. Section II looks at private data, attacks, and attack surface. Section III examines the behaviour of B2C companies in terms of why they collect and store personal information. It also looks at the nature of the collected information. Section IV presents the approach, including a mathematical description of how it reduces the risk of data loss. Section V gives examples of how the approach can fit with different types of e-commerce companies. Section VI describes related work. Section VII gives conclusions and future work.

II. Private Data, Attacks, and Attack Surface

This section explains private data, attacks, and attack surface.

A. Private Data, Attacks, and Attack Surface

Private data consists of information about a person that can identify or be linked to that person and is owned by that person [3]. Thus, private data is also "personal information", and consists of "personal data". For example, a person's height, weight, or credit card number can all be used to identify the person and are considered as personal information. There are other types of personal information, such as buying patterns and navigation habits (e.g., websites visited) [4]. An individual's privacy refers to his/her ability to control the collection (what private data and collected by which party), purpose of collection, retention, and disclosure of that data, as stated in the individual's privacy preferences [3]. In many countries, private data is protected by legislation in which the concept of "purpose" for collecting the personal information (how the collected information will be used) is important. Companies must disclose the purpose for collecting the personal information and cannot use the information for any other purpose. Private data needs protection and must not fall into the wrong hands.

DEFINITION 1: An *attack* is any action carried out against an organization's computer system that, if successful, results in the system being compromised.

This work focuses on attacks that compromise the private data (PD) held in the online systems of organizations. The attacker who launches an attack may be internal (inside attacker) or external (outside attacker) to the organization. An internal attacker usually has easier access to the targets of his/her attack and he/she may hide his/her attacks in the guise of normal duty. This work focuses on outside attackers. Reference [5] gives a good account of how to mitigate insider attacks.

Salter et al. [6] give an interesting insight into what enables a successful attack: "Any successful attack has three steps: One, diagnose the system to identify some attack. Two, gain the necessary access. And three, execute the attack. To protect a system, only one of these three steps needs to be blocked." Thus, an attack surface must contain a target that the attacker deems worthy of attack (suit his/her purpose for the attack) and that target must be accessible to the attacker. For this work, the target that is potentially worthy of attack is the PD that is accessible to attackers. In a computer system, this PD is either moving (travelling from one location to another), at rest (stored), or being used (by some process). This leads to the following definition of attack surface:

DEFINITION 2: The *attack surface* for private data, also called the *private data attack surface*, contained in an online computer system is the set of all locations in the system that contain attacker accessible PD in the clear, where the PD is moving, at rest, or being processed.

In Definition 2, "attacker accessible PD" means that the attacker is able to exfiltrate the PD using some agent of attack, such as malware against stored PD and PD being processed, or a man-in-the-middle attack against a link containing moving PD. Also, we assume that attackers would attack PD that is in the clear rather than PD that is encrypted. In the rest of this paper, by "attack surface" we mean the private data attack surface, unless otherwise indicated. Figure 1 shows an example private data attack surface.

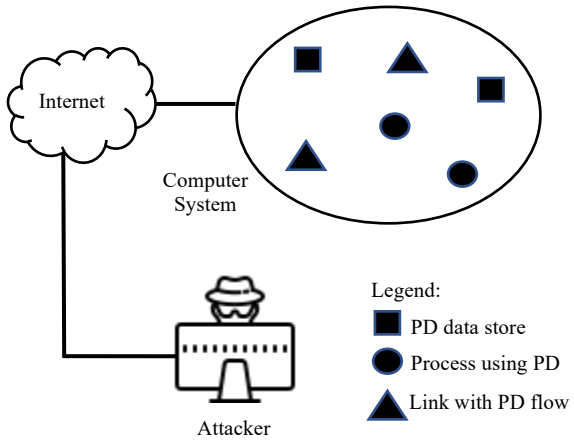


Figure 1. Example private data attack surface consisting of the set of all 6 attacker accessible locations in the system that contain PD in the clear.

An alternative definition of attack surface for PD contained in a computer system is the set of ways the attacker has to exfiltrate the PD. However, given the complexity of computer systems and the fact that the tools available to the attacker to use in his/her attacks are unknown to us, it is next to impossible to determine this set. On the other hand, locations that contain attacker accessible PD are easier to identify. Since an exfiltration must be from a location that contains PD, the set of such exfiltrations depends on the set of such locations. The larger the set of locations, the larger the set of exfiltrations. The smaller the set of locations, the smaller the set of exfiltrations. Therefore, Definition 2 in a

sense includes this alternative definition, but in addition, is more easily applied.

As mentioned above, in the first step of a successful attack, the attacker diagnoses the system to identify the attack [6]. A smaller attack surface will make this step more difficult for the attacker. Therefore, a smaller attack surface corresponds to higher security, which is why we wish to reduce the attack surface. Definition 2 also gives rise to this conclusion: a smaller attack surface means a smaller number of locations that contain PD, which in turn means fewer opportunities for exfiltration of the PD, or in other words, higher security.

Definition 2 is consistent with the intuitive understanding of an attack surface (the usual meaning), which is "the set of ways in which an adversary can enter the system and potentially cause damage" [7]. Each "way" corresponds to a location in Definition 2 that in turn corresponds to methods for exfiltrating PD from the location.

III. The Collection and Storage of Personal Information by B2C Companies

In this section we examine why B2C companies collect personal information and discuss the nature of this information.

A. Purposes for Collecting Personal Information

Companies engaged in B2C e-commerce, collect personal information for the following purposes:

- Transaction Requirements (self-evident): Personal information is needed and used in carrying out the transaction. For example, making an online purchase requires your name and address for goods delivery.
- Communication (self-evident): Personal contact information is needed to communicate with customers for resolving order issues or to answer product questions.
- To Secure Other Data: A personal biometric is needed for further authentication, e.g., a voice print, prior to allowing the customer to access more secure areas of his or her account [8]. The biometric may also be required for use in multi-factor authentication.
- Establishing Loyalty: A personal history of past transactions may be required to establish a customer's loyalty in order to reward the customer with certain benefits such as free shipping or product discounts [9].
- Targeted Advertising: A personal history of past transactions is needed to understand the type of products a particular customer has purchased in the past, and thereby create more appealing and effective ads directed at the customer [10].
- Market Research: The personal histories of past transactions for all customers are studied in order to understand what products appeal to customers in order to make decisions for stock purchases, or to provide a better customer experience in terms of app or website design [8].
- Sharing or Selling: Personal information collected is shared or sold to other organizations for a profit [8].

B. E-Commerce Data

In B2C e-commerce, online companies sell items and services to consumers. Example types of such companies include sellers of goods and services (e.g., Amazon.com), hotels (e.g., Mariott.com), travel agencies (e.g., Expedia.ca), financial services (e.g., CIBC.com), and the list goes on. All these companies share common data types. Each company offers products that customers purchase. Table 1 identifies the products for the e-commerce company types mentioned above.

Each customer has a set of personal identifying information, such as name, postal address, and phone number that identify the customer, and depending on the service provided by the company, include personal information such as credit card details, date of birth, amount of mortgage on house, and so on. We group all such personal identifying information under the heading Customer Personal Data (CPD). Each customer makes one or more product selections and effects payment for the product(s) selected. In addition, there is ancillary data, such as type of payment, date ordered, date shipped, date delivered (from delivery agent, e.g., courier), and so on. Table 2 shows these data types and whether they originate from the company or the customer.

TABLE 1. PRODUCTS ASSOCIATED WITH EACH COMPANY TYPE.

Company type	Products
Sellers of goods and services (e.g., Amazon.com)	Physical items such as pots, clothing, and electronics; services such as selling your items for you
Hotels (e.g., Mariott.com)	Rooms
Travel Agencies (e.g., Expedia.ca)	Travel bookings
Financial services (e.g., CIBC.com)	Fee-based banking accounts

TABLE 2. DATA TYPES AND WHERE THEY ORIGINATE.

Data type	Origin
Products	Company
CPD	Customer
Product selection	Customer
Amount paid	Company
Ancillary data	Company

We can see that each online customer order involves the data types shown in the left column of Table 2. Depending on the company, the instantiation of these data types will be different, with the possible exception of Amount paid. For example, the "Products" of Amazon.com would be different from the "Products" of eBay.com and the CPD for CIBC.com may be different from that for TD.com (another Canadian bank). Thus, each customer order may be represented by a data collection as shown in Figure 2. We wish to emphasize

that there is no implied ordering of the data types in Figure 2, i.e., Figure 2 does not state that the data types should be stored in any particular order one after the other. These data collections would be stored by the company in its own databases, which may be on company premises or on a cloud server. If the company were to suffer a data breach, this data (including CPD) would be exposed.

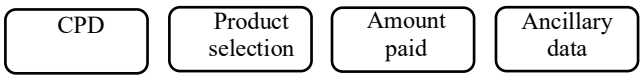


Figure 2. Data collection for a customer order.

IV. Approach

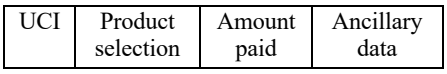
This section details our approach for minimizing the loss of PD from data breaches.

A. Strategy for Storing a Customer's Personal Data

The goal of this strategy is to reduce the storage of personal data on the company's computer system by storing the bulk of the personal data on customers' own devices, while allowing for all the purposes described in Section III-A to be carried out. The strategy consists of five parts, as follows:

1. Identification of data (Figure 2) to be stored on the customer's device: CPD.
2. Design for linking the data on the customer's device to the rest of the data stored on the company's computer system: Use a Unique Customer Identifier (UCI) that the company assigns to each customer. The UCI is the hash (e.g., SHA-3) of the customer's User ID and password for accessing the company. It will form part of the records shown in Figure 3 (shown as relational records without loss of generality since we could have shown them as other types of data structures, e.g., linked lists).

a) Record of personal data stored on customer's device.



b) Record of order data stored on company's system.

Figure 3. Data records corresponding to a customer order. Encrypted data types are shaded.

3. Design for enabling the company to carry out its communication purpose: Use the "Contact information" data record in Figure 4 to contact the customer, where "Contact information" consists of email address and telephone number. Figure 5 shows how the UCI links the three types of data records together.
4. Design to keep the CPD record should the customer a) use a new device with the company after using other devices, or b) loses a device used with the company. For a), the customer can register a new device with the company on its website after logging in. The company would then transfer the CPD record from a previously used device (on which the customer is also logged in) to the new device. For b), the customer may have used other devices with the company and wishes to replace the lost device, in which case the resolution for a) applies. If the lost device is the only device used with the company, the customer would need to re-enter his/her CPD. See also the third paragraph of Section IV-C below.

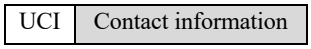


Figure 4. Data record for a customer's contact information. Encrypted data types are shaded

5. Enabling security: Use authenticated symmetric encryption (e.g., AES-GCM [11]) to encrypt the UCI and CPD in Figure 3(a), as well as the Contact information in Figure 4 (encrypted data types are shaded). The UCI in Figure 4 is not encrypted. The UCI and remaining data types in Figure 3 (b) are not encrypted, as it would be difficult for the attacker to use them alone to identify the customer, should the data be breached.

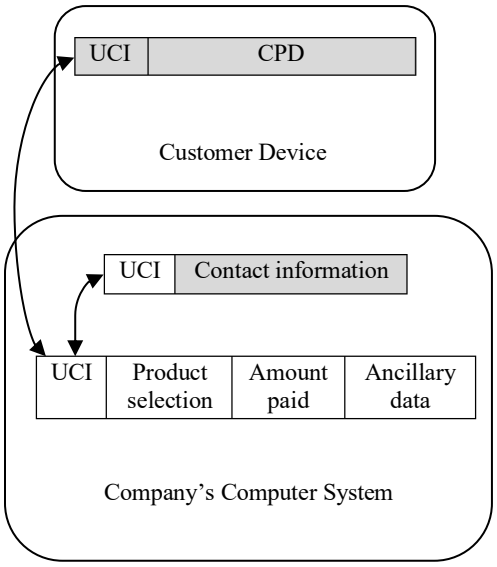


Figure 5. How the UCI links data records together.

B. Customer Walk-Through of the Strategy

1. The customer accesses the company using its website, running either on a desktop computer or on a mobile device such as a smart phone or tablet. In the following, all data transfers between the user's device and the company's system is done though a secure channel (e.g., TLS).
2. If it's the customers first use of the website on this device (detected by the absence of the CPD record), he/she will be asked if he/she has a different device that was used

with the website. If not, he/she will be prompted to enter his/her CPD. The company then generates the UCI, forms the record in Figure 3(a), encrypts it, and stores this encrypted record on the customer's device. The company then uses the unencrypted CPD entered by the customer for processing the current order. In addition, the company checks if the customer's Contact information is already in the system (possible if the customer's device was lost or stolen) and if not, creates and stores the record in Figure 4, after encrypting the Contact information (obtained from the CPD). If the customer has used the website before on a different device, he/she will be asked to also login using the other device, at which point the company stores the CPD record from the old device on the new device, decrypts the CPD record, and uses it for the current transaction.

If the customer has used the website before on this device (detected by the presence of the encrypted CPD record), the company automatically retrieves the encrypted CPD record (Figure 3(a)) from the customer's device and decrypts it for use in the current transaction.

Note that the only time the company retrieves the CPD record from a customer device is when the customer logs in to do a new transaction.

3. The customer proceeds with his/her shopping. Once the customer completes the shopping, the company creates and stores the customer's order data record as shown in Figure 3(b). Note that this record may have to be updated for some ancillary data (e.g., date delivered) once the data is available. This update process as out of scope for this work.

Figure 6 shows a message sequence diagram illustrating the case where the customer uses a device with the company's system for the first time and has not used any other device with the company in the past. Figure 7 presents a message sequence diagram for the case where the customer uses a device with the company that he/she has used before. Figure 8 gives a message sequence diagram depicting the case where the customer uses a device with the company for the first time and has used a different device with the company before.

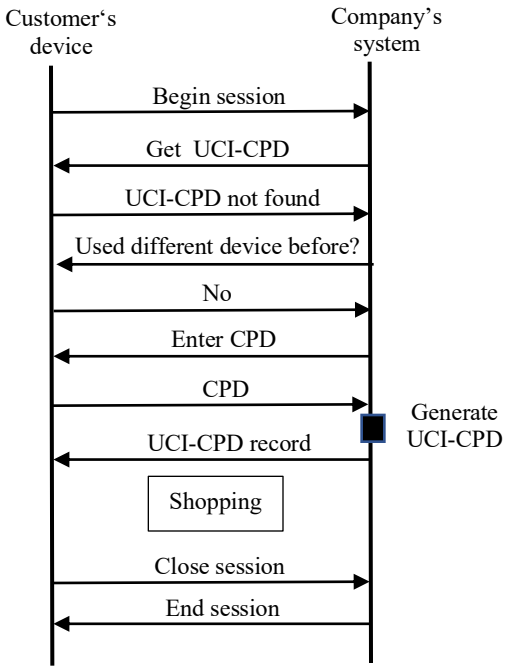


Figure 6. Customer uses a device with the company for the first time and has not used any other device before.

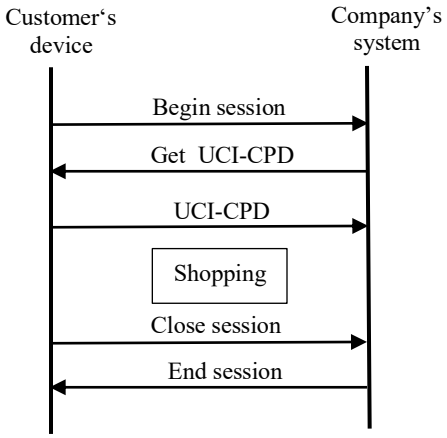


Figure 7. Customer uses a device with the company that he/she has used before.

C. Security Analysis

We first consider outside attacks against the company. Such attacks would result in breaching the company's data stores leading to the loss of the Contact information and the order data (Figure 5). This loss could be in the form of a copy taken of the data, deletion of the data from the company's data stores, modification of the data in the company's data stores, or certain combinations of these, namely copy followed by deletion, and copy followed by modification. However, the attacker fails to read the Contact information since it is encrypted. The attacker would be able to read the UCI from both the Contact information and the order data records but the UCI would appear as meaningless (hash). The attacker could also read the order data but would have a hard time identifying the customer using only this data. Further, deleting or modifying the data will also fail to damage the company provided that the company is aware of the attack and is able to re-populate the data stores using data back-ups. We assume that the company has implemented other security measures, including making data backups and having ways to detect attacks (e.g., intrusion detection system). Any modification of the encrypted Contact information would also be detected by a failure to decrypt the modified version, i.e., the modified encrypted data fails authentication. Note that for the rest of this paper, whenever we refer to failing to decrypt attacker-modified encrypted data, we mean that the modified encrypted data has failed authentication. In any case, the probability of being attacked after applying the approach is low, since the only attraction for attackers is

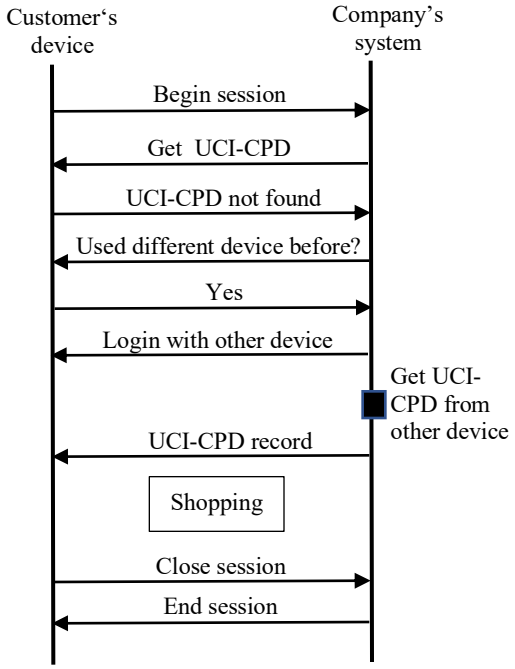


Figure 8. Customer uses a device with the company for the first time, having used a different device with the company before.

encrypted Contact information, consisting only of email address and telephone number. Attacks on the company side could also involve malware, that for example, exfiltrates the customer's CPD while in the clear. However, these attacks are not peculiar to the approach and can occur for any website that collects information from users. We assume that the company already has security measures for such attacks.

As for insider attacks against the company, we admit that our security scheme is vulnerable to such attacks. For example, an insider could simply access the CPD in its unencrypted form. Insider attacks are always among the harder ones to defend against and given their seriousness, we expect the company to have implemented other security measures (e.g., [5]) specifically against insider attacks. An exploration of these measures is outside the scope of this paper.

Attacks on the customer side with the device in the customer's possession or not (device lost or stolen) could also result in a copy taken of the customer's CPD record, deleting it, modifying it, or combinations thereof. Since the data is encrypted, the attacker would not be able to read the data if a copy is taken. Deletion or modification of the encrypted CPD record would be detected by the company's system when it fails to find it or fails to decrypt it, in which case the company's system would inform the customer that he/she needs to re-enter his/her CPD or have it transferred from another device (see Section IV-A, part 4).

The secure communication channel between the company's system and the customer device may also be attacked, but this is again not peculiar to the approach. Such attacks would be handled the same way as is done for the many other applications of secure communication channels.

D. Implementation Notes

The following are suggestions on how the above strategy should be implemented.

- On the company side, the implementation should include functionality to warn that its data stores have been compromised when it is unable to decrypt attacker-modified encrypted data, or when it finds its data stores empty. The implementation should also warn the customer that his/her device has been attacked when the encrypted CPD record was expected but is missing, or when it is unable to decrypt the attacker-modified record.
- If the customer changes or forgets his/her password for accessing the service (if forgotten, a conventional password reset procedure would be used), the company's computer system will need to generate a new UCI corresponding to the new user-ID/password combination. The company will have to create a new CPD record with the new UCI, and upload this new record to all customer devices via the website. The company will also have to update the UCI in the records of Figure 3(b) and Figure 4.
- The company's system needs to allow the customer to update his/her CPD and/or Contact information, and update the relevant records with the new information. For CPD, the system would need to upload an updated CPD record to all customer devices.

E. Verification of Purposes

We verify that the approach allows the company to carry out its purposes (Section III-A) for collecting private data.

- Transaction Requirements: The customer's CPD record is obtained from the customer's device for every transaction (either pre-existing or currently entered) and is available for carrying out the transaction.
- Communication: For contacting the customer, the customer's Contact information (Figure 4) can be obtained using the UCI link from the order data records since contacting is done for an order issue. The customer can contact the company by logging into the company's website. The company can determine the customer's UCI from the customer's User ID and password, and use it to access the contact information for the reply.
- To Secure Other Data: The personal biometric, once captured, can be stored as part of the customer's CPD record on the customer's device. Once the customer logs in for a new transaction, the CPD record is retrieved from the customer's device, at which point the personal biometric is available for use.
- Establishing Loyalty: The company has access to a customer's order history in the form of the order data records. These records (Figure 3(b)) are identified as belonging to a particular customer through the UCI link to the Contact information records. The company can thus establish the loyalty of a particular customer.

- Targeted Advertising: Understanding the type of products a customer has purchased in the past may be done by accessing the customer's order data records, as explained above for establishing loyalty.
- Market Research: The histories of past transactions for all customers can be studied by accessing the order data records, ignoring the UCI in each order record, since there is no need to identify the customers. We assume that market research is carried out without the CPD records, since the company probably does not have the customer's consent for such use of his/her CPD. If the company does require the CPD records, the company can always capture and store them, but would have to accept the risks of those records being breached and being sued for illegally using the CPD for market research.
- Sharing or Selling: There is nothing stopping the company from copying each customer's CPD record and sharing or selling the data. The company would have to accept the risks of the CPD records being breached and being sued for illegally sharing or selling the customer's CPD.

F. Strengths and Weaknesses of the Approach

The approach has the following strengths: a) it is straight-forward, which may make it easier to "sell" to upper management for approval, b) it is efficient in that attackers would have to breach the devices of all the company's customers, in order to breach the same quantity of personal data that are traditionally all stored in the company's system, c) it minimizes the risk of data loss (see subsection G below), d) it makes the company less attractive to attackers who intend to cause a data breach due to its efficiency as stated above and the fact that the only private data left on the company's system to be breached is the encrypted customer Contact information, and e) it should please customers who want more control over their private data, since most of it is stored only on their own devices.

The approach seems to have three weaknesses: a) the storage/retrieval of the CPD record may attract attacks on the secure transmission channel, b) there is additional overhead cost due to encryption / decryption operations, and c) it is vulnerable to insider attack. Weakness a) does not represent significant extra risk over conventional transactions since personal data is transmitted in conventional transactions as well. For weakness b), the extra overhead should not be significant. Finally, weakness c) is not exclusive to this approach, since it can arise wherever there are insiders. Potential remedies include the installation of specific security measures to defend against insider attacks [5].

G. Showing that the Approach Minimizes the Risk of Data Loss

Our approach of having most of a user's private data stored on his/her computing device rather than on the company's system minimizes data loss according to beliefs 1 and 2 as follows:

1. Much less private data is lost in the event of a system breach, because the storage of most of the private data has been relocated to user devices, and
2. There is a much-reduced risk of theft of the users' private data if that data is stored on user devices rather than stored in the company's system.

Belief 1 is self-evident. To verify belief 2, compare Case 1 where a portion of each users' private data is stored on the system, with Case 2 where the portions of private data in Case 1 are instead stored on user devices. Let D and D_i represent the private data in Cases 1 and 2 respectively, where D_i is the private data belonging to user i. Let E be the event that D is stolen in Case 1. Let E_i be the event that D_i is stolen from user i in Case 2. Let $P(E) = p$ where $P(E)$ is the probability of E. Finally, let $P(E_i) = q_i$. Figure 9 illustrates D and D_i . We postulate that for n users,

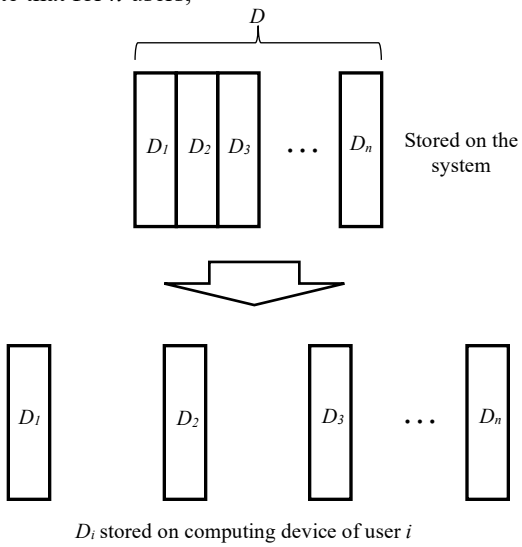


Figure 9. Moving the storage of private data from the server to user devices.

$$P(E_1 \cap E_2 \cap \ldots \cap E_n) \ll P(E) \quad (1)$$

meaning that the risk of theft of all the private data moved to user devices from the system (Case 2) is much lower than the risk of theft of that same data were it to remain on the system (Case 1), which is a statement of belief 2 above. Thus, to verify belief 2, we need to prove (1). To do this, let C be the event that an attacker chooses to attack the system. Let C_i be the event that an attacker chooses to attack the computing device of user i. Let S be the event that the attacker successfully defeats the security controls of the system. Let S_i be the event that the attacker successfully defeats the security controls of user i's device. We note that

$$P(E) = P(C)P(S|C) = p \quad (2)$$
$$P(E_i) = P(C_i)P(S_i|C_i) = q_i \quad (3)$$

What can we say about the conditional probabilities here? Since D has a lot more private information than D_i, an attacker would be more likely to choose D over D_i as his/her target. In other words, a company is a more attractive target than a user device. Thus, $P(C)>P(C_i)$ for all i. Further, the attacker would be more motivated to defeat the security controls of the company compared to the security controls of the user device, again due to the attractiveness of the company as a target. Thus, $P(S|C)>P(S_i|C_i)$ for all i. Equations (2) and (3) then give $p > q_i$ for all i. Now since the E_i are independent events, we have

$$P(E_1 \cap E_2 \cap \ldots \cap E_n) = \prod_1^n P(E_i) \quad (4)$$
$$= \prod_1^n q_i \quad (5)$$
$$< p^n \ll p = P(E) \quad (6)$$

proving (1) as we set out to do. Note that (5) follows from (4) due to $p > q_i$ and (6) follows from (5) due to the fact that p is a probability with $0 < p < 1$.

Another way to reason about (1) is simply to notice that the product in (4) decreases monotonically with increasing n due to the fact that the q_i are probabilities between 0 and 1. Thus, since $P(E)$ is fixed, (1) will be true for sufficiently large n. Since we are dealing with systems that have many users, it would not be difficult to achieve sufficiently large n.

We now have beliefs 1 and 2 both true, meaning that storing the private data on user devices instead of on the system does indeed minimize the risk of data loss.

V. Application Examples

We instantiate the data types in Figure 2 for four types of B2C companies, demonstrating that the approach can fit with different B2C companies.

Example 1: Seller of goods (e.g., Amazon.com). Table 3 shows the instantiation of the data types for this example.

TABLE 3. INSTANTIATION OF DATA TYPES FOR EXAMPLE 1.

CPD	Product selection	Amount paid	Ancillary data
Name	Camera	$159.00	Date ordered
Billing address	Hair clipper	$49.00	Date shipped
Default shipping address	Laser printer toner	$68.00	Date delivered
Alternate shipping address			Payment method
Email address			Product returned
Phone number			Reason for return
Credit card data			Refund status

Example 2: Travel Agency (e.g., Expedia.ca). Table 4 shows the instantiation of the data types for this example.

TABLE 4. INSTANTIATION OF DATA TYPES FOR EXAMPLE 2.

CPD	Product selection	Amount paid	Ancillary data
Name	Vacation package	$1059.00	Date ordered
Billing address	Trip insurance	$189.00	Date mailed
Default address			Date delivered
Alternate address			Payment method
Email address			Product returned
Phone number			Reason for return
Credit card data			Refund status

Example 3: Hotel (e.g., Mariott.com). Table 5 shows the instantiation of the data types for this example.

TABLE 5. INSTANTIATION OF DATA TYPES FOR EXAMPLE 3.

CPD	Product selection	Amount paid	Ancillary data
Name	Room - double	$200 / night	Date of reservation
Billing address			Arrival date
Home address			Departure date
Email address			Payment method
Phone number			Airport shuttle y/n
Credit card data			Daily laundry y/n
Loyalty ID number			Daily cleaning y/n
Country of origin			Wake-up call y/n
Passport country			Stay extended y/n
Passport number			
Room preferences			
Floor preference			

Example 4: Online Training (e.g., Udemy.com). Table 6 shows the instantiation of the data types for this example.

TABLE 6. INSTANTIATION OF DATA TYPES FOR EXAMPLE 4.

CPD	Product selection	Amount paid	Ancillary data
Name	Guitar	$30.00	Date of purchase
Billing address	Photography	$50.00	Date training started
Home address	Programming	$60.00	Date training ended
Email address			Certificate issued y/n
Phone number			Comprehension test taken y/n
Credit card data			Comprehension score
Training type preferred			Comprehension score issued y/n
Training length preferred			

We could have included other examples here, but the above examples suffice for demonstrating that the approach can be applied to different types of B2C companies.

VI. Related Work

Work that is most closely related to this work are as follows: Aggarwal et al. [12] propose that an organization outsource its data management to two untrusted servers to break associations of sensitive information. They show how the use of two servers, together with the use of encryption where needed, enables efficient data partitioning and guarantees that the contents of any one server does not violate data privacy. However, it is unclear if attackers can re-construct the sensitivity associations by breaching both servers. Ciriani et al. [13] present what they claim to be a solution that improves over Aggarwal et al. [12] by first splitting the information to be protected into different fragments so that sensitive associations represented by confidentiality constraints are broken, and minimizing the use of encryption. The resulting fragments may be stored at the same server or at different servers. Our work differs from Aggarwal et al. [12] and Ciriani et al. [13] as follows: a) the above two papers are solutions for securing databases, whereas our work is focused on reducing the loss of data in the event of a data breach by simply not storing some of the data in the company's computer system, b) we do not use data partitioning or fragmentation; rather, our data is distributed between the company and its customers from the point of data creation, c) we do not need to rely on breaking any sensitivity associations, d) our approach has been designed to satisfy the business needs of the organization, and e) our approach is more straightforward, and is therefore easier to apply.

Other work in the literature mostly deal with the prevention or risks of data breaches, the discovery of a data breach, and the aftermath of a data breach. Within these categories, the most closely related works have to do with preventing or evaluating the risks of data breaches. We describe some of these papers below, to give the reader a sense of this research. Note that these works all differ from this paper in that this paper aims to minimize the data lost if a breach were to happen, whereas the works described in the following are largely focused on preventing breaches from happening. Panou et al. [14] describe a framework for monitoring and describing insider behaviour anomalies that can potentially impact the risks of a data breach. The framework also enhances a company's understanding of cybersecurity and increases awareness of the threats and consequences related to breaches, and eventually enable faster recovery from a breach. Guha and Kandula [15] propose a data breach insurance mechanism together with risk assessment methodology to cover the risk from accidental data breaches and encourage best practices to prevent the breaches. They also present data supporting the feasibility of their approach. Zou and Schaub [16] interviewed consumers after the Equifax data breach and discovered that consumers' understanding of credit bureaus' data collection practices was incomplete. As such, consumers did not take sufficient protective actions to deal with the risks to their data. The authors describe the implications of their findings for the design of future security tools with the aim of empowering consumers to better manage their data and protect themselves from future breaches. Nicho and Fakhry [17] look at the application of system dynamics to cybersecurity, specifically to the Advanced Persistent Threat (APT) that can employ technical, as well as organizational factors to cause a data breach. They applied system dynamics to the APT that led to the Equinox breach and identified key independent variables contributing to the breach. Their work provides insights into the dynamics of the threat and suggests "what if" scenarios to minimize APT risks that could lead to a breach. Luh et al. [18] present an ontology for planning a defence against APTs that can lead to a data breach. The ontology is mapped to abstracted events and anomalies that can be detected by monitoring and helps with the understanding of how, why, and by whom certain resources are targeted. Other references in this category are readily available.

In terms of identifying and reducing the attack surface, this work is unique in reducing the attack surface of a company's system by storing private data on user devices. This author has published works [19][20][21] that deal with reducing the attack surface during software design, by identifying vulnerabilities using a model of the software system under development. A. Kurmus et al. [22] look at reducing the attack surface of commodity OS kernels by identifying code that is not used and removing it or preventing it from executing. T. Kroes et al. [23] investigate reducing the attack surface through dynamic binary lifting, removal of unnecessary features, and recompilation. M. Sherman [24] investigates attack surfaces for mobile devices. This author claims that mobile devices exhibit attack surfaces in capabilities, such as communication, computation, and sensors, that are generally not considered in current secure coding recommendations. C. Theisen et al. [25] propose the use of risk-based attack surface approximation (RASA) which uses crash dump stack traces to predict what code may contain attackable vulnerabilities. Their goal is to help software developers prioritize their security efforts by providing them with an attack surface approximation. It is worthwhile noting that some works propose to increase security through attack surface expansion rather than attack surface reduction. For cloud services, T. Al-Salah et al. [26] propose three attack surface expansion approaches that use decoy virtual machines co-existing with the real virtual machines in the same physical host. They claim that simulation shows that adding the decoy virtual machines can significantly reduce the attackers' success rate. For enterprise networks, K. Sun and S. Jajodia [27] propose a new mechanism that expands the attack surface, so that attackers have difficulty in identifying the real attack surface from the much larger expanded attack surface. Note that these two works do not contradict reducing the attack surface to improve security, since the attack surface is not really expanded but only appears to be expanded due to the addition of decoys.

VII. Conclusion and Future Work

We have presented an attack surface reduction approach, applicable to B2C e-commerce companies, that minimizes the loss of private data in the event of a data breach by storing most of a customer's private data in his/her own device rather than in the company's computer system. This redistribution of private data reduces the attack surface of the company's system, minimalizing the amount of data that would be lost in a breach. Not all of the private data is moved to the customer's device since we still allow some necessary personal data (customer contact information) to be stored on the company's system. We also verified that the approach allows the company to carry out its purposes for collecting private data, which is an important requirement of any company that may wish to implement this approach. Some readers may consider the approach overly simple, but if a simpler solution gets the job done, it should be preferred over a more complex solution. As well, a large contribution of this work is showing how the approach can be done securely. We look forward to readers' feedback and correcting any inadvertent omissions, if found, in a future paper.

In terms of future work, we would like to explore the application of the approach to other types of businesses and organizations, and adapt it where necessary. We would also like to have implementations of the approach in order to fine tune it, measure implementation effort, and check performance.

Acknowledgment

The author is grateful to Aptusinnova Inc. for financially supporting this work. He expresses his thanks to the reviewers of this paper for their insightful comments.

References

[1] G. Yee, "Towards Reducing the Impact of Data Breaches," Proc. Fourteenth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2020), November 2020, pp. 75-81.

[2] M. Heiligenstein, "Top 10 Biggest Data Breaches of 2022 – So Far," Firewall Times, [retrieved: October, 2022] https://firewalltimes.com/biggest-data-breaches-2022/

[3] G. Yee, "Visualization and Prioritization of Privacy Risks in Software Systems," International Journal on Advances in Security, issn 1942-2636, vol. 10, no. 1&2, pp. 14-25, 2017, [retrieved: Dec., 2020] http://www.iariajournals.org/security/

[4] E. Aïmeur and M. Lafond, "The scourge of Internet personal data collection," Proc. 2013 International Conference on Availability, Reliability and Security (AReS 2013), Sept. 2013, pp. 821-828.

[5] CERT National Insider Threat Center, "Common Sense Guide to Mitigating Insider Threats, Sixth Edition," Technical Report CMU/SEI-2018-TR-010, Software Engineering Insititute, Carnegie Mellon University, December 2018.

[6] C. Salter, O. Sami Saydjari, B. Schneier, and J. Wallner, "Towards a Secure System Engineering Methodology," Proceedings of New Security Paradigms Workshop, Sept. 1998, pp. 2-10.

[7] P. K. Manadhata and J. M. Wing, "An Attack Surface Metric," IEEE Transactions on Software Engineering, vol. 37, no. 3, pp. 371-386, May/June, 2011.

[8] Business News Daily, "How businesses are collecting data (and what they're doing with it)," [retrieved: October, 2020] https://www.businessnewsdaily.com/10625-businesses-collecting-data.html

[9] R. Sarcar, "How to set up an ecommerce loyalty program to improve retention, build community and drive 5X in sales," [retrieved: October, 2020] https://www.bigcommerce.com/blog/online-customer-loyalty-programs/#how-to-create-and-implement-a-customer-loyalty-program

[10] PC, "How companies turn your data into money," [retrieved: October, 2020] https://www.pcmag.com/news/how-companies-turn-your-data-into-money

[11] M. Dworkin, "Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC," NIST Special Publication 800-38D, November 2007.

[12] G. Aggarwal et al., "Two can keep a secret: a distributed architecture for secure database services," Proc. Second Biennial Conference on Innovative Data Systems Research (CIDR 2005), Jan. 2005, pp. 1-14.

[13] V. Ciriani et al., "Combining fragmentation and encryption to protect privacy in data storage," ACM Transactions on Information and System Security (TISSEC), Vol. 13, Issue 3, article 22, pp. 1-33, July 2010.

[14] A. Panou, C. Ntantogian, and C. Xenakis, "RiSKi: A framework for modeling cyber threats to estimate risk for data breach insurance," Proc. 21st Pan-Hellenic Conference on Informatics (PCI 2017), article no. 32, Sept. 2017, pp. 1-6.

[15] S. Guha and S. Kandula, "Act for affordable data care," Proc. 11th ACM Workshop on Hot Topics in Networks (HotNets-XI), Oct. 2012, pp. 103-108.

[16] Y. Zou and F. Schaub, "Concern but no action: consumers' reactions to the Equifax data breach," Extended Abstracts, 2018 CHI Conference on Human Factors in Computing Systems (CHI EA '18), paper no. LBW506, Apr. 2018, pp. 1-6.

[17] M. Nicho and H. Fakhry, "Applying system dynamics to model advanced persistent threats," Proc. 2019 International Communication Engineering and Cloud Computing Conference (CECCC 2019), Oct. 2019, pp 29-33.

[18] R. Luh, S. Schrittwieser, and S. Marschalek, "TAON: an ontology-based approach to mitigating targeted attacks," Proc. 18th International Conference on Information Integration and Web-based Applications and Services (iiWAS '16), Nov. 2016, pp. 303-312.

[19] G. Yee, "Reducing the Attack Surface for Sensitive Data," International Journal on Advances in Security, issn 1942-2636, vol. 13, no. 3&4, pp. 109-120, 2020, [retrieved: Oct., 2022] http://www.iariajournals.org/security/

[20] G. Yee, "Modeling and Reducing the Attack Surface in Software Systems," Proceedings, 11th Workshop on Modelling in Software Engineering (MiSE'2019), May 2019, pp. 55-62.

[21] G. Yee, "Attack Surface Identification and Reduction Model Applied in Scrum," Proceedings, 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), June 2019, pp. 1-8.

[22] A. Kurmus, A. Sorniotti, and R. Kapitza, "Attack Surface Reduction for Commodity OS Kernels: Trimmed Garden Plants May Attract Less Bugs," Proceedings of the Fourth European Workshop on System Security (EUROSEC '11), April 2011, article no. 6 (no page number available).

[23] T. Kroes, A. Altinay, J. Nash, Y. Na, and S. Volckaert, "BinRec: Attack Surface Reduction Through Dynamic Binary Recovery," Proceedings of the 2018 Workshop on Forming an Ecosystem Around Software Transformation (FEAST '18), October 2018, pp. 8-13.

[24] M. Sherman, "Attack Surfaces for Mobile Devices," Proceedings of the 2nd International Workshop on Software Development Lifecycle for Mobile (DeMobile 2014), November 2014, pp. 5-8.

[25] C. Theisen, B. Murphy, K. Herzig, and L. Williams, "Risk-Based Attack Surface Approximation: How Much Data is Enough?," Proceedings of the 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP '17), May 2017, pp. 273-282.

[26] T. Al-Salah, L. Hong, and S. Shetty, "Attack Surface Expansion Using Decoys to Protect Virtualized Infrastructure," Proceedings of the 2017 IEEE International Conference on Edge Computing (EDGE), June 2017, pp. 216-219.

[27] K. Sun and S. Jajodia, "Protecting Enterprise Networks through Attack Surface Expansion," Proceedings of the 2014 Workshop on Cyber Security Analytics, Intelligence and Automation (SafeConfig '14), November 2014, pp. 29-32.

Improving IT Security of Medical IoT Devices:

A Maturity Evaluation and a Labeling Approach

Michael Gleißner[1,2], Johannes Dotzler[1], Juliana Hartig[1], Andreas Aßmuth[2], Clemens Bulitta[1] and Steffen Hamm[1]

Technical University of Applied Sciences OTH Amberg-Weiden
[1] Hetzenrichter Weg 15, 92637 Weiden, Germany
[2] Kaiser-Wilhelm-Ring 23, 92224 Amberg, Germany

Email: {m.gleissner | jo.dotzler | j.hartig | a.assmuth | c.bulitta | s.hamm}@oth-aw.de

***Abstract*—The healthcare industry worldwide is currently being transformed by digitization and the Internet of Things. As the level of digitization increases, the number of devices within a network of a healthcare facility grows exponentially. The consequential complexity of the infrastructure poses a substantial challenge for IT professionals to keep their networks secure. This paper aims to provide two different ways to aid administrators and decision makers to help integrate the increasing amount of interconnected medical devices into their infrastructure more securely. Additionally, two mobile ultrasonic scanners were tested in regard to their security as well as privacy to show where problems with such devices might occur.**

Keywords*—*Internet of Things; healthcare; Medical IoT; cloud services; IT security; IoT labeling; IoT evaluation.

I. Motivation

With the rise of the Internet of Things (IoT), IT security has become an ever increasing challenge. Additionally, one of the main reasons why the focus on IT security is going to be amplified is the fact that future communication networks will be based on software-defined networks (SDN). SDNs will be exposed to a large number of known attack vectors, which are already available on the market since SDNs are increasingly implemented using architectures similar to the Representational State Transfer (REST) schematic. Therefore, attacks can be carried out by anyone without specific expert knowledge. This risk is consciously accepted, and solutions are developed for it. The reasoning is that potential gains for industries that come with Next Generation Mobile Networks (NGMNs) exceed the known risks. Potential benefits of NGMNs include, for example, network-slicing or SIM provisioning. From an economic point of view, NGMNs require new use cases, e.g., the density of connected IoT devices, to make it a profitable investment for adopters. The most promising adoption of 5G networks in that context is the so-called "massive Machine Type Communication" (mMTC) [2]. New use cases are still evolving, for example, for branches like public safety, the automotive industry, healthcare, factory automation etc. These use cases are based on the concepts of IoT and promise a steep increase in productivity across a variety of business processes and industries [3]. To implement these use cases, it will be necessary to integrate and administrate up to 100,000 devices per 1 km² [4] in the future. This is going to present a challenge that needs to be carefully considered. Undoubtedly, managing such a massive IoT ecosystem demands highly secure architectures and certified processes with a strong focus on IT security, particularly for healthcare providers. The goal of this paper is to provide two different methods for healthcare facilities to improve their IT security posture. A maturity model is presented, which provides guidance on how an environment for Medical IoT (MIoT) integration needs to be shaped in order to maintain a secure infrastructure while still reaping the benefits of the devices. Additionally, a labeling concept is shown. This concept allows personnel responsible for procuring MIoT devices to quickly assess if a product fulfills the minimum requirements to be considered secure for integration. It is meant as a supporting tool for gaining a brief overview rather than a replacement for an in-depth security analysis of the MIoT device. The rest of the paper is structured as follows. In Section II, a brief review of currently published IoT security-related reference models from accredited stakeholders, e.g., industry associations, consortia and alliances, are presented. Section III highlights the background to understand the topic and underlines essential aspects. Section IV introduces a majority model focusing specifically on the healthcare sector. Section V presents a labeling approach for technology to empower healthcare facilities and consumers. Section VI outlines a security test of two IoT devices in detail. At last, an outlook and future thoughts are given.

II. Recent Works

This section presents a brief literature overview of common standards and reference models targeting IoT-related security models and architectures by accredited consortia, alliances and standardization bodies. During the literature research, it turned out that relevant standardization efforts mainly originate from the manufacturing and production sector. Consequently, it is not surprising that most (industrial) IoT reference models, architectures and blueprints target manufacturing and production sites and are, therefore, not fully compatible with the healthcare sector. A possible reason for these one-sided efforts could be the fact that several government programs, e.g., "Industrie 4.0" from Germany [5] or the "Made in China 2025" initiative from the Chinese government [6] have been established.

Literature research of already existing IoT reference models targeting manufacturing and surrounding topics has already been conducted by many researchers. Some examples are the research from Nakagawa et al. with the title "Industry 4.0 reference architectures: State of the art and future trends" [7] and the work of Mazon-Olivo and Pan titled "Internet of Things: State-of-the-art, Computing Paradigms and Reference Architectures" [8]. For the sake of completeness, some well-recognized reference models are mentioned. These are the "Referenzarchitekturmodell Industrie 4.0" (RAMI4.0) [9], the "NIST Smart Manufacturing Ecosystem" [10] and the advanced IoT reference models for the Internet of Things from the "IoT World Forum Reference Model" [11]. Also, the European Union published a consolidated IoT standard and announced the "3D Reference Architecture Model" [12].

Those architectures and frameworks targeting IoT security all share that security cannot be achieved by merely applying software and / or technologies, e.g., blockchain, alone. Security has to be an integral part even before the beginning of the actual development process. During this process, the type of users and the intended use cases are vital parameters to consider. Some alliances apply security-relevant topics to the entire supply chain. Starting with the component manufacturer (producer of hardware, e.g., chips and processors), over to retailers and operational users. The goal is to provide recommendations targeting those specific groups to security by design into practice, which results, for example, in the (Hardware) Root of Trust ((H)RoT) [13]. Others shed light on detailed processes, e.g., on an auditable and verifiable boot process [14], when setting up and integrating IoT devices in an existing network.

From a German perspective, the Federal Office for Information Security published in their recommendations several useful proposals for how IoT devices can be used safely in institutions [15] and how they can be operated securely [16]. These recommendations might find attention in well-financed production industries with up-to-date IoT devices, which are intended to perform their tasks in a network. But the reality shows an entirely different picture because other industries, e.g., the German healthcare industry, cannot rely on up-to-date equipped departments, and thus, some outdated medical devices will be modified to act as IoT devices. This approach leads to a very error-prone infrastructure. A scenario has been considered in this manner neither by the publications of the Federal Office for Information Security nor by other sources listed above. Another critical topic is the kind of data in the healthcare industry. Health data or data related to patients need to be treated with special care because this data describes various medical conditions of people and can cause damage in the wrong hands. In order to implement a legal basis, the EU states in its General Data Protection Regulation (GDPR) [17] a set of regulations which enforces compliant handling of personal data. This is done to handle potential misconduct of such sensitive information (e.g., healthcare data). A proper application of the GDPR needs to be considered, especially in the healthcare sector, where highly sensitive data is collected [18]. Additionally, a certain set of rules and requirements need to be defined in order to provide a minimum in the safety and security of the technology used in practice. Therefore, it is imperative to answer the corresponding questions about what IoT devices will be deployed and thus purchased and integrated in a future healthcare environment. This paper aims to outline the common understanding and need for the definition of references related to safety and transparency labeling models, focusing on the healthcare industry in Germany. Complementary to this, a maturity model for Medical IoT devices is proposed, which allows to evaluate if a secure integration of these products by the corresponding actor is possible. Section V proposes a concept that will enable consumers to obtain a comprehensive picture of the functionality, built-in components, generated data and responsible parties of an IoT device. This allows customers to gain an overview of the corresponding product even before purchase. Last but not least, two exemplary Medical IoT devices are examined to show the current deficits of the industry in regards to IT security. In the following section, the mentioned assessment model is introduced.

III. Background

IoT is now influencing many areas of business and private life and is gaining increasingly technical, social and economic importance. IoT can be defined as "an emerging concept comprising a wide ecosystem of interconnected services and devices, such as sensors, consumer products and everyday smart home objects, cars, and industrial and health components" [19]. This work focuses on IoT devices in the healthcare sector. It aims to be an extension to our previous work in [1] with the goal of analyzing MIoT with a focus on IT security. Especially in the healthcare sector, device failure can have devastating consequences for human beings.

The reason for this is the increasing focus on digitization in the healthcare sector. The introduction of the new 5G mobile network will enable better and more efficient connectivity between IoT devices, which means that the number of these devices in the sensitive healthcare environment is expected to grow exponentially over the next few years. In numbers, this means that USD 60.83 billion were spent on IoT devices in the healthcare sector in 2019, whereas in 2027, the investment is expected to reach USD 260.75 billion [20]. The goals of MIoT devices are, among others, to reduce the workload of medical staff, to make diagnostics more efficient and safer, and to make everyday life easier for patients. One possible way might be monitoring interconnected devices in the network to analyze utilization, location or maintenance intervals. A reduction in search times and an increased efficiency in the use of equipment (e.g., mobile ultrasound scanners) are potential benefits.

IV. Maturity Model for Secure Medical IoT Integration

The following model is embedded in the 5G4Healthcare research project, which is briefly presented below to provide context.

A. Project 5G4Healthcare

The 5G4Healthcare project (5G4HC) at the Technical University of Applied Sciences Amberg-Weiden (OTH-AW) is one of six research projects funded under the 5G Innovation Program of the German Federal Ministry for Digital and Transport. The project's objective is to establish a platform based on 5G technology on what digital applications can be integrated into healthcare scenarios. The scenarios will focus on measurable improvements in the effectiveness and efficiency of healthcare delivery. The project also aims to explore opportunities and limitations in improving healthcare delivery through 5G. Primarily related to the two defined use cases "Homecare" and "Integrated Care", the opportunities and potentials of the 5G technology in healthcare will be explored. Part of the 5G4HC project is developing an evaluation model specifically for the digital health sector. Based on the work done on the general evaluation model, the following model was developed for Medical IoT devices with a focus on the secure integration of MIoT devices in healthcare facilities. The methodology of the general model is explained below.

B. Methodology of the General Evaluation Model

The model developed takes the essential aspects of established evaluation systems in a mixed-method approach and combines them to form a new evaluation model. Initially, the basis for this system is the model of the European Foundation for Quality Management (EFQM) [21]. The EFQM model is based on a comprehensive analysis of elements in three levels: structure, process and result relevant to quality. In its original model, it is divided into a total of nine criteria and subdivisions (e.g., management, personal, law and regulatory, etc.). These criteria have been adapted for the Medical IoT model (see Section IV-C). In the next step, the sub-dimensions of the EFQM model are assessed using the systems of a maturity model. These five maturity levels are divided into beginnings (1), first steps (2), on the way (3), developed organization (4) and mature organization (5).

The essential novelty of the developed evaluation model consists in the systems that a holistic consideration will take place by means of the nine sub-dimensions. The intention is to ensure that the results provide a weighted statement about the development status of a technology, a process or even an entire system.

C. Methodology of the Medical IoT Model

The generic evaluation model is modular. One module was adapted Medical IoT devices, with IT security as the main criterion. There are many recommendations on IT security of IoT devices in the international literature (see Section II). However, the market has lacked a separate elaboration tailored to integrating Medical IoT devices into a healthcare environment so far. The following assessment model is intended to fill this gap. Based on the recommendations for general IoT devices from industrial and other sectors [22], an overview was created that includes special conditions for the medical industry. The available literature includes recommendations and guidelines from organizations such as the IoT Security Foundation, Industrial Internet Consortium (IIC), Online Trust Alliance (OTA), European Union Agency for Cybersecurity (ENISA), and many other official entities. The criteria found in these guidelines were thus adapted to this specific use case in the healthcare sector and divided into five maturity levels. Before explaining the model, the specifics of the healthcare sector will be discussed.

Normally, Medical IoT devices are used by medical personnel. Both doctors and nurses operate diagnostic and therapeutic IoT devices. It can therefore be assumed that the user has a low level of digital competence. Furthermore, medical personnel is under time pressure in their daily work. Due to staff shortages or acute indications of patients, seconds can play a decisive factor in care. In the context of Medical IoT devices, this means that failures or complex handling are not suitable to be an actual relief for the staff. Dedicated IT personnel are also typically few to nonexistent and require a broad knowledge of medical devices. It is common, especially in outpatient practices, that no trained IT staff is on site. Instead, separate external companies that have a 24-hour response time are used. In the medical sector, the availability of IoT devices must therefore be close to 100 %, especially for critical applications. Otherwise, the well-being of patients is at risk. Another critical point is the environment the Medical IoT devices have to be integrated into. The IoT devices must be embedded into existing infrastructure. However, in most cases, that infrastructure is outdated, especially in hospitals, nursing homes and outpatient practices, which directly impacts IT security. Even if an IoT device was developed and sold by the manufacturer using the security-by-design approach, there is still a risk of unauthorized access or tampering simply because of the infrastructure in the healthcare facility. To minimize this risk, investments in infrastructure need to be made, highlighting the next problem in the healthcare sector: Lack of financial means. Depending on the country, healthcare facilities have a different financing structure. Facilities can be governmental, private or public nonprofit. Government health facilities, in particular, often lack the money for new investments. Primarily, financial investments are made in more urgently needed areas, such as additional staff or an expansion of treatment services. Investment in infrastructure is rarely the first priority. These particular problems make it clear that, from an IT security perspective, a good and trustworthy environment cannot be assumed. However, there is hope for the future. Many countries (e.g., Germany with the Hospital Future Act) are switching to state support for digitization in healthcare facilities. The potential funding amounts are enormous depending on the country (e.g., in Germany, 4.3 billion euros in 2021). These subsidies should be used urgently

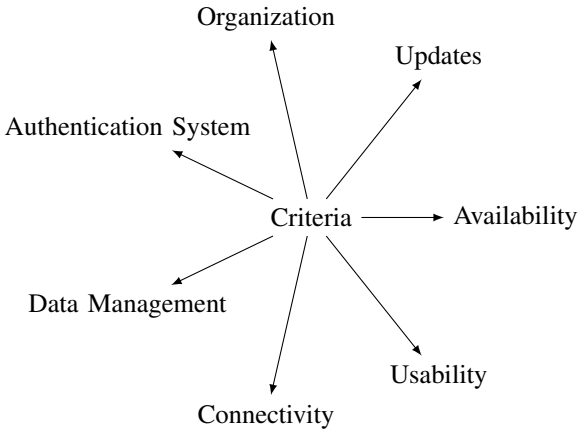


Figure 1. Criteria as basis for the evaluation model

to make the environment in healthcare facilities more secure, as the potential risk here is exceptionally high for the reasons mentioned. In the following, however, we will still focus on Medical IoT devices, as many threats can be prevented through good preparation and a structured approach. To structure the evaluation of the Medical IoT integration process and thus reduce complexity, seven criteria were formed similar to the generic evaluation model (see above). These can be seen in Figure 1.

These criteria were divided into the dimensions of structure, process, and outcome according to the Donabedian approach [23]. Following the approach, means that with a good structure and a good process based on it, a good result is automatically achieved. The dimensions thus build on each other and influence each other. All criteria were deliberately chosen to provide a controlled setting for Medical IoT devices to be embedded into. By providing such an environment, security and safety for staff and patients can be significantly improved both during the implementation phase and during regular operation. All detailed criteria can be seen in the appendix.

The system in the maturity model states that all criteria of one level must be fulfilled to attain the next higher level. For example, even if individual criteria of level four are fulfilled, but one criterion from level three is still not fulfilled, the IoT device is only awarded level three. The three matrices for evaluation can be found in the appendix. After shedding light on the maturity model, the upcoming sections refer to a concept that aids consumers in their decision-making regarding IoT products.

V. Sovereign Technology Labeling

The idea is to provide visual indications for products (e.g., IoT devices) to help consumers and decision-makers in sensitive and critical industries. The healthcare industry must provide accessible and understandable information about MIoT devices that go beyond price and functionality. That information needs to be even more detailed if IoT interacts closely with humans. To be able to perform an adequate evaluation of MIoT devices, some kind of additional labeling (obligation) might be helpful. A system of this kind could be a beneficial addition to the evaluation model presented in Section IV, making it easier to assess criteria such as data management or updates. The labeling should reflect key figures that best represent individual IT security-related aspects. Looking at hardware, labels should be displayed on the respective product packaging or the devices themselves. For software, on the other hand, a digital indication should be given before the final purchase / sign-up is made. Furthermore, future IoT devices must also be equipped with an expiry date that clearly reflects a time frame for action to be taken in order to further continue the use of integrated hardware components, installed software and intended operating environments.

An already applied and working analogy, which proves the increase of safety and security, can be noticed in the food industry and their products. The food industry must ensure that its products do not cause any harm to consumers. That is the reason why various procedures have been developed to increase the safety of food. To make the safety aspect transparent to consumers, various information and visual labels have been developed and put on products. Guidelines and labels could also be a foundation and possible approach to evaluate different technologies and their adaptations in products, e.g., IoT, software, or services. These guidelines and labeling requirements for food products are defined precisely and even required by law. Almost every country has governmental regulations for food safety, for example, the food regulations introduced by the European Union [24]. The quality assurance tools and mechanisms for the food industry have already proven that it is possible to shift the issue of safety to the manufacturer and, thus, away from the consumer. It would be appropriate to establish such guidelines for technology as well. An example of a mapping of food safety scenarios applied to technology is presented as follows:

$$\text{Nutrition Facts Label} \rightarrow \text{IoT Components Facts Label}$$
$$\text{Food Handling} \rightarrow \text{IoT Lifecycle Facts}$$
$$\text{The Best Before Date} \rightarrow \text{IoT Best Before Inspection Date}$$

A first approach is presented to map necessary information from the nutrition to the IoT domain by declaring precisely what components, protocols etc., were integrated or used during operation. It should be noted that this approach is not supposed to end up with a confusing set of different labels. One or two labels that make the most important indicators available must be sufficient to allow the consumer to make a quick and comprehensive assessment. A QR code will be provided should there be a need for more in-depth details. At the moment of writing, a list of parameters, which should be displayed, has not been defined. It is emerging that the areas affecting the human in this context will be a focus point. Until now, these areas are hardware, software and data (flow). The standard IoT component facts label suggested above would be a first step towards a more transparent evaluation of an IoT device itself and supports decision-makers to evaluate IoT devices in more detail. Specifications may vary depending on the

IoT Components	Used in Item
Sensors / Detectors	**real time, post processing**
1. Temperature Sensor	Temperature in Celsius
2. Location Sensor	GPS, Latitude and Longitude
Actuators	
1. Electric Motor	Rotation
Connectivity / Network	Cloud / Local
1. Protocol Name	MQTT
2. Protocol Name	Bluetooth
Gateways	
Cloud Location	Italy, EU
Storage Location	Netherlands, EU
Responsible Entity	Company name, phone, email, country
Stored Attributes	**Name / Cycle**
1. Attribute	GPS (latitude and longitude) / every minute
2. Attribute	Temperature / every minute
User Control	App, device itself
Deletion of Date	**Easy to complex**
1. Electric Motor	Rotation

TABLE I. An example of a possible IoT component labeling approach

product category, intended use and criticality. The following section presents the second part of the presented approach with all relevant runtime facts of a specific MIoT device that needs to be measured and labeled by the manufacturers.

A. IoT Runtime Facts

The IoT runtime facts provide information about nominal or target values for different stages of usage of an IoT device. Those stages are presented as follows:

1) Integration Stage (Initial Setup): This stage of an IoT device represents the initial integration into an existing environment by recording the boot process of the IoT device in detail. Reference values should be specified by the manufacturer. These values are to be expected during the initial boot process. Examples are CPU usage, energy consumption, standard boot time, successful boot loader verification, etc. Having reference points would help detect tampered IoT devices from the beginning. Comparing the original values (manufacturer's specifications) with the actual values when a device is first set up allows the detection of anomalies. This approach can not only be applied during initial integration into an IoT environment, but it can also be utilized in the day-to-day monitoring efforts of IoT devices during operation. Threshold values could also be defined and specified by the manufacturer which are not exceeded during everyday use.

2) Operating Stage: This stage should reflect the IoT device metric in operation mode. It should list the same parameters as mentioned in the integration stage but with adjusted values. Additional values when operating in a production environment could be listed. An example might be the data throughput (amount of processed data). Furthermore, the IoT runtime facts in operation enable responsible parties to identify malicious IoT devices by monitoring the given reference values with the current information when in use. This allows for the detection of misconfiguration or of tampered devices without having to shut down an entire MIoT infrastructure as a precaution. To meet the above requirements, the IoT Device IdentificAtion and RecoGnition (IoTAG) [25] approach might present a possible solution. The IoTAG approach proposes that all IoT devices used in an IoT environment report their security-relevant parameters, such as a unique ID, a device name, the current software version, active services, etc., in order to manage IoT networks securely [25].

3) Fail Safe Stage: Within this stage, extreme values for security-relevant parameters need to be defined by the manufacturer. Those extreme values (maxima and minima) should never be exceeded in any operation stage of a MIoT device. Should this still happen, a reaction chain must be invoked, and the MIoT device has to automatically be removed from the operating stage and be forced to pause operation.

B. Best Before Inspection Date

To support a more transparent labeling and thereby strengthen the role of consumers, an additional important indicator is suggested: the best-before-inspection date. This date is not a fixed value as known from food safety. Instead, it is intended as an indication for decision-makers. It represents how long the IoT device can securely operate, at least without the need to apply changes. The date can be extended by updates, patches, etc. The best-before-inspection date for MIoT proposed depends largely on the activities and reaction time in terms of further development by the manufacturers. Parameters, which influence the best-before-inspection date, are, among others, the following:

- Update cycle
- How many new product variants were newly developed by the manufacturer?
- What is the average end of lifetime period for this particular manufacturer?
- etc.

Many more parameters could be mentioned to modify the best-before-inspection date. The mentioned parameters are used to get the idea across. The approach to calculating an accurate best-before-inspection date is quite difficult, as no average values regarding the lifetime of individual hardware components are available. This is amplified by the fact that the lifetime is also dependent on its operating time and operating environment. If average values were available for the lifetime of individual components considering the actual operating time and operating environment, it would be possible to calculate the best-before date of hardware. Results could be based on the component with the shortest life time. It should also be noted that a fixed best-before date could negatively impact our ecological environment, as IoT devices would be disposed of when the best-before date is exceeded. Reevaluating whether the IoT device can still be used for its intended purpose from a technical point of view might not be done. Hence, there is a need to develop a more flexible best-before-inspection date. The best-before-inspection date is intended to specify a point in time when it becomes necessary to reevaluate the IoT device for the first time after its initial integration. Otherwise, IoT turns into an avoidable risk. With this definition in mind, it is more comprehensible to calculate an accurate best-before-inspection date. The calculation starts with the date of manufacturing or, if not available, with the date of purchase.

After a starting point is declared, the best-before-inspection date can vary based on certain parameters. Parameters that have a positive effect could be the frequency of updates, if the product or software is still purchasable or if the product line still exists. Parameters with a negative effect could be that the manufacturer does not provide support or updates anymore. This kind of behavior of a manufacturer would automatically lead to a negative label. Labeling a product in such a way provides decision-makers an indicator that the manufacturer does not provide continuous and recurring updates. This might influence the decision of whether buying an IoT or MIoT is beneficial. The precise definition of parameters that can be used to classify values as positive or negative in relation to the best-before-inspection date will be identified in future research efforts.

The approach of determining an approximate best-before-inspection date enables IT security managers to initiate various actions. The best-before-inspection date is primarily intended to initiate an action on a specific day. An action can be a comprehensive screening of the IoT device by checking if the firmware is up to date. Restarting the IoT device and then comparing the measured values during the reboot process with the original ones provided by the manufacturer is also possible. Another option is ensuring that actual support activities offered by the manufacturer are still active. Furthermore, the best-before-inspection date can also be used to start a new threat modeling or the maturity modeling process. The latter is suggested in Section IV.

Ultimately, it can be said that the three proposed labeling approaches have the potential to provide two benefits. On the one hand, the decision-making power of end consumers is increased. On the other hand, decision makers in critical businesses, for example, hospitals, can be strengthened as well. Above all, the MIoT components facts label contributes to greater transparency and thus increases trust in MIoT devices, the manufacturers and the technologies themselves. To achieve a labeling system for technology and to encourage manufacturers to participate, the government is in charge of establishing incentives and / or regulations, as it can be observed in the food industry. In the following section, two specific MIoT devices have been examined. The focus of the examination is on security in order to emphasize the relevance of the previous suggestions.

VI. Security Testing of Ultrasonic Scanning Equipment

In our previous work [1] we concluded that common security guidelines for Medical IoT devices are needed to build the resilience necessary to provide a safe and secure environment for patients in the long term. But is a need for such guidelines and recommendations warranted? To answer this question, we monitored the connections of two mobile ultrasonic scanners. Analyzing only two MIoT devices does not allow drawing conclusions on how security is handled in the MIoT industry as a whole. However, as it is already laid out in [1] other entities did take a look at a larger number of devices and deduced that many MIoT devices lack basic security features. This section is meant to see if those results are still relevant for up-to-date products currently sold on the market.

Both scanners require smartphones for operation. On each smartphone, the respective app needs to be installed. These apps allow connecting to the scanner and provide additional functionalities such as

- saving previous scans,
- creating patient records,
- live video conferencing whilst sharing the image of the ultrasonic scanner,
- synchronizing patient records with the cloud of the manufacturer or
- sharing patient records through the cloud.

The scanners are multi-purpose ultrasonic imaging systems. They allow the examination of different organs of the human body, such as the abdomen, bladder, lung or prostate. One product uses a WiFi connection, while the other requires a wires USB Type-C connection to communicate with the corresponding smartphone app.

A. The Security Test Setup

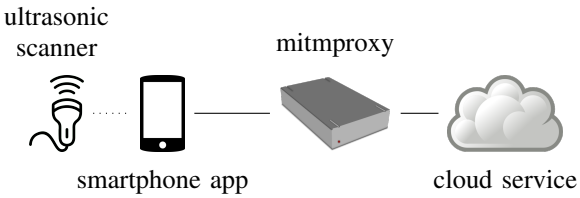


Figure 2. Abstract structure how app traffic is intercepted

The apps of both scanners require the user to log in with an account at the respective cloud service. A connection to the cloud services is mandatory to use the products. Monitoring the connection between the smartphone and the cloud service is therefore of interest in order to gain information about how connections are being handled and what type of information is being sent.

The smartphone used for testing was a rooted Android device. For intercepting the traffic between the smartphone and the cloud services, the software mitmproxy [26] has been set up. The traffic can be decrypted on the fly by installing the root certificate generated by mitmproxy on the smartphone as a system-level certificate. The traffic of the scanner app is being redirected towards the proxy by leveraging the firewall rules of the smartphone. A simplified structure of how the traffic is intercepted can be seen in Figure 2. To ensure that any outgoing connections can be attributed correctly, only traffic from and towards the respective app is being redirected to the mitmproxy software.

B. Pitfalls and Limitation During Testing

Both tested devices and apps presented certain challenges when trying to intercept their communications. In the following section, these will be presented to give an understanding

of the limitations of the security tests within the scope of this specific work.

1) Communication Interfaces: Each scanner uses a different way to establish a connection with the smartphone. One scanner opens up a WiFi access point. The respective smartphone needs to connect to that. The other scanner uses a wired USB Type-C connection to exchange data with the smartphone. Both ways put certain restrictions on the means of how the connections can be monitored.

The wireless scanner reserves the WiFi connection for the data exchange with the smartphone. A simultaneous connection with the cloud services is therefore only possible by using the mobile broadband connection of the smartphone. Hence, it was tried to build a reverse tethering connection between the smartphone and the device where the mitmproxy software is running [27]. This approach came with its own issues. Since the software used generates the reverse tethering connection by tunneling all network traffic through a Virtual Private Network (VPN) client towards the proxy device, all network communication is forced to go through this proxy computer. As a result, the app cannot communicate with the scanner while the reverse tethering connection is active. No signals can be exchanged even if the wireless device is connected to the smartphone via WiFi.

Due to these constraints, it was only possible to evaluate the connections made by the app itself. Connections made while using the wireless scanner were not the subject of the evaluation.

2) Root Detection: The scanner, which uses the wired connection, allows for a simultaneous connection of the smartphone with the computer where the mitmproxy software is running in order to intercept the traffic. However, this device puts mechanisms in place to detect if the app is running on a rooted smartphone. If the app detects that the smartphone has root access enabled, it then simply refuses to start. Extra steps had to be taken to trick the scanner app into accepting the rooted environment. After hiding the root privileges, the app successfully started. Still getting past the app's login screen was not possible, even with all these modifications in place. This only allowed inspecting connections made right after first starting the app, as well as authentication attempts made when trying to use login credentials. Rooting the device was required to install the TLS certificate of mitmproxy as a system-level certificate, which allows the inspection of TLS-encrypted traffic. Decrypting the TLS traffic on a non-rooted device was not possible.

C. Results of the Traffic Monitoring

All connections captured were secured using Transport Layer Security (TLS) 1.2 or higher. No plaintext communication between the manufacturers' apps and cloud services was discovered.

The smartphone app of the wireless scanner only connected to two different URLs:

- https://cloud.-manufacturer-.com and
- https://*.amazonaws.com.

Taking a look at the second domain reveals that the corresponding IP address belongs to the Amazon Web Services (AWS) platform. The entire cloud service for the wireless scanning system is therefore hosted on servers belonging to the company Amazon. The IP addresses can be assigned to the city of Montreal, Quebec, Canada, according to the service IP2Location [28].

On the wired ultrasonic scanning system, significantly more communication activity can be seen. After first starting the app, connection attempts to the following URLs can be observed:

- https://firebase-settings.crashlytics.com
- https://firebaseinstallations.googleapis.com
- https://crashlyticsreports-pa.googleapis.com
- https://clientstream.launchdarkly.com
- https://mobile.launchdarkly.com
- https://firehose.us-east-1.amazonaws.com
- https://cdn-settings.segment.com

The first three URLs listed belong to the Firebase product provided by the company Google. The connections captured infer that the Firebase cloud service is mainly used to process application-related logging events, such as crash reports. A report sent to Firebase contains additional meta data besides the error message created by the application. The metadata consists, among other things, of the build number of the app, the smartphone model, the smartphone fingerprint, the built-in chipset, the language of the operating system, the manufacturer name and the operating system version.

The next two URLs belong to Launchdarkly. Launchdarkly is a feature management platform for mobile app development. It allows the developer to enable or disable certain features through an online portal without needing to redeploy or update the application. The app is told by the Launchdarkly server, whose features are supposed to be enabled or displayed. A regular synchronization mechanism between smartphone and server is therefore leveraged. It should be mentioned that similar metadata to what is sent to the Google Firebase service is sent to the Launchdarkly servers. The IP addresses of Launchdarkly suggest that their servers are located in the US and are part of the AWS infrastructure.

Amazon Kinesis Data Firehose is a platform by Amazon that allows data streams of high volumes and from many sources to be saved and processed within the Amazon infrastructure. This is most likely the service used to store all user information, such as previous scans or patient data. The data streams sent and received during the tests could not be decoded. Therefore, it was not possible to reconstruct what kind of data was actually sent. The servers are located in the US and are part of AWS's infrastructure.

The last service monitored during testing was Segment. Segment is a service dedicated to giving the app developer the ability to collect user analytics data. The focus is on tracking user and device behavior to optimize the user experience. While Firebase and Launchdarkly both collect some user information, the number of parameters sent was far less than what Segment is transferring to their servers. The additional

information is, for example, the screen resolution of the smartphone, active wireless connections, mobile carrier information or the timezone the user is in. Again, the location of the Segment servers is in the US, and they belong to the AWS infrastructure.

D. Implications of the Monitoring Results on Security and Privacy

The security of both ultrasonic scanners can not be sufficiently evaluated to make a qualified statement about their resilience against an attacker. This is either due to technical constraints or obstacles put in place by the developer to restrict tampering with or evaluating the software used with the scanner. It can be said that all connections observed were using at least TLS 1.2 or higher to encrypt the traffic between the apps and the cloud services. This ensures a sufficient level of confidentiality when transferring data from one endpoint to another.

However, in terms of privacy, bigger issues become apparent. In both cases, third parties save and process metadata and patient data directly. None of the vendors tested were hosting their own cloud solution. Instead, both manufacturers decided to use the AWS solution, which is apparently hosted in Canada and the US. The wired ultrasonic scanning system shares information with four different companies, which are not part of the manufacturer or vendor. All data observed was secured via TLS but was not end-to-end encrypted. That means all information stored in the cloud servers is stored in plain text. Patient data is also saved in plain text. This was verified by creating a dummy patient record in the app of the wireless scanning system and monitoring the connection activity. It is worth mentioning that cloud synchronization of patient data is an optional feature of the device.

Storing sensitive patient data in plain text on cloud services can be an issue. Both manufacturers state on their website that they comply with the European GDPR. However, the data is being stored on foreign servers in plain text. Therefore, the data could be accessed by foreign entities or agencies. Additionally, every actor with access to the cloud storage could read and manipulate any data they want. In both cases, the manufacturer of the ultrasonic scanner, as well as Amazon, have access to the medical data provided to them by their customers. This can be a problem if customers want to ensure a high level of privacy for their patients. The customer must trust the vendor or service provider to handle the information given with absolute discretion. Misuse of the data stored can not be prevented on a technical level. It is up to the service provider to adhere to the contractual agreements. Furthermore, the service provider needs to ensure that their infrastructure has state-of-the-art IT defense mechanisms in place to prevent cyber attacks. In a worst-case scenario, a security breach at a vendor could lead to a data leak of all customers.

But patient data is not the only information transmitted to third parties. In Section VI-C it was shown that additional metadata is being sent to Google, Launchdarkly and Segment. These companies are able to see what equipment was used at a certain time in a specific location. They might even be able to retrace usage statistics of employees handling these devices. So, not only is information about patients given to third parties, but it is also plausible to argue that employees' privacy using these scanners might be compromised.

In conclusion, the customer needs to trust that all parties linked to these ultrasonic scanners handle the data given to them responsibly. No technical precautions have been put in place to prevent misuse of the given data. Given the sensitivity of the handled data, better security mechanisms can be expected from the manufacturers in question.

VII. Outlook

As presented in this work, it is applicable that many efforts will be spent on future security topics, e.g., architectures and processes, starting with best practices for manufacturers. Trustworthy security, safety and trust begin not by signing contracts, e.g., Service Level Agreements (SLA). The trust root starts long before. Politicians and official authorities should consider the derived proposed labeling concepts from the food industry. Of course, those labeling concepts require further research and broad consensus among manufacturers and global technology consortiums. But as we all know, it is possible to agree on labeling and enclosed concepts that provide more transparency for consumers and additionally strengthen safety, security and trust. The National Institute of Standards and Technology (NIST) is already discussing labeling IoT products targeting mostly security-related aims [29].

Research already provides different processes, methods and models that can be used to realize a more secure, safe and trustworthy technological evolution; for example, the process described by Roots of Trust (RoT) [3] is a promising and practical way to achieve absolute trust in a hyper-stakeholder environment targeting manufacturers from the first breath up to the retailers. Bringing the roots of trust into action requires a non-editable approach to audit and trace. A promising technological fundament would be distributed ledger technology to fulfill the required needs. Actual Blockchain-enhanced RoT solutions are already discussed [30], [31]. Another well-promising enhanced version of the "roots of trust" is the "hardware roots of trust" to validate and ensure trust in hardware components. Also, this approach is being researched by Javaid et al. [32]. With the mentioned RoT processes, it would also be relatively easy to accurately define a best-before-inspection date, which can be, for now, only roughly estimated.

Unquestionably, all the above-mentioned suggestions are worth further exploration to foster security, safety and trust in the IoT domain. This paper should not only summarize already existing efforts. Instead, it is intended to present a (Medical) IoT labeling approach and a new paradigm that seems worth focusing on.

VIII. Conclusion

In this paper, we pointed out that while advisories, guidelines and certain regulations for common IT networks exist, the Medical IoT sector still severely lacks these documents and frameworks. This has the potential to become a severe issue in the future since the amount of IoT devices in operation is rapidly growing, and the data processed is very sensitive information which requires robust protection mechanisms.

To provide guidance for stakeholders and authorities, we proposed an evaluation system to help actors within the healthcare sector. This methodology aims to identify the current Medical IoT security posture. Additionally, this maturity model can be used to understand which steps are necessary to bring the IT security of the infrastructure in question to the next level.

Furthermore, a labeling system for Medical IoT devices was proposed. With such a system, stakeholders should be able to get an overview of the key facts and components of a MIoT system to ascertain the risks and benefits it provides. With that information, decision-makers can manage risk more reliable and faster. However, such a system needs to be standardized. It is the responsibility of governments and regulatory bodies to define the rules for creating such a label to guarantee the sufficiency of the values included and ease of use for the stakeholders.

Finally, two Medical IoT devices were subjected to a basic security test. This test showed that for the connections of the IoT devices to their respective cloud services, sufficient security mechanisms had been put in place. However, in terms of privacy and confidentiality of patient data, it is not clear to the consumer or stakeholder what parties are involved to provide the services. Since the security posture between different parties can vary significantly, it is misleading to think that the security of the IoT device only depends on the vendor or manufacturer.

That is why it is essential for stakeholders to have the tools available to assess the security of their networks and to have a concise overview of the components and parties involved in providing a Medical IoT service.

IX. Acknowledgment

This research is funded as part of recently granted 5G4Healthcare project by the German Federal Ministry for Digital and Transport within the 5x5G Initiative.

References

[1] M. Gleißner, J. Dotzler, J. Hartig, A. Aßmuth, C. Bulitta, and S. Hamm, "It security of cloud services and iot devices in healthcare," in *CLOUD COMPUTING 2021*, B. Duncan, Y. W. Lee, and M. Popescu, Eds. Wilmington, DE, USA: IARIA, 2021, pp. 1–7.

[2] S. Dahmen-Lhuissier, "Etsi - technologies - 5G," 2022. [Online]. Available: https://www.etsi.org/technologies/5g?tmpl=component&jjj=1642854003450 [retrieved: 2022.12.15].

[3] National Institute of Standards and Technology, "roots of trust," 2022. [Online]. Available: https://csrc.nist.gov/glossary/term/roots_of_trust[retrieved:2022.12.15] [retrieved: 2022.04.11].

[4] ITU Radiocommunication Sector, "Minimum requirements related totechnical performance for IMT-2020 radio interface(s)," ITU Radiocommunication Sector, Nov. 2017. [Online]. Available: https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2410-2017-PDF-E.pdf [retrieved: 2022.12.15].

[5] Bundesministerium für Bildung und Forschung, "Industrie 4.0 [industry 4.0]," Jan. 2016. [Online]. Available: https://www.bmbf.de/bmbf/de/forschung/digitale-wirtschaft-und-gesellschaft/industrie-4-0/industrie-4-0.html [retrieved: 2022.12.15].

[6] Harvard University, "Made in china 2025 explained," 2022. [Online]. Available: https://projects.iq.harvard.edu/innovation/made-china-2025-explained [retrieved: 2022.12.15].

[7] E. Y. Nakagawa, P. O. Antonino, F. Schnicke, R. Capilla, T. Kuhn, and P. Liggesmeyer, "Industry 4.0 reference architectures: State of the art and future trends," *Computers & Industrial Engineering*, vol. 156, p. 107241, 2021.

[8] B. Mazon-Olivo and A. Pan, "Internet of things: State-of-the-art, computing paradigms and reference architectures," *IEEE Latin America Transactions*, vol. 20, no. 1, pp. 49–63, 2022.

[9] R. Heidel, M. Hoffmeister, M. Hankel, and U. Döbrich, Eds., *Industrie 4.0: The reference architecture model RAMI 4.0 and the Industrie 4.0 component*, ser. Beuth Innovation. Berlin and Wien and Zürich and Berlin and Offenbach: Beuth Verlag and VDE Verlag, 2019.

[10] Y. Lu, K. C. Morris, and S. Frechette, "Current standards landscape for smart manufacturing systems," Feb. 2016.

[11] Juxtology, "Iot: Architecture." [Online]. Available: https://www.m2mology.com/iot-transformation/iot-world-forum/ [retrieved: 2022.12.15].

[12] IoT European Large-Scale Pilots Programme, "Create-IoT." [Online]. Available: https://european-iot-pilots.eu/create-iot/consortium/ [retrieved: 2022.12.15].

[13] National Institute of Standards and Technology, "Roots of trust." [Online]. Available: https://csrc.nist.gov/Projects/Hardware-Roots-of-Trust [retrieved: 2022.12.15].

[14] IoT Security Foundation, "Device secure boot." [Online]. Available: https://www.iotsecurityfoundation.org/best-practice-guide-articles/device-secure-boot/ [retrieved: 2022.12.15].

[15] Bundesamt für Sicherheit in der Informationstechnik, "SYS.4.3 Eingebettete Systeme [SYS.4.3 Embedded Systems]," 2021. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium_Einzel_PDFs_2021/07_SYS_IT_Systeme/SYS_4_3_Eingebettete_Systeme_Edition_2021.pdf [retrieved: 2022.12.15].

[16] Bundesamt für Sicherheit in der Informationstechnik, "Sys.4.4: Allgemeines iot-gerät," 2021. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium_Einzel_PDFs_2021/07_SYS_IT_Systeme/SYS_4_4_Allgemeines_IoT_Geraet_Edition_2021.pdf [retrieved: 2022.12.15].

[17] European Union, "Regulation (eu) 2016/679 of the european parliament and of the council," 2016. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN#d1e3265-1-1 [retrieved: 2022.12.15].

[18] Public Health, "Factsheet - european health data space (ehds)," 2022. [Online]. Available: https://ec.europa.eu/health/latest-updates/factsheet-european-health-data-space-ehds-2022-05-03_en [retrieved: 2022.12.15].

[19] European Union Agency for Cybersecurity, "Baseline security recommendations for IoT," Nov. 2017, [retrieved: 2022.12.15].

[20] BioSpace, "IoT in healthcare market to reach USD 260.75 billion by 2027– reports and data," Jul. 2021. [Online]. Available: https://www.biospace.com/article/iot-in-healthcare-market-to-reach-usd-260-75-billion-by-2027-reports-and-data/ [retrieved: 2022.12.15].

[21] EFQM, "Efqm_model_2020_v2_german_summary: 2. überarbeitete ausgabe," 2021. [Online]. Available: https://mailchi.mp/b505ea74b885/61o67uatr8 [retrieved: 2022.12.15].

[22] Brookfield, "Mapping of iot security recommendations, guidance and standards to the UK's code of practice for consumer IoT security," Oct. 2018. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/973928/Mapping_of_IoT__Security_Recommendations_Guidance_and_Standards_to_CoP_Oct_2018_V2.pdf [retrieved: 2022.12.15].

[23] Avedis Donabedian, "Evaluation the quality of medical care," p. 166–206, 1965. [Online]. Available: https://www.jstor.org/stable/3348969

[24] European Union, "EUR-Lex - 32018r0775 - EN," May 2018. [Online]. Available: https://eur-lex.europa.eu/legal-content/DE/TXT/?qid=1568299917869&uri=CELEX:32018R0775 [retrieved: 2022.12.15].

[25] L. Hinterberger, S. Fischer, B. Weber, K. Neubauer, and R. Hackenberg, "Iot device identification and recognition (iotag)," in *CLOUD COMPUTING 2020, The Eleventh International Conference on Cloud Computing, GRIDs, and Virtualization*, 2020, pp. 17–23.

[26] A. Cortesi, M. Hils, and T. Kriechbaumer, "mitmproxy - an interactive HTTPS proxy," 2022. [Online]. Available: https://mitmproxy.org/ [retrieved: 2022.12.15].

[27] R. Vimont, J. Shuali, S. Testa, and A. Aslanyan, "Gnirehtet (v2.5)," 2022. [Online]. Available: https://github.com/Genymobile/gnirehtet [retrieved: 2022.12.15].

[28] IP2Location, "IP address to IP location and proxy information | IP2Location," 2022. [Online]. Available: https://www.ip2location.com/ [retrieved: 2022.12.15].

[29] National Institute of Standards and Technology, "IoT product criteria," Feb. 2022. [Online]. Available: https://www.nist.gov/itl/executive-order-improving-nations-cybersecurity/iot-product-criteria [retrieved: 2022.12.15].

[30] *2018 International Conference on Smart Communications and Networking (SmartNets)*. IEEE, 2018.

[31] Elisa Bertino, Dan Lin, and Jorge Lobo, Eds., *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies*, ser. ACM Conferences. New York, NY: Association for Computing Machinery, 2018. [Online]. Available: http://dl.acm.org/citation.cfm?id=3205977

[32] U. Javaid, M. N. Aman, and B. Sikdar, "Defining trust in IoT environments via distributed remote attestation using blockchain," in *Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. New York, NY, USA: ACM, 2020.

APPENDIX

TABLE II. Criteria dimension structure

Maturity / Criteria	Organisation	Data Management	Authentication System
1	• Health facility's management commitment to the implementation of IT security for all IoT devices in the whole healthcare area	• Detailed description of end-to-end-security and cryptographic principles • Utilize crypto coprocessors for key creation and storage • All products related to web servers have their HTTP trace and trace methods disabled	• No default credentials for Medical IoT devices • No use of any function by unauthorized user or guest users incl. patients (also changing credentials) • Applications operated at the lowest privilege level possible
2	• Definition of basic objectives, scope, roles and tasks regarding IT security of Medical IoT devices • Determining contractual clauses with Medical IoT suppliers about IT security	• Encrypted data on application layer • All communications keys are stored with industry standards (e.g., FIPS 140) • All communication ports (e.g., USB, RS232) only communicate with authorized and authenticated entities • Minimized sharing principle of resources	• Different secret keys for each Medical IoT or product family • Complex password management (no blanks, no containing user account name, etc.) for all Medical IoT devices
3	• Definition of all Medical IoT processes including risk level • All products contain a unique and tamper-resistent device identifier (e.g., chip serial number)	• Key management incl. generation, distribution, storage and maintenance • Utilize trusted platform modules (TPM) and hardware security modules (HSM) • Communication protocols are at most secure version (e.g., Bluetooth 4.2 rather than 4.0)	• No hard coded passwords in IoT software code • 2-Factor authentication for all Medical IoT devices
4	• Training for medical staff about IT security of IoT devices • All OS non-essential services have been removed from product's software	• Storage of sensitive data in hardware (not software) • Only use secure boot methods	• Multi factor authentication or certificates for all Medical IoT devices • Secure mechanism for updated credentials (fixed time intervals) for all medical stuff
5	• Manufacturers consider compliance with ISO 30111 for vulnerability report handling	• Encrypt data parameters using a Direct Acess Recovery (DAR) encryption key stored I a physically locked module • Using Root of Trust (certificates, signing keys)	• No secret credentials left in application code of Medical IoT devices • Biometric authentication for all medical staff

TABLE III. Criteria dimension process

Maturity / Criteria	Updates	Malfunction Management	Usability
1	• Regular updates of security measures of all Medical IoT devices • User notification when updates and patches modify user-configured preferences, security and privacy settings	• Defined use of error handlers • Generic error messages and use of custom error pages • Enable restore secure state after security breach • Runtime Protection mechanism	• Basic training for medical staff is done
2	• Agile and prompt response to new security or other flaws of IT in the health facility • Validation of authenticity and integrity of all updates (e.g., signing certificate) • Restore secure state (if update was not successful or occurred)	• Log all authentication attempts and failures of the medical staff • Log all access control failures of the medical staff • Log all apparent tampering events	• Advanced training for staff is done
3	• Automated update process for all Medical IoT devices • Use of libraries that are actively maintained and supported	• Defined bug reporting system from Medical IoT suppliers • Log all backend TLS connection failures • Automated alerting system for tampering events	• Regular training sessions incl. innovations are taught for medical staff
4	• Defined limitation of device functionality for all Medical IoT devices after security support period ends (e.g., remote control) • Backward compatibility of updates (compatible with previous versions) for all Medical IoT devices	• Mechanisms for self-diagnosis and self-repair for all Medical IoT devices	• No training necessary for usage or all medical staff is trained for usage
5	• Updates include cryptographic checks and cipher suites • Complete end-to-life update strategy for all Medical IoT devices incl. awareness of potential risks beyond its expected expiry date	• Participation in information sharing platform to report vulnerabilities and current cyber threats of Medical IoT devices	• No training necessary for usage or all medical staff is trained for usage incl. IT security handling

TABLE IV. Criteria dimension outcome

Maturity / Criteria	Costs for IT Security	Downtime reg. criticality	Failsafe	Threats and attacks
1	• Less than 1 % of the complete health facility budget	• All Medical IoT devices with low criticality have a max. downtime of 3 days	• Failure affects the whole system / the whole Medical IoT device / the whole IoT product family	• there were less than 25 security-related events (e.g., threats, attacks) last year in the health facility
2	• Less than 2 % of the complete health facility budget	• All Medical IoT devices with low criticality have a max. downtime of 24 hours	• Failure affects parts of the system / the whole Medical IoT device / the whole IoT product family	• there were less than 20 security-related events (e.g., threats, attacks) last year in the health facility
3	• Less than 4 % of the complete health facility budget	• All Medical IoT devices with medium criticality have a max. downtime of 3 days	• Failure affects the availability of operation or use of the system / the whole Medical IoT device / the whole IoT product family	• there were less than 15 security-related events (e.g., threats, attacks) last year in the health facility
4	• Less than 6 % of the complete health facility budget	• All Medical IoT devices with medium criticality have a max. downtime of 24 hours	• Failure has no effect on the medical operation (no human damage possible)	• there were less than 10 security-related events (e.g., threats, attacks) last year in the health facility
5	• More than 8 % of the complete health facility budget	• All Medical IoT devices with high criticality have a max. downtime of 24 hours	• Failure has no effect on the operation or use of the whole system / the whole Medical IoT device / the whole IoT product family	• there were less than 5 security-related events (e.g., threats, attacks) last year in the health facility

Microservices Authentication and Authorization from a German Insurances Perspective

Arne Koschel
Andreas Hausotter
Hochschule Hannover
University of Applied Sciences & Arts
Faculty IV, Department of Computer Science
Hannover, Germany
Email: {arne.koschel, andreas.hausotter}
@hs-hannover.de

Pascal Niemann
Christin Schulze
Hochschule Hannover
University of Applied Sciences & Arts
Faculty IV, Department of Computer Science
Hannover, Germany
Email: {pascal.niemann, christin.schulze}
@stud.hs-hannover.de

***Abstract*—Even for the more traditional insurance industry, the *Microservices Architecture (MSA)* style plays an increasingly important role in provisioning insurance services. However, insurance businesses must operate legacy applications, enterprise software, and service-based applications in parallel for a more extended transition period. The ultimate goal of our ongoing research is to design a microservice reference architecture in cooperation with our industry partners from the insurance domain that provides an approach for the integration of applications from different architecture paradigms. In Germany, individual insurance services are classified as part of the critical infrastructure. Therefore, German insurance companies must comply with the Federal Office for Information Security requirements, which the Federal Supervisory Authority enforces. Additionally, insurance companies must comply with relevant laws, regulations, and standards as part of the compliance requirements. Note: As Germany is considered relatively strict with respect to the privacy and security demands, meeting these requirements may well be suitable (if not even "over-fulfilling") for insurance companies in other countries. The question raises thus of how insurance services can be secured in an application landscape shaped by the MSA style to comply with the architectural and security requirements depicted above. This article highlights the specific regulations, laws, and standards the insurance industry must comply with. We present conceptual approaches for authentication and authorization in a MSA tailored to the requirements of our insurance industry partners. In particular, we focus on different architectural patterns for service-level authorization as well as approaches for service-level authentication and discuss their advantages and disadvantages.**

Keywords*—*Security; Authorization; Authentication; Insurance Industry; Microservices Architecture.

I. Introduction

In this article, which is an extended version of our previous work [1], we look at *Information Technology (IT)* security within a microservices-based reference architecture for at least German insurance companies. IT security is absolutely a "must-have" for insurance companies, especially for customer data, self-written and third-party applications, and their IT infrastructure in general. General regulations, such as the European *General Data Protection Regulation (GDPR)* [2], are applied to the insurance domain, as well as insurance-specific laws and rules regarding security and other regulations [3] [4], for example, data protection and secured IT communication infrastructure. This article mainly focuses on securing insurance business applications [5]. Over time, several technologies from monolithic mainframe applications, functional decomposition-based software, traditional *Service-Oriented Architecture (SOA)*, and third-party enterprise software, such as SAP systems, were and are used together in insurance business applications.

Recently, the *Microservices Architecture (MSA)* style [6] [7] and cloud computing joined the field. The ultimate goal of our currently ongoing research [8] is to develop a "*Microservice Reference Architecture for Insurance Companies (RaMicsV)*" jointly with partner companies from the insurance domain, which is taking all those typical cornerstones from (overtime grown) insurances into account. Placed within our work on RaMicsV is the question: "how to help secure (insurance) business applications using potentially several logical parts from RaMicsV, mainly including microservices combined with other typical insurance applications technologies"?

Only a few authors (see Section II) look at such technology combinations, and they especially do not take (German) insurance domain specifics into account. Thus, the present article constitutes an initial step in that direction.

In particular, we contribute here our ongoing work and initial results regarding:

- An introduction to IT security regulations in Germany for insurance companies, including:
  - A brief explanation of when an institution is considered critical infrastructure and the resulting consequences.
  - Functions and regulations of the *Federal Office of Information Security (BSI)* and the *Federal Financial Supervisory (BaFin)* in this context.
- Evaluate existing patterns for achieving protection goals and weigh their advantages and disadvantages.

- Take a look at properties of service- and edge-level authentication and authorization.
- An overview of approaches to service-level authentication.
- Consider patterns concerning the requirements of the insurance industry with SOA and an *Enterprise Service Bus (ESB)*.

Especially our Sections V, VI, and VII include several new illustrations and are altogether more detailed than our previous work from [1].

The remainder of this article is structured as follows: After looking at related work in Section II, we place our current work into our initial logical reference architecture from [8] in Section III. Next, Section IV looks at requirements for German insurance companies. Initial work to meet those requirements is contributed in Section V, which discusses edge- vs. service-level authorization and authentication, in Section VI, which examines authorization patterns, and in Section VII, which details authentication patterns. Both parts are evaluated concerning their potential application within our overall work. Finally, Section VIII summarizes our results, draws a conclusion, and looks at future work.

II. Related Work

Our research is based on the literature of well-known authors in microservices, especially Chris Richardson (Microservices Pattern) [6]. His book describes fundamental statements for the advantages and disadvantages of the edge-level security pattern and the service-level security pattern.

We adopted our definition of components for authorization and authentication from the *National Institute of Standards and Technology (NIST)* [9] and the patterns described in Sections V and VI originate from [10]. Furthermore, [10] discusses service-level authentication through mutual transport layer security and a token-based approach, whereas Section VII also briefly discusses this and adds Hypertext Transfer Protocol and Password Authenticated Key Exchange to this topic. In contrast, this paper uses the above content to a certain extent and places it in the context of our reference architecture and the legal scope of our partners in a German insurance company.

Kai Jander et al. compare general transport layer security, transport layer security with service and microservice frameworks for authentication and encryption of microservices. They provide an overview of password authentication, symmetric keys and key pairs, and then present an implementation of a password authenticated key exchange [11]. In contrast, this paper uses a different scope and establishes a connection to legal regulations for German insurance companies.

Regarding legal regulations and specifications, we use, among others, *the Act on Federal Office for Information Security (BSIG)* [12]. Especially the part for critical infrastructures and, accordingly, *the Regulation for the Determination of Critical Infrastructures according to the BSI Act (BSI-KritisV)* [13] is used to underpin the relevance of our reference architecture. In addition, this is supplemented with *the Insurance Regulatory Requirements for IT (VAIT)* [3] published by the BaFin, as this is the responsible authority of the insurance industry.

In our previous work [8], we presented the logical microservices reference architecture that we created in the German insurance domain with our partners by logical and technical details in the area of logging and monitoring components. So far, components in the area of security have not been considered within this reference architecture, which is now started in the present article.

Additionally, in [14], we dealt with the consistency of microservices, among other things. Here, compliance aspects were described, which arose during the service design using Domain Driven Design. The requirements specific to German insurance companies were briefly mentioned. Based on this, the legal constraints and controlling constitutions are described in more detail.

To the authors' knowledge, this is the first work to address the legal regulations for German insurance companies in the context of a reference architecture for microservices with a focus on patterns for security and, in particular, authentication and authorization. In addition, we address the requirement of this reference architecture for microservices to work together or side by side with an ESB (see Section III).

III. Reference Architecture for Insurance Companies

This section will present our logical reference architecture for microservices in the insurance industry (RaMicsV).

RaMicsV defines the setting for the architecture and the design of a microservices-based application for our industry partners. The application's architecture is out of scope, as it heavily depends on the specific functional requirements.

When designing RaMicsV, a wide range of restrictions and requirements given by the insurance company's IT management have to be taken into account. Concerning this contribution, the most relevant are:

- ESB: The ESB, as part of the SOA, must not be questioned. It is part of a successfully operated SOA landscape, which seems suitable for our industry partners for several years to come. Thus, from their perspective, the MSA style is only suitable as an additional enhancement and only a partial replacement of parts from their SOA or other self-developed applications.
- Coexistence: Legacy applications, SOA, and microservices-based applications will be operated in parallel for quite an extended transition period (several years to come). This means that RaMicsV has to provide approaches for integrating applications from different architectural paradigms – looking at it from a high-level perspective, allowing an "MSA style

best-of-breed" approach at the enterprise architectural level as well.

Figure 1 depicts the building blocks of RaMicsV, which comprises layers, components, interfaces, and communication relationships. Components of the reference architecture are colored yellow; those out of scope are greyed out.

A component may be assigned to one of the following *responsibility areas*:

- **Presentation** includes components for connecting clients and external applications such as SOA services.
- **Business Logic & Data** contains the set of microservices to provide the desired application-specific behavior.
- **Governance** consists of components that contribute to meeting the IT governance requirements of our industrial partners.
- **Integration** contains system components to integrate microservices-based applications into the industrial partner's application landscape.
- **Operations** consist of system components to realize unified monitoring and logging, which encloses all systems of the application landscape.
- **Security** consists of components to provide the goals of information security, i.e., confidentiality, integrity, availability, privacy, authenticity & trustworthiness, non-repudiation, accountability, and audibility.

Components communicate via HTTP—using a RESTful API, or message-based—using a *Message-Oriented Middleware (MOM)* or the ESB. The ESB is part of the *integration* responsibility area, which contains a message broker (see Figure 1).

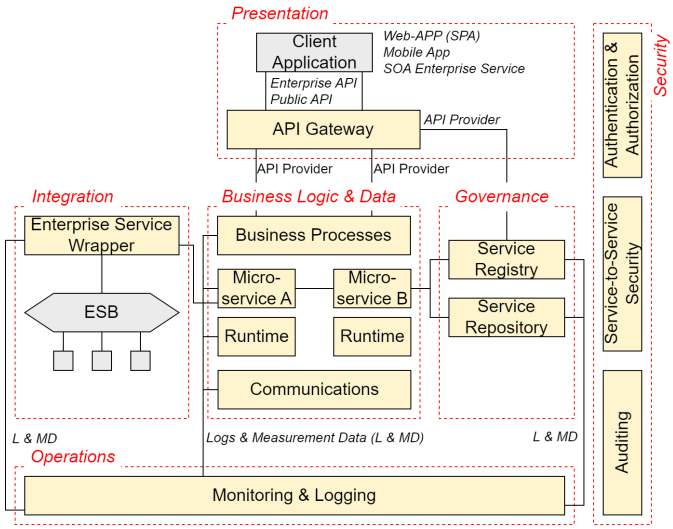


Fig. 1. Building Blocks of the Logical Reference Architecture RaMicsV

In addition to data transformation and message routing and delivery, an ESB also implements security policies. For example, WS02 ESB supports *Web Services (WS)*-Security and WS-Policy specifications [15]. Beyond that, the WSO2 Identity Server can be used to generate an *OAuth Base Security Token* that microservices may employ to authenticate and authorize client applications and API clients. This corresponds to the edge- level authentication & authorization depicted in Section V.

In the next sections, we will look at the *security* responsibility area.

IV. Requirements for German Insurance Companies

Security is a fundamental aspect of any software architecture and should never be neglected, mainly when there is a legislative framework where specific regulations exist. In Germany, insurance companies, regarded as critical infrastructure, are obligated to comply with the requirements of the BSIG, which the BaFin enforces. The Federal Office for Information Security has determined this consideration. Note: In our work, we did not look at regulations and legal requirements in other countries, but, as stated above, German regulations are seen as "somewhat tough" already.

A. Federal Office for Information Security and Critical Infrastructures

The BSI is a federal agency in Germany responsible for security standards inside federal authorities and is a central reporting point for security incidents. Companies that are running critical infrastructures are obligated to report to the BSI. The Council of the European Union defined that a critical infrastructure "... is essential for the maintenance of vital societal functions, health, safety, security, economic or social well-being of people, and the disruption or destruction of which would have a significant impact in a Member State ..." [16]. Therefore an ordinance (BSI-KritisV [13]) from 2016 defines critical infrastructures in Germany. It could easily have dramatic consequences for the economy, state, and society if an infrastructure from one of the seven mentioned sectors (energy, water, food, information technology and telecommunications, health, finance and insurance, transport, and traffic) were attacked. Under Section 7 (1) no. 1 to 5, examples are given of critical financial and insurance services, which are of corresponding importance. Some examples mentioned are payment transactions or, among other things, insurance services and social security benefits. However, either a system or a part of it must be assigned to column B (System category) of Annex 6 Part 3 and, at the same time, exceed the corresponding threshold value in column D of the specific metric to be considered critical infrastructure. A general example would be a contract administration system in which the number of life insurance claims per year exceeds 500,000. Therefore, some of our partners' systems are considered critical infrastructure and are liable to other requirements.

Because of the BSIG from 2009 [12], under Section 8a, "Security regarding the information technology of critical infrastructures," institutions with critical infrastructures are obligated to a security standard. They need to provide evidence to the BSI every two years that they took precautionary

measures to achieve the protective goals of IT security. Specifically mentioned are **availability, integrity, authenticity, and confidentiality**. In addition, precautions are described here as reasonable if the effort required to secure the protection goals is in proportion to the consequences of the failure. Moreover, the BSI has published a document [17] that specifies the requirements imposed by Section 8a (1) BSIG.

Section 8a (2) of the BSIG states that it is possible to establish an industry-specific security standard that meets the requirements. The Federal Office of Civil Protection and Disaster Assistance and the corresponding regulatory authority will determine whether this standard is appropriate. Thus, there has to be a Federal Office that determines whether the company is complying with the requirements.

B. Federal Financial Supervisory Authority

The BaFin is responsible for the supervision of banks and financial and insurance providers. They published VAIT [3] in the year 2018. This publication contains the general conditions and specifications for IT risk and security management. There is a reference to the BSI-KritisV, which has an entire section dedicated to critical infrastructures. All aspects are essential, from detection over definition to implementation of security measurements. The goal is to secure the protective objectives of IT security, which are named in Subsection IV-A, and to minimize all risk factors inside the critical infrastructure. Therefore, German insurance companies must provide evidence through audits, certificates, or examinations every two years to fulfill their obligations. That is why every aspect of security needs to be addressed while or even better before implementing new systems.

C. Further Motivation for the Commitment to Confidentiality

There is a wide range of security aspects that need to be addressed. At this point, we would like to refer to a document published by the BSI entitled ”Supervision of critical infrastructures in finance and insurance” [4]. This briefly discusses the legal requirements for critical infrastructures and the introduction of these requirements in 2019. The document states that most of the deficiencies and shortcomings did not pose a direct threat to maintaining the operation of the infrastructures concerned. Nevertheless, according to ISO/IEC 27002, eight percent of the deficiencies were attributable to access control.

Additionally, in 2021 the *Open Web Application Security Project (OWASP)* Top Ten 2021, first place is ”Broken Access Control,” and seventh place is ”Identification and Authentication Failures” [18]. Compared to 2017, ”Broken access control“ came up from place 5 [19]. This shows that the importance of authorization and authentication continues to increase. As a result, it is increasingly important to find mechanisms that protect system boundaries with a low potential for error by business logic development teams.

Concerning Subsections IV-A and IV-B, the four security properties that are explicitly named are listed below:

- **Confidentiality** includes read access by authorized subjects only.
- **Integrity** describes writing access by authorized subjects only.
- **Availability** implies access by authorized subjects at any time.
- **Authenticity** verifies the identity of the sender.

Through conversations with our partners, the focus of this paper will first be on different patterns of the service-level authorization aspects as part of the confidentiality and partly integrity protection goal. Since authorization can be close to authentication in terms of implementation, it will also be included in the following section concerning the implementation location.

V. Authorization and Authentication - Edge- vs. Service-level

In distributed systems, authentication and authorization can be realized at different locations. While there is typically one place where authentication and authorization are performed in monolithic systems, there are various system locations where authentication and authorization might occur in distributed systems. This section describes the fundamental differences in properties when using authorization or authentication for microservices depending on the implementation location.

Authentication and authorization have a crucial difference in the choice of location. As seen in Figure 2, the distinction between the two locations is easy to see. Already recognizable from the name, the service-level is located on the microservices level. On the other hand, the edge-level is the boundary to the outside, represented by the API Gateway.

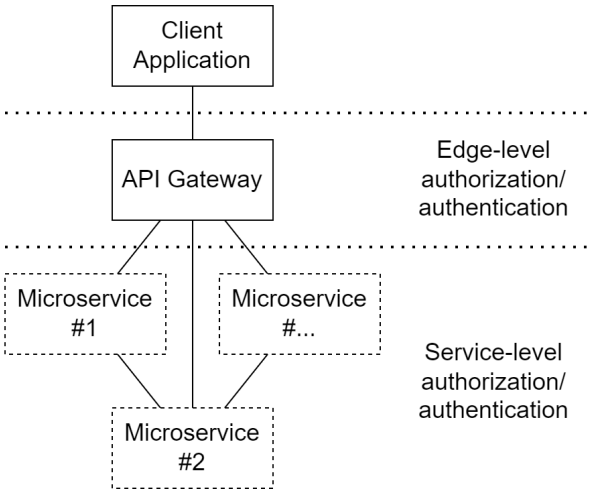


Fig. 2. Visual abstraction of a part of RaMicsV to represent the location of edge- and service-level.

Scalability is the critical factor in positioning authentication, as there is no business reason to prefer edge-level or service-level. Authentication needs a database to check credentials and calculate any security token; domain knowledge is unnecessary [6]. In the case of authorization, on the other hand, it

is not only scalability that is important but also how access is controlled. If *Role-based Access Control (RBAC)* is the only requirement, decisions can be made without domain knowledge, e.g., by roles per URL path. In this case, edge-level authorization is usable. An *Access Control List (ACL)* is called when more explicit authorization is required. In this case, domain information is needed, and service-level authorization is suitable.

This section does not discuss technical authentication and authorization solutions but highlights the authentication and authorization positioning and the resulting properties for the system's performance and development. For both authentication and authorization, two fundamentally different approaches are possible. At the edge-level, the required components are frequently located in an API Gateway, whereas at the service-level, the components are located in each service. In the following section, we first discuss edge-level authentication.

A. Edge-level Authentication

If there is an API Gateway, it may be used for authentication decisions. This is a quick-to-develop but hard-to-scale solution. Using an API Gateway has the following properties [6]:

- Domain logic development teams are very little involved with authentication.
- API Gateway development teams have to deal with more complexity.
- Only one team is responsible for the authentication. This lowers the risk of security vulnerability.
- Faster development by lower complexity.
- Poor scalability due to a single point of control.
- Risk of too strong coupling of API Gateway and microservices; independent deployment is usually impossible.

B. Service-level Authentication

An alternative to the API Gateway implementation is authentication at the service-level. This solution is slow and expensive to develop but scales well. The service-level authentication has the following properties [6]:

- Domain logic development teams have to deal with more complexity.
- Higher risk for security vulnerabilities due to multiple development teams.
- Slower development due to higher complexity in any microservice.
- Higher scalability, which stresses one of the essential properties of an MSA.
- If RBAC is used and only one role exists for a specific microservice, e.g., only the admin, authentication failures play less of a role for this microservice because regular users are not allowed to access it anyway.

The difference between authentication at the edge- and service-level should have become clearer now: Both approaches provide the authentication basis for the protection goals of confidentiality and integrity, which are described in Section IV. There are different strategies to deal with service-level authentication. These will be mentioned in Section VII. In the next section, edge-level and service-level authorization will be presented.

C. Edge-level Authorization

With edge-level authorization, all the logic resides in the API Gateway. This brings the following characteristics:

- Easy implementation and maintenance.
- It may create problems when scaling.
- Designing complex systems can be challenging.
- Back-end microservices must only be accessible via the API Gateway.
- Risk of too strong coupling of API Gateway and microservices—no independent deployment is possible.

This is a suitable solution for a lightweight MSA with few roles. Next, we will look at service-level authorization, which is increasingly attractive for more complex systems [10].

D. Service-level Authorization

Like authentication, authorization can also be implemented at the service-level. An additional component is added to each microservice for authorization, authentication, or both. In this context, the following terms are important (Figure 3) [9]:

- ***Policy Enforcement Point (PEP)*** enforces the authorization decision.
- ***Policy Decision Point (PDP)*** computes the authorization decision.
- ***Policy Administration Point (PAP)*** comprises an interface to administrate the policies.
- ***Policy Information Point (PIP)*** provides additional information for the PDP to make authorization decisions [9].

As shown in Figure 3, the PEP and PDP form the authorization functionality.

The subsequent patterns are determined by the localization of the PEP and PDP in relation to a microservice. PAP and PIP are only mentioned for completeness. At first, we consider the general properties change compared to edge-level:

- Responsibility moves from the API development team to the microservices development team.
- Complex microservices environments are possible.
- Implementation and maintenance are more complex because changes affect each microservice.

Overall, this sets out the fundamentals. In the following section, different patterns regarding service-level authorization are presented. These take an essential role in architectural decisions regarding the use of microservices.

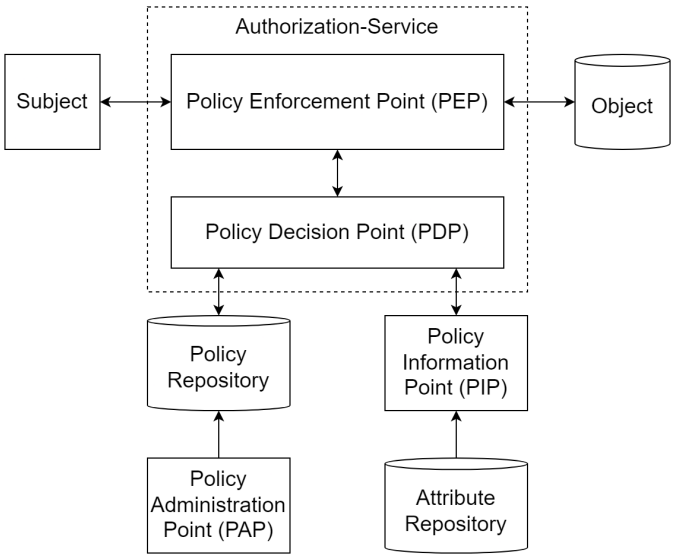


Fig. 3. Fundamental points of ACM [9].

VI. Service-level Authorization - Patterns

There are three different patterns of service-level authorization: Decentralized pattern, centralized pattern with a single PDP and centralized pattern with embedded PDP. Each pattern offers different advantages and disadvantages and differs architecturally. In the following, the three patterns are fundamentally described and architecturally visualized. A brief theoretical evaluation of the properties in the context of our reference architecture RaMicsV will be provided in order to simplify decision-making depending on the context.

A. Decentralized pattern

The decentralized pattern is the solution to create a microservice that is wholly controlled by the development team. All software and data components for making authorization decisions reside inside the microservice. A visualization of this can be seen in Figure 4. This is optimal for scaling, but it requires much effort to implement and maintain since any change in the authorization process requires changes in each microservice. Another challenge is propagating policy or attribute changes to all microservices. The challenge just mentioned becomes even more complex and grows linearly with the increasing number of microservices. This must also take into account that microservices can fail at this point and not receive the information. Thus, ensuring that the information is passed on has a high priority. On the other hand, there are scenarios where this pattern may be suitable, e.g., if there is a microservice with a high number of requests [10]. Furthermore, it has the advantage that no additional functionality needs to be provided by the ESB within RaMicsV since all functionality is contained within the respective microservices themselves.

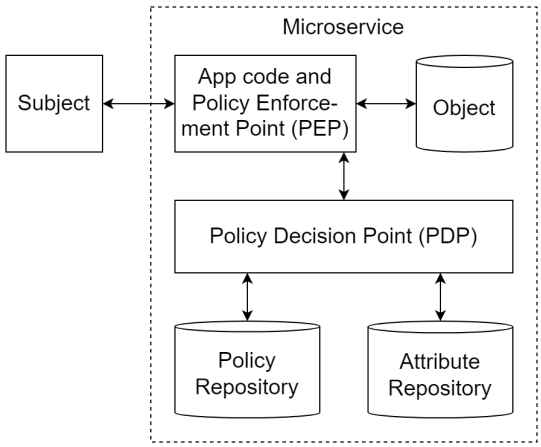


Fig. 4. Service-level Authorization - Decentralized pattern [10].

B. Centralized pattern with single PDP

With the centralized single PDP pattern, the PEP is located within each microservice, and the PDP resides in a different central location, as shown in Figure 5. This implies that every request to the microservice will result in a network call to the PDP. Thus, this is not a suitable solution if a very low response time is required. Also, if high scalability is needed, a single decision point is associated with limitations. In addition, updating policies and attributes is unproblematic due to relocation. Accordingly, the just mentioned things can be updated detached from the microservices.

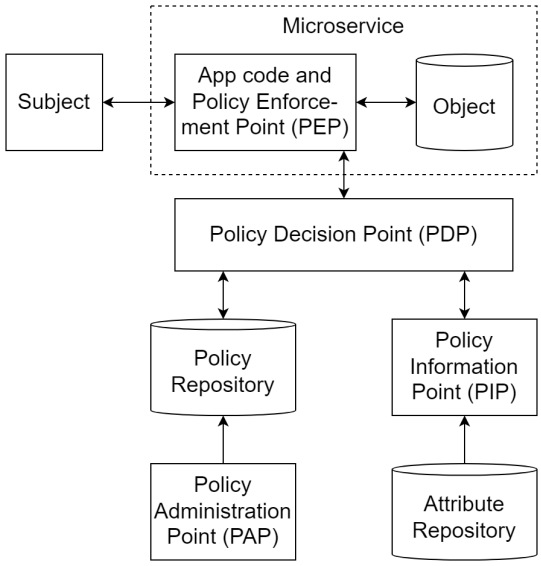


Fig. 5. Service-level Authorization - Centralized pattern with single PDP [10].

However, in the case of a central PDP, all microservices are independent of changes within the PDP. It should be taken into account that possible failures of the latter can have critical consequences. Moreover, this approach could be faster to be implemented in cooperation with a required

ESB (see Section III), because then, the PEP resides in each microservice, and the PDP is provided by the ESB [10].

C. Centralized pattern with embedded PDP

In the centralized pattern with embedded PDP, the data and attributes are centralized, but the PDP is part of each microservice. This is why Figures 5 and 6 are almost identical, since only the content of the microservice changes, and in this case, the PDP is within that service. Unlike the decentralized pattern (see Subsection VI-A), the PDP is not part of the code but is embedded using a microservices library. So, the PDP is part of the microservice for quick decisions, but the development team doesn't have a lot of development work. Similar to the centralized pattern with a single PDP (see Subsection VI-B), unlike the decentralized pattern (see Subsection VI-A), there is no difficulty in propagating policy or attribute changes to all microservices.

For interoperation with the required ESB (see Section III), this pattern combines the advantages of a decentralized pattern and a quick implementation. The ESB could be used for data and attribute sharing. All other components could make fast decisions through the microservices [10]. Concerning the protection goals described in Section IV, the authorization enforces confidentiality and integrity.

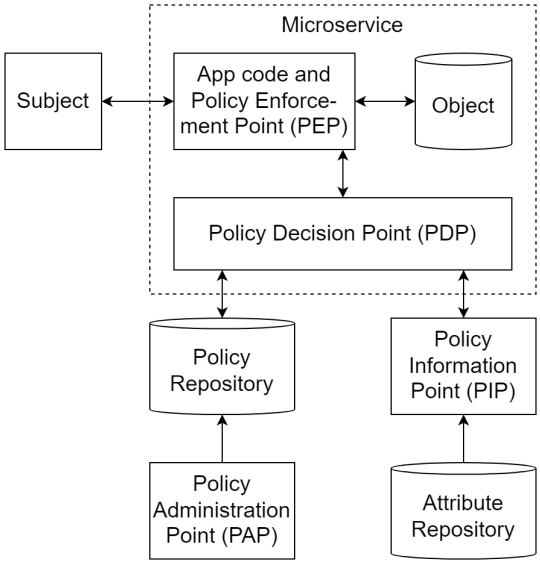


Fig. 6. Service-level Authorization - Centralized pattern with embedded PDP [10].

D. Summary

Insurance companies are running large and complex systems with many different services and fine-grained access control. For this reason, edge-level authorization is suitable only in specific scenarios, for example, if RBAC can be used for a given microservice.

The application landscape of our partners in the insurance industry comprises an ESB as part of the reference architecture (see Section III). Therefore, each pattern has its use case, as we explained above. The decentralized pattern (see Subsection VI-A) is recommended when performance is the most crucial requirement. The centralized pattern with a single PDP (see Subsection VI-B) is suitable if performance is less critical and RBAC is needed. In addition, there is also the possibility that the PDP could be integrated directly within the ESB. The centralized pattern with embedded PDP (see Subsection VI-C) brings together the advantages of the previously mentioned patterns and is, therefore, from our point of view, the most promising one.

Nevertheless, even if service-level authorization can be ensured, the counterpart of authentication still needs to be addressed. Accordingly, within the following section, different approaches of service-level authentication within our context will be described and considered.

VII. Service-level Authentication - Approaches

As in the previous section, this part is based on a theoretical consideration of possible approaches to service-level authentication in the context of the BSI specifications and our partners' reference architecture. With regard to the reference architecture, the aim is to keep the overhead as low as possible by using an appropriate service-level authentication. Accordingly, it does not deal with technical implementations or the precise flow of protocols, but references for more in-depth information are given where appropriate. This is to give a rough overview. The most critical points concerning authentication, which are at the authors' discretion, are highlighted. The sequence of approaches builds on each other in specific parts and tries to solve a posed problem of the previously mentioned approach. In its simplest form, the handling of authentification can be such that it is not taken into account. Instead, essential trust is established within the system. Since this is questionable under consideration of different security aspects to carry out, different approaches follow now.

A. Hypertext Transfer Protocol

Within the *Hypertext Transfer Protocol (HTTP)*, there are two ways of authentication. One is the "Basic Authentication" and the other is the "Digest Access Authentication". In the first variant, authentication takes place using credentials, and the message is encoded using Base64. In this case, anyone can read the message exchange. In the second variant, a challenge is also introduced, in which a nonce and a checksum verified at the end ensure that both parties know the secret [20], [21]. With this type of authentication and communication, sending messages in plain text is particularly critical. As also described in one of the publications of the BSI with regard to requirements to be implemented for critical communication paths in Chapter 2 in the section of technical information security number 33, encryption and authentication must be provided when transmitting data of a critical service [17]. Therefore, it is obligatory to disregard this type of authentication due to

the non-existent encryption and consider its advanced security variant of it.

B. Hypertext Transfer Protocol Secure

The *Hyper Text Transfer Protocol Secure (HTTPS)* consists of the implementation of HTTP using *Secure Sockets Layer (SSL)* or *Transport Layer Security (TLS)*. According to the minimum standard of the BSI for the use of Transport Layer Security according to §8 paragraph 1 sentence 1 BSIG version 2.3 dated 15.03.2022, the recommendations of the technical guideline TR-02102-2 Cryptographic Procedures dated 24.01.2022 [22] must be complied with. Following this technical guideline, TLS1.0, TLS1.1, SSL v2, and SSLv3 are not recommended. Instead, only TLS 1.2 and TLS 1.3 are recommended. Barabanov and Makrushin list six different sources that declare mTLS as the most popular variant of authentication for microservices [10]. As can be seen in Figure 7, the use of HTTPS requires a certificate authority, which is an anchor of trust. Basically, none, one, or both communication partners can authenticate each other [23]. The latter is referred to as *mutual TLS (mTLS)*. The issuing and verification of certificates take place within the certificate authority. But the problem of key management must be overcome. Both recalls and rotations must be made possible. The options to manage this are not mentioned, but for the sake of completeness, however, this aspect has been included. In addition, the use of a PKI results in further obligations with regard to measures to be implemented towards the BSI in the case of using a critical infrastructure. Apart from the problems of key management also mentioned above, PKI generates a certain overhead, which is attempted to be avoided within the reference architecture in this context. However, no additional PKI should possibly be implemented. Therefore, it would also be possible to use *JSON web tokens (JWT)* for authentication.

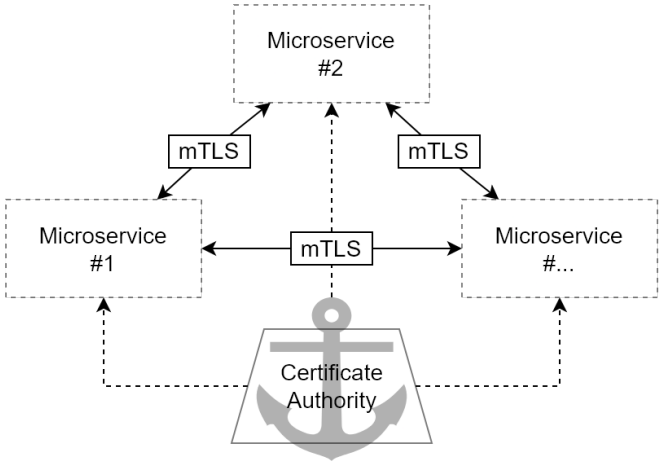


Fig. 7. Authentication with HTTPS.

C. JSON Web Tokens

A wide variety of information can be stored within JWT [24]. Authentication can therefore be performed using context/user information in particular. If the microservices create and sign JWT themselves, a *Public Key Infrastructure (PKI)* is also required. Otherwise, a *Security Token Service (STS)* is introduced, which can issue new tokens and act as an intermediary between two microservices. This process is illustrated in Figure 8. In this case, when two microservices communicate, a new JWT is used each time. Access controls can be performed at the STS. *JSON Web Encryption (JWE)* [25] should be used as the format of the tokens at this point in order to meet the BSI's requirements for encrypting data in critical infrastructure [17]. As with HTTPS, other things have to be taken care of. For example, the detection of recalled or compromised tokens. Basically, JWT can also be used in combination with mTLS, whereby certain security aspects can be handled in each case. Basically, as with HTTPS, the problem of increasing overhead also arises here. There is another option for service-level authentication without the additional need for another infrastructure, such as PKI or STS. In that case, the *Password Authenticated Key Exchange by Juggling (J-PAKE)* protocol is promising.

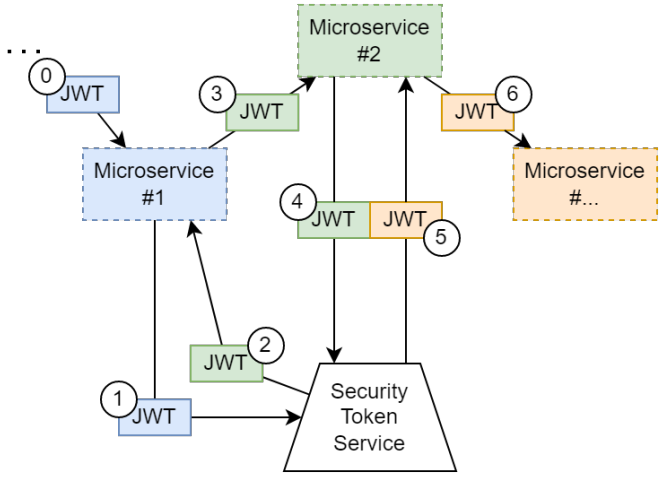


Fig. 8. Authentication with JWT.

D. Password Authenticated Key Exchange by Juggling

The J-PAKE protocol [26] belongs to the family of password authenticated key exchange protocols. This protocol is used to create a cryptographic key based on a shared secret, which is used for further secure communication. No PKI or third party is additionally needed. It also covers other security features. For example, *Perfect Forward Secrecy (PFS)* is also required by the BSI as a property when using TLS [22]. PFS describes the property of the keys so that when a key is known, previous and still later following keys cannot be determined [27]. For the sake of completeness, other security features are also only mentioned but can be traced in some detail within the RFC [26]: Offline and online dictionary attack resistance and known-key security.

However, since this protocol works on the basis of a common shared secret, one difficulty is how and when the services'

secrets are created and propagated accordingly. A more in-depth description of an implementation of the J-Pake protocol in the context of an MSA with an example of using the Jadex Active Components framework was given by Kai Jander et al. [11]. A possible problem of this protocol within an MSA is the level of awareness and the number of implementations, as they do not seem to be very widespread in this case. Especially in the enterprise area of an insurance company. Nevertheless, according to the RFC, J-PAKE has been used in applications such as Firefox sync, Pale moon sync, and Google Nest products [26]. In addition, some protocols of this family also have patents, which makes it difficult to use them, and therefore extra new protocols have been developed [28]. The potential advantage of service-level authentication without additional infrastructure and the fact that it fulfills certain requirements of the BSI in the area of German insurances should be enough to encourage further research in this area.

E. Summary

There are several ways to deal with the authentication of service-level communication. Especially in the context of German insurance or the special requirements of the BSI, it should not be assumed that one's own system will never be compromised. Therefore, there should also be no basic trust in the network, and the communication should be designed accordingly and secure. For service-level authentication, it would make sense to combine both JWT and mTLS so that, for example, authentication can be guaranteed by JWT and further context information can be sent. However, at the same time, the message itself can also be encrypted by mTLS. Within a large insurance company, a multitude of services exist, and the existing system landscape has reached a certain level of complexity. Therefore, the use of mTLS and JWT creates a larger overhead since both a PKI and an STS have to be maintained at the same time. In addition, there are also other problem areas, such as key management. A suitable but, at the same time, theoretical solution to the problem would be the use of the J-PAKE protocol. This protocol enables authentication and the additional creation of a secure communication channel without the need for a PKI or other additional infrastructure. Therefore, the use of J-PAKE seems promising and is recommended by the authors. Nevertheless, more in-depth research, especially on the practical application and also on compliance with regulatory requirements, is needed when using the protocol in an MSA with critical infrastructure within a German insurance company. In addition, there are questions regarding the practical implementation and safeguarding of governance, which should not be neglected. Finally, this approach should continue to be considered until further commercial uses develop.

VIII. Conclusion and Future Work

The security aspect is indispensable in any realization or evolution of application architecture. Especially in Germany, insurance companies have to fulfill legal requirements according to the BSIG if general framework conditions are met, and the resulting status of critical infrastructure is achieved. Every two years, proof must be provided to the BSI that the corresponding security standard is met. The BaFin is responsible for the regulation of this proof. Our partners from the insurance industry, thus, should still be compliant with those requirements if adding a critical (defined based on BSI-KritisV) system part based on RaMicsV.

For better guidance on authorization patterns from a confidentiality perspective, authentication has also been included, as the two security properties are usually close in terms of implementation. Relevant points regarding the implementation at the service-level and edge-level have been included. The paper's main focus was on the different patterns of service-level authorization, which were considered and evaluated in the context of our partners within the insurance industry. Furthermore, approaches to service-level authentication were also described, and their challenges were fundamentally addressed.

Finally, the advantages and disadvantages of the individual patterns were weighed up. The pattern of choice depends on the requirements for scalability and performance. In the context of (grown) insurance and microservices, implementation at the service-level seems the most appropriate. Furthermore, the centralized pattern with the single or the embedded policy decision point comes in closer selection due to the use of the required ESB within RaMicsV. Approaches to service-level authentication were also described, and their challenges were fundamentally addressed. While mTLS seems to be the common standard, PAKE protocols are promising and may take on a larger role in microservices in the future than they have in the past. Thus, an important part of the protection goal confidentiality was addressed. Still, it also took another step closer to answering the initially asked question: "how to help secure (insurance) business applications using potentially several logical parts from RaMicsV, mainly including microservices combined with other typical insurance applications technologies"?

Within this contribution, some guidelines for selecting patterns regarding authorization and authentication at service- and edge-level of critical infrastructure have been started and will be continued within our future work. In addition, our future work also deals with the approach of validity and consistency of embedded policies. To continue to remain oriented towards the protection goals, a prominent topic, service-to-service authentication, will be addressed in more detail in future work as well. In particular, the practical implementation of the theoretical approaches will be considered. It will also look at how the industry standard deals with this. Here, the available options for implementing authentication will be considered inside RaMicsV, and the respective advantages and disadvantages will be weighed against each other. In particular, the highlighted and recommended protocol J-PAKE will also be considered in more depth. Both in terms of legal regulations to be complied with in the context of critical infrastructures of a German insurance company, as well as problems regarding a practical implementation based on RaMicsV.

Furthermore, relevant and current aspects of the broad subject's availability and integrity will then be evaluated one by one to address later emerging security aspects of the MSA, such as deployment options and resulting security domains. The exact order is made in consultation with our partners from the insurance industry, depending on current topics or preferences.

Initial prototypes and proof of concepts have been developed and implemented for the reference architecture and were described in previous publications [8] and [14]. While similar work has not yet been done for the security domain from this publication, the effort required to implement parts or all of the reference architecture in a commercial system depends on the existing SOA, specific functional requirements, and the number of critical systems components to be implemented.

References

[1] A. Koschel, A. Hausotter, R. Buchta, P. Niemann, C. Schulze, C. Rust, and A. Grunewald, "The Need of Security Inside a Microservices Architecture in the Insurance Industry," in *Proceedings of the Fourteenth International Conference on Advanced Service Computing (Service Computation 2022)*, Barcelona, Spain, 2022, [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=service_computation_2022_1_20_10006. [accessed: 2022-12-15].

[2] The European Parliament and the Council of the European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation - GDPR))," [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng. [accessed: 2022-12-15].

[3] Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin) - Federal Financial Supervisory (BaFin), "Versicherungsaufsichtliche Anforderungen an die IT (VAIT) (Insurance Supervisory Requirements for IT (VAIT))," [Online]. Available: https://www.bafin.de/SharedDocs/Downloads/EN/Rundschreiben/dl_rs_1810_vait_va_en.pdf?__blob=publicationFile&v=5. [accessed: 2022-12-15].

[4] Bundesamt für Sicherheit in der Informationstechnik (BSI) - Federal Office of Information Security (BSI), "Aufsicht über Kritische Infrastrukturen im Finanz- und Versicherungswesen (Supervision of Critical Infrastructures in the Finance and Insurance Industry)," [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/KRITIS/Nachweispruefungen_im_Finanz-und_Versicherungswesen.pdf?__blob=publicationFile&v=3. [accessed: 2022-12-15].

[5] Gesamtverband der Deutschen Versicherungswirtschaft e.V. (General Association o.t. German Insurance Industry)), "VAA Final Edition. Das Fachliche Komponentenmodell (VAA Final Edition. The Functional Component Model)," 2001.

[6] C. Richardson, *Microservices Patterns: With examples in Java*. Shelter Island, New York: Manning Publications, 2018.

[7] S. Newman, *Building microservices: designing fine-grained systems*. Sebastopol, California: O'Reilly Media, Inc., 2015.

[8] A. Koschel, A. Hausotter, R. Buchta, A. Grunewald, M. Lange, and P. Niemann, "Towards a Microservice Reference Architecture for Insurance Companies," in *Proceedings of the Thirteenth International Conference on Advanced Service Computing (Service Computation 2021)*, Porto, Portugal, 2021, [Online]. Available: https://www.thinkmind.org/index.php?view=article&articleid=service_computation_2021_1_20_10002. [accessed: 2022-12-15].

[9] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to attribute based access control (abac) definition and considerations (draft)," *NIST special publication*, vol. 800, no. 162, pp. 1–54, 2013.

[10] A. Barabanov and D. Makrushin, "Authentication and authorization in microservice-based systems: survey of architecture patterns," *CoRR*, vol. abs/2009.02114, 2020, [Online]. Available: https://arxiv.org/abs/2009.02114. [accessed: 2022-12-15].

[11] K. Jander, L. Braubach, and A. Pokahr, "Defense-in-depth and role authentication for microservice systems," *Procedia Computer Science*, vol. 130, pp. 456–463, 2018, the 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018), [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918304009. [accessed: 2022-12-15].

[12] Bundesamt für Sicherheit in der Informationstechnik (BSI) - Federal Office of Information Security (BSI), "Act on the Federal Office for Information Security (BSI Act - BSIG) - courtesys translation -," [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/BSI/BSI_Act_BSIG.pdf?__blob=publicationFile&v=4. [accessed: 2022-12-15].

[13] Bundesamt für Sicherheit in der Informationstechnik (BSI) - Federal Office of Information Security (BSI), "Verordnung zur Bestimmung Kritischer Infrastrukturen nach dem BSI-Gesetz (BSI-Kritisverordnung - BSI-KritisV) (Regulation for the Determination of Critical Infrastructures according to the BSI Act (BSI-Kritisverordnung - BSI-KritisV))," [Online]. Available: https://www.gesetze-im-internet.de/bsi-kritisv/BJNR095800016.html. [accessed: 2022-12-15].

[14] A. Koschel, A. Hausotter, M. Lange, and S. Gottwald, "Keep it in Sync! Consistency Approaches for Microservices - An Insurance Case Study," in *SERVICE COMPUTATION 2020, The Twelfth International Conference on Advanced Service Computing*, Nice, France, 2020, [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=service_computation_2020_1_20_10016. [accessed: 2022-12-15].

[15] WSO2, "WS02 Enterprise Service Bus Documentation: Securing APIs," [Online]. Available: https://docs.wso2.com/display/ESB481/Securing+APIs. [accessed: 2022-12-15].

[16] The Council of the European Union, "COUNCIL DIRECTIVE 2008/114/EC," [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32008L0114. [accessed: 2022-12-15].

[17] Bundesamt für Sicherheit in der Informationstechnik (BSI) - Federal Office of Information Security (BSI), "Konkretisierung der Anforderungen an die gemäß § 8a Absatz 1 BSIG umzusetzenden Maßnahmen (Specification of the requirements for the measures to be implemented in accordance with Section 8a (1) BSIG)," [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/KRITIS/Konkretisierung_Anforderungen_Massnahmen_KRITIS.pdf. [accessed: 2022-12-15].

[18] CWE Content Team, "Weaknesses in OWASP Top Ten (2021)," 2022, [Online]. Available: https://cwe.mitre.org/data/definitions/1344.html. [accessed: 2022-12-15].

[19] OWASP, "Welcome to the OWASP Top 10 - 2021," 2022, [Online]. Available: https://owasp.org/Top10/. [accessed: 2022-12-15].

[20] J. Reschke, "The 'basic' http authentication scheme," Internet Requests for Comments, RFC Editor, RFC 7617, September 2015.

[21] J. Franks, P. M. Hallam-Baker, J. L. Hostetler, S. D. Lawrence, P. J. Leach, A. Luotonen, and L. C. Stewart, "Http authentication: Basic and digest access authentication," Internet Requests for Comments, RFC Editor, RFC 2617, June 1999.

[22] Bundesamt für Sicherheit in der Informationstechnik (Federal Office for Information Security)), "Technische Richtlinie TR-02102-2 Kryptographische Verfahren: Empfehlungen und Schlüssellängen (Technical Guideline TR-02102-2 Cryptographic methods: Recommendations and key lengths)," [Online]. Available: https://verdata.de/wp-content/uploads/2021/10/BSI-TR-02102-2.pdf. [accessed: 2022-12-15].

[23] E. Rescorla, "Http over tls," Internet Requests for Comments, RFC Editor, RFC 2818, May 2000.

[24] M. Jones, J. Bradley, and N. Sakimura, "Json web token (jwt)," Internet Requests for Comments, RFC Editor, RFC 7519, May 2015.

[25] M. Jones and J. Hildebrand, "Json web encryption (jwe)," Internet Requests for Comments, RFC Editor, RFC 7516, May 2015.

[26] F. Hao, "J-pake: Password-authenticated key exchange by juggling," Internet Requests for Comments, RFC Editor, RFC 8236, September 2017.

[27] C. Eckert, *IT-Sicherheit: Konzepte - Verfahren - Protokolle (IT-Security: Concepts - Procedures - Protocols)*. Berlin, München, Boston: De Gruyter Oldenbourg, 2014.

[28] T. Wu, "What is SRP?" [Online]. Available: http://srp.stanford.edu/whatisit.html. [accessed: 2022-12-15].

Detecting Novel Application Layer Cybervariants using Supervised Learning

Etienne van de Bijl
Centrum Wiskunde & Informatica
Amsterdam, the Netherlands
Email: evdb@cwi.nl

Jan Klein
Centrum Wiskunde & Informatica
Amsterdam, the Netherlands
Email: jan_g_klein@outlook.com

Joris Pries
Centrum Wiskunde & Informatica
Amsterdam, the Netherlands
Email: jorispries@gmail.com

Rob van der Mei
Centrum Wiskunde & Informatica
Amsterdam, the Netherlands
Email: mei@cwi.nl

Sandjai Bhulai
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
Email: s.bhulai@vu.nl

***Abstract*—Cyberdefense mechanisms such as Network Intrusion Detection Systems predominantly use *signature-based* approaches to effectively detect known malicious activities in network traffic. Unfortunately, constructing a database with signatures is very time-consuming and this approach can only find previously seen variants. Machine learning algorithms are known to be effective software tools in detecting known or unrelated novel intrusions, but if they are also able to detect unseen variants has not been studied. In this research, we study to what extent binary classification models are accurately able to detect novel variants of *application layer* targeted cyberattacks. To be more precise, we focus on detecting two types of intrusion variants, namely *(Distributed) Denial-of-Service* and *Web* attacks, targeting the *Hypertext Transfer Protocol* of a web server. We mathematically describe how two selected datasets are adjusted in three different experimental setups and the results of the classification models deployed in these setups are benchmarked using the Dutch Draw baseline method. The contributions of this research are as follows: we provide a procedure to create intrusion detection datasets combining information from the transport, network, and application layer to be directly used for machine learning purposes. We show that specific variants are successfully detected by these classification models trained to distinguish benign interactions from those of another variant. Despite this result, we demonstrate that the performances of the selected classifiers are not symmetric: the test score of a classifier trained on A and tested on B is not necessarily similar to the score of a classifier trained on B and tested on A. At last, we show that increasing the number of different variants in the training set does not necessarily lead to a higher detection rate of unseen variants. Selecting the right combination of a machine learning model with a (small) set of known intrusions included in the training data can result in a higher novel intrusion detection rate.**

***Keywords*—Cybersecurity; network intrusion detection; anomaly detection; binary classification; open-world learning.**

I. Introduction

A partial and preliminary version of this paper was presented in [1]. In our increasingly digitized world, network security has become more challenging as the Internet is used for virtually all information operations, such as storage and retrieval. The rat race between attackers and defenders is perpetual as new tools and techniques are continuously developed to attack web servers containing this information. Tremendous problems for organizations and individuals arise when legitimate users cannot access data due to cyberattacks. Modern attacks are designed to mimic legitimate user behavior and target vulnerabilities in *application-layer* protocols, such as the *Hypertext Transfer Protocol* (HTTP). This mix makes detecting them a challenging and complex task.

Defenders often use an *Intrusion Detection System* (IDS) to perform the task of detecting intrusions. An IDS can be viewed as a burglar alarm in the cybersecurity field [2]. It monitors network traffic, aims to detect malicious activities, and an alarm is triggered when this is the case. Generally speaking, the two used methodology classes by these systems are *signature-based* and *anomaly-based* [3]. A signature-based detector compares observed network events against patterns that correspond to known threats.

In contrast, anomaly-based detectors search for malicious traffic by constructing a notion of normal behavior and flags activities that do not conform to this notion. Where signature-based is time-consuming but effective, anomaly-based often suffers from a high false-positive rate. Within anomaly detection methods, *Machine Learning* (ML) algorithms are getting more attention as they might overcome this problem.

The thought of using ML algorithms to detect intrusions is not new. Various studies are performed on using ML for detecting cyberattacks. Unfortunately, there is a striking imbalance between the extensive amount of research on ML-based anomaly detection techniques for intrusion detection and the rather clear lack of operational deployments [4]. ML algorithms are highly flexible and are adaptive methods to find patterns in big stacks of data [5], but they seem better at this task than discovering meaningful outliers [4]. Modern cyberattacks often occur in large quantities and thus do not entirely conform to the premise that patterns cannot be found for these outliers. Therefore, using ML for the task of detecting these attacks should be appropriate.

There appear two issues when looking at anomaly-based ML research in intrusion detection [6][7]. Firstly, the performance of most of these methods is measured on outdated datasets [8]. This makes it hard to estimate the performance of these methods on modern network traffic. A major issue is that the composition of benign and malicious traffic in these datasets does not represent modern real-time environments.

Also, there used to be a lack of representative publicly available intrusion detection datasets, but this lack was noticed by the cyberdefense community and recently more intrusion datasets have been generated [10]. Still, the available datasets are often limited to features extracted from the transport and network layer and lack application layer features. Thus, not all attainable features are extracted in these datasets. Secondly, it is not examined how supervised learning methods perform in detecting novel variants of known attacks. The performance of these methods is measured in either a *closed-world learning setting* in which training and test classes are the same or an *open-world learning setting* with unrelated attacks. However, it is not tested how the methods perform in an open-world setting with novel variants.

Similar to our previous work [1], the aim of this paper is to study to what extent ML models are accurately able to detect novel variants of known cyberattacks. To be more precise, we use supervised binary classifiers to learn from a dataset containing benign and application layer cyberattacks and we evaluate them on their ability to detect unseen variants of these attacks. In this work, we do not only focus on one cyberattack, the *Denial-of-Service* (DoS) or its distributed form (DDoS) but include a second variant: Web attacks. We examine how the selected classifiers perform when using a single cyberattack in the training dataset on this task. Afterward, we study the effect of combining malicious variants in the training phase on the performance of classifiers detecting unseen variants. The results of this binary classification problem are benchmarked using the Dutch Draw baseline method [9]. Furthermore, we provide a procedure to transform raw network traffic data into ML-usable datasets containing information from the network, transport, and application layer. The code of this procedure is publicly available [11].

The main contributions can be summarized as follows: Firstly, we show that ML classifiers are to a great extent able to detect known cyberattacks in a closed-world setting when presented with sufficient data. Secondly, we show that there are situations where these classifiers are able to detect a novel variant when they are trained to detect a different variant. However, this is not a two-way street: learning to detect attack A and being able to also detect attack B does not imply that the reverse is the case. Thirdly, we show that training on imbalanced data has an adverse effect on the evaluation performance of some ML classifiers. We have demonstrated that variants included in the CIC-IDS-2017 seem not identical to the same variants in the CIC-IDS-2018. Finally, we demonstrate that it is not necessary to use many variants to detect a novel attack. Sometimes a few known attacks can already lead to the highest detection rate.

The organization of this paper is as follows. Section II gives a literature overview regarding detecting novel intrusions with ML. Section III describes the selected datasets and how they are modified into ML-applicable datasets and states metadata about them. In addition, a set of ML models used for conducting the experiments are given. Section IV outlines the conducted experiments. Section V shows the results of the conducted experiments. Finally, we conclude and summarize in Section VI.

II. Related Work

Detection of novel attacks with supervised learning techniques has been studied before in the context of *Transfer Learning* (TL). TL is an ML paradigm where a model trained on one task is used as a starting point for another task. [12] introduces a feature-based TL approach to find novel cyberattacks by mapping source and target datasets in an optimized feature representation. This approach is however very dependent on a similarity parameter and the dimensions of the new feature space. Therefore, [13] extended this method by proposing another approach to automatically find a relationship between the novel and known attacks. Both of these approaches are tested on an outdated dataset and it does not contain variants of a single cyberattack. In our research, we are interested in the detection of novel variants rather than novel variants. In [14], a Convolutional Neural Network is used to detect novel attacks also in a TL setup, but it is not studied if learning one specific attack affects the detection of another novel variant. The experiments conducted in our research resemble the experiments performed in [15]. In their research, an intrusion detection method is introduced that transfers knowledge between networks by combining unrelated attacks to train on. More recent work focuses on applying Deep Neural Networks in the context of TL for intrusion detection tasks [16].

III. Data

We discuss the procedure to convert raw network traffic into usable intrusion detection datasets containing information from the network, transport, and application layer for ML purposes. The converted and extracted features are described in detail so it is clear which features are included. Furthermore, we provide metadata describing the final datasets. At last, the classification models and their set of considered hyperparameters are given for detecting novel variants.

A. Data Sources

A perfect intrusion detection dataset should at least be up-to-date, correctly labeled, publicly available, contain real network traffic with all kinds of attacks and normal user behavior, and span over a long time [10]. The main reasons for a lack of appropriate datasets satisfying these properties are (1) privacy concerns regarding recording real-world network traffic and (2) labeling being very time-consuming. However, synthetic or anonymized datasets have been generated that satisfy some of these ideal properties. It is therefore recommended to test methodologies on multiple datasets instead of only one [4]. In this research, we focus on the detection of malicious variants and for that reason, we have selected the CIC-IDS-2017 [17] and the CIC-IDS-2018 [18] datasets created by the Canadian Institute for Cybersecurity (CIC). These datasets are correctly labeled, publicly available, up-to-date, and contain several malicious cyberattacks.

B. Feature Extraction

The selected datasets are provided by the CIC in two formats: a set of raw network traffic (`pcap`) files and a set of files containing extracted features by a network analysis tool called CICFlowMeter [19]. These features mainly describe network and transport protocol activities. However, there are no features describing application activities. As this study focuses on detecting application layer cyberattacks, it is desirable to have a dataset also containing application layer features. Therefore, we start with the raw internet traffic format and have selected a feature extraction tool matching this requirement.

The feature extraction tool used in this study is the open-source network traffic analyzer called Zeek (formerly Bro) [20]. Zeek is a passive standalone IDS and derives an extensive set of logs describing network activity. These logs include an exhaustive record of all sessions seen on the wire. Zeek was also used as a feature extraction tool for the creation of other popular network intrusion detection datasets, e.g., DARPA98 [21] from the Defense Advanced Research Projects Agency (DARPA) and the UNSW-NB15 [22] from the University of New South Wales (UNSW). Zeek has a good track record in creating intrusion detection datasets and therefore an appropriate tool.

By default, Zeek generates a large set of log files, but not all of them are required for this research. We limit ourselves to the *Transmission Control Protocol* (TCP) entries given in the connection logs (`conn.log`), describing network and transport layer activity, and HTTP interactions given in HTTP logs (`http.log`). These log files include entries showing malicious activities. The entries in the connection log files are transport-layer sessions, while the HTTP log file consists of entry logs showing conversations between a client and a web server. Entries between these logs are unilaterally linked as each HTTP entry is assigned to a single connection entry. Malicious activities that are not (D)DoS or Web attacks are excluded as we only focus on these attacks.

C. Feature Engineering

We describe how the extracted features are converted into ML-admissible features. This section states the additional created features, which features are replaced for better extraction of patterns, and how we smartly one-hot-encode categorical features. We start with describing the feature engineering steps in the connection log file and afterward do the same for the HTTP log file.

a) Connection log: Zeek counts the number of packets and bytes transferred in each connection. Table I shows additional created features from these counters. A higher-level statistic called the *Producer-Consumer Ratio* (PCR) [23] shows the ratio between sending and receiving packets between the hosts. In a TCP connection, an originator host is an uploader if a PCR is close to 1.0 and purely a downloader if it is close to -1.0.

The feature `conn_state` constructed by Zeek refers to the final state of a TCP connection. This state is determined by registering flags exchanged during the communication between hosts. Looking only at the end of a connection implies that the establishment and termination of the connection are merged. Preliminary results showed that classifiers were better able to find patterns in (D)DoS traffic when differentiating between the establishment of a connection and the termination of it. On this note, we replaced the `conn_state` feature with features describing both ends of a connection. The *3-Way Handshake* is the correct way to establish a TCP connection before data is allowed to be sent. This procedure is however not always correctly executed and incorrect establishments can indicate misuse. Hosts can terminate TCP connections gracefully, or not. A graceful termination occurs when both hosts send a packet with a *final* (FIN) flag. When a host sends a packet containing a *reset* (RST) flag, it will abruptly end a TCP connection, which is very common in practice. If neither is the case, connections are in theory still open. In Table II, we distinguish different establishment and termination scenarios by looking at the exchanged flags between the hosts. Each of these scenarios is included in the data as a binary feature. Other Zeek connection log flags are one-hot-encoded for both the originator and responder.

TABLE I. NETWORK LAYER ENGINEERED FEATURES.

Feature	Description	Type
orig_bpp	$\frac{\text{orig\_bytes}}{\text{orig\_packets}}$	Float
resp_bpp	$\frac{\text{resp\_bytes}}{\text{resp\_packets}}$	Float
PCR	$\frac{\text{orig\_bytes}-\text{resp\_bytes}}{\text{orig\_bytes}+\text{resp\_bytes}}$	Float

TABLE II. TCP CONNECTION ESTABLISHMENT AND TERMINATION SCENARIOS.

Feature	Description
S0	No SYN packet is observed
S1	Merely a connection attempt (SYN), but no reply
REJ1	A connection attempt but replied with a RST packet
S2	A connection attempt followed by SYN-ACK, but no final ACK
REJ2O	Scenario S2 but originator sends RST packet
REJ2R	Scenario S2 but responder sends RST packet
S3	Connection is established according to the 3-way handshake
WEIRD	A connection attempt but none of the above cases were observed
OPEN	A connection was established, but no FIN or RST flag is observed
TERM	Connection gracefully terminated by originator and receive
CLSO	Originator sends a FIN flag but receiver did not respond
CLSR	Receiver sends a FIN flag but originator did not respond
RSTO	Originator abruptly ends connection by sending an RST flag
RSTR	Receiver abruptly ends connection by sending an RST flag

b) HTTP log: Communication in this protocol starts with a client sending a request message to a web server and this server will, hopefully, reply with a response message. Both message types consist of a start-line, zero or more header fields, an empty line indicating the end of the header fields, and possibly a message body. The start-line of a request message, called the request-line, contains three components: a method (command), a path to apply this command on, and an HTTP version indicating the version a client wants to use. Hosts must agree on the HTTP version to use before they continue talking.

If they did not agree on the HTTP version, a "−1" is imputed to distinguish it from other versions.

The feature `method`, showing the command given in the request message, is a feature showing the one-word command given in the request line. Commonly used commands are 'GET', 'HEAD', 'POST', 'PUT', 'DELETE', 'CONNECT', 'OPTIONS', 'TRACE', and 'PATCH', but other commands also exist. This categorical feature is one-hot-encoded to one of those common commands to limit the number of options. In case an uncommon command is given, it will be assigned to a feature called `method_other`, while if no command is given at all, it is assigned to `method_-`.

A web server applies a `method` on the Uniform Resource Identifier(`URI`) stated in a request line. This `URI` can be parsed in different components by a library called *urllib* [24]. Figure 1 gives an example of how this tool splits a Uniform Resource Locator (*URL*) into four components. We extracted descriptive statistics from each component by counting the number of special characters (not letters or digits), the number of characters, and the number of unique characters. A typical `URI` constitutes three components: a path, a query, and a fragment. Statistics are extracted for each of those components. For example, one extracted feature called `URI_path_len` describes the length of the path of a URI. In addition, Zeek extracts `host` (only netloc), the `referrer` (all components), and these descriptive statistics are also extracted for these features.

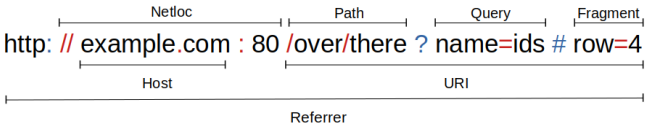


Figure 1. Example URL showing the four components parsed by urllib and the component coverage of extracted features by Zeek.

Web servers process received request messages and reply to them with a response message. In the status line of this message is the agreed HTTP version stated and a response code if the web server is able to process the request. The response codes are grouped by their first digit. So, for example, the error code 404 is assigned to the `4xx` code. Furthermore, it registered what type of data (e.g., application, audio, example, font, image, model, text, or video) is sent by the web server to the client or vice versa. This info is one-hot-encoded in a similar manner as the `method` for both directions.

D. Final Dataset

The log files are merged into a single dataset after feature engineering them. The resulting dataset consists of HTTP interactions, while in contrast, the datasets provided by the CIC consist of connection flows. Connection log features are added to the HTTP entry features to combine application, network, and transport layer features. This merge gives a dataset with a total of 103 features. The CIC-IDS-2017 consists of 533,845 instances and the CIC-IDS-2018 has 9,595,037 instances.

Table III shows the distribution of the labels of the entries. The benign/malicious ratio is roughly balanced for the CIC-IDS-2017, while it is more imbalanced for the CIC-IDS-2018. If we differentiate between cyberattacks, we observe that there is a clear imbalance between the malicious classes. For example, the *Hulk* (HTTP Unbearable Load King) attack generated a lot more HTTP entries in comparison to a *Slowloris* or *GoldenEye*. The same can be observed for Web attacks. The *Brute Force* and *XSS* web attacks are more occurring in the dataset than the *SQL injection* attack.

TABLE III. CLASS DISTRIBUTION OVER THE HTTP ENTRIES.

		CIC-IDS-2017		CIC-IDS-2018	
Class	**Type**	**Amount**	**Percentage**	**Amount**	**Percentage**
Botnet	DDoS	736	00.28%	142,925	04.28%
GoldenEye	DoS	7,908	02.97%	27,345	00.82%
HOIC	DDoS	0	00.00%	1,074,379	32.15%
Hulk	DoS	158,513	59.48%	1,803,160	53.95%
LOIC	DDoS	95,683	35.90%	289,328	08.65%
SlowHTTPTest	DoS	1,416	00.53%	0	00.00%
Slowloris	DoS	2,245	00.84%	4,950	00.15%
		266,501	100.00%	3,342,807	100.00%
Brute Force	Web	7,311	79.93%	13,144	54.02%
SQL Injection	Web	12	00.13%	57	00.23%
XSS	Web	1,824	19.94%	11,134	45.75%
		9,147	100.00%	24,335	100.00%
Benign	-	258,197	48.37%	6,252,950	65.00%
Malicious	-	275,648	51.63%	3,366,422	35.00%
		533,845	100.00%	9,619,372	100.00%

E. Models

Four ML algorithms are selected for our classification problem: Decision Tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN), and Gaussian Naive Bayes (GNB). A grid search approach is performed to find the optimal hyperparameters for these algorithms. Table IV shows the considered parameters for each model. The optimal set of parameters for each model is used on the test dataset by selecting the highest F_1 score achieved on a validation set. As there was a limited amount of computational time, the hyperparameter space of computationally expensive models like KNN is smaller than simpler models like DT.

TABLE IV. HYPERPARAMETERS OPTIONS FOR THE SELECTED CLASSIFIERS.

Model	**Scikit Parameter**	**Options**
GNB	var_smoothing	1e-200
DT	criterion	[Gini , Entropy]
	splitter	[Best, Random]
	class_weight	[None, Balanced]
	max_features	[Auto, None, Sqrt, log2]
RF	criterion	[Gini, Entropy]
	class_weight	[None, Balanced]
	max_features	Auto
	n_estimators	[10, 50, 100, 250]
KNN	n_neighbors	5
	algorithm	[Ball Tree, KD Tree]

IV. EXPERIMENTAL SETUP

In this section, we elaborate on the conducted experiments. Three different experimental setups were created in which we tested ML models to detect cyberattacks. Before we elaborate

in detail on those three setups, we start with mathematical preliminaries, how it is split into the train, validation, and test sets, and how we turned each experiment into a binary classification problem. After describing the experiments we will discuss how the models are evaluated and tested against a benchmark method named the Dutch Draw.

A. Preliminaries

Suppose we have an intrusion detection dataset $\mathbb{X}$ consisting of M instances and K feature values each. Without loss of generality, we assume $\mathbb{X} \in \mathbb{R}^{M \times K}$. Say $S \coloneqq \{1, 2, \ldots, M\}$ are the indices of the instances. We assume that each of those instances is labeled. Each instance s has a corresponding label y_s. Let $y \coloneqq \{y_1, y_2, \ldots y_M\}$ be the vector containing all labels. The datasets consist of *Benign* traffic and a set of malicious cyberattack variants. Therefore, let $C \coloneqq \{n, p_1, p_2, \ldots p_L\}$ be the set of possible values each instance y_i could have as label. Here, label n is the *Benign* label, and $\{p_1, p_2 \ldots, p_L\}$ is the set of different cyberattack labels included in our data. Let $B \coloneqq \{s \in S | y_s = n\}$ be the set of instances that are of the *Benign* class and let $P_k \coloneqq \{s \in S | y_s = p_k\}$ indicate the instances that have malicious class p_k as a label.

B. Train-Test split

It is common practice in ML to split a dataset into two non-overlapping sets: a training set and a test set. Binary classification models use the training set to learn a relationship between the response variables and the explanatory variable. Let us denote S_{train} as the instances that are assigned to the training dataset and S_{test} as the instances that are assigned to the test dataset. It should hold that $S_{train}, S_{test} \subset S$ with the property that $S_{train} \cap S_{test} = \emptyset$, $|S_{train}| + |S_{test}| = M$ and we should select a ratio R such that $|S_{train}| \cdot R = |S_{test}|$. Typically, R is selected in such a way that there is an 80:20 train-test ratio. As we investigate stochastic and deterministic prediction models, the train and test procedure is repeated multiple times to get a proper average performance for these models.

There are multiple classes in the dataset, so we have added another requirement for the train-test split: we want to have a similar class distribution in both the training dataset as well as the test dataset. This is also known as *stratified sampling*. The train-test split of the instances should match the class distribution of the original dataset as closely as possible:

$$\frac{|S_{train} \cap P_k|}{|S_{train}|} \approx \frac{|S_{test} \cap P_k|}{|S_{test}|} \quad \forall k \in \{1, 2, \ldots, L\}.$$

Furthermore, we enforce that there are at least two observations of each class selected to make hyperparameter tuning possible.

C. Hyperparameter tuning

In Section III-E, we discussed the considered hyperparameters for the selected ML classifiers. Selecting the right hyperparameters is vital for good performance. Hyperparameters are compared by taking the average performance of the ML models tested on different validation datasets. Train-validation splits are created in the same manner as the train-test split. The hyperparameters yielding the highest F_1 score are selected to test the models on the testing dataset.

D. Setups

The classification problem at hand could be a multiclass classification problem when $L > 1$. We will, however, treat each problem as a binary classification problem by mapping all malicious classes towards a single label $p_{Malicous}$ when y is presented to the model to train on and when evaluating. The classifiers are tested on these datasets in three different experimental setups:

1) Detecting Known Attacks: Firstly, we study to what extent the selected classifiers are able to detect known attacks in a closed-world learning setting. The achieved detection rate by the different ML models could indicate an upper bound to the novel detection rate of the corresponding malicious class. To test this, the training data and the evaluation data contain the same two classes: *Benign* and one malicious variant. More specifically: the training dataset for this first experiment $T_{1,k}$ consists of instances given to a model:

$$T_{1,k} = (B \cup P_k) \cap S_{train}.$$

And we evaluate the performance of the models on the following test set $E_{1,k}$ on which we evaluate:

$$E_{1,k} = (B \cup P_k) \cap S_{test}.$$

For example, we let a model train to distinguish *Benign* from *Hulk* and test on the same two classes.

2) Detecting Novel Variants: Secondly, we examine to what degree classifiers are able to detect a novel variant when the training dataset only contains *Benign* traffic and one different variant. For example, we let a classifier train on distinguishing *Benign* from *Hulk* entries and evaluate the trained model on a test set containing *Benign* and *LOIC* (Low Orbit Ion Cannon). Both the *Hulk* and *LOIC* labels are converted to one Malicious label, to keep the binary classification setting. This experiment shows us how similar the novel test attack is to the known training attack. Let us define $T_{2,i}$ as the training instances:

$$T_{2,i} = (B \cup P_i) \cap S_{train}.$$

In contrast to the previous experiment, the included malicious instances are not from the same class and the test dataset E consists of instances:

$$E_{2,j} = (B \cup P_j) \cap S_{test}.$$

When $i = j$ holds, we simply get the same results as experimental setup 1.

3) Class Importance to Detect Novel Variants: Finally, we study what we call *class importance*: how crucial is including a variant in the training dataset on the novel cyberattack detection performance? Does learning on a combination of multiple attacks help identify novel variants? We look at combinations of cyberattacks in the training set and test the

trained model on detecting a novel attack. For example, we train on *Benign* and a combination of attacks such as *LOIC* and *Hulk* entries and test on a dataset containing *Benign* and a different novel attack such as *SlowHTTPTest*. More formally, let us first select an evaluation dataset $E_{3,j}$ containing variant p_j:

$$E_{3,j} = (B \cup P_j) \cap S_{test}.$$

Now, we want to find which set of malicious variants leads to the highest novel cyberattack (p_j) detection rate. Say $\mathcal{P}(S)$ is the powerset of set S. By definition, the powerset of a set is the set of all subsets. So, the considered cyberattack combinations in the training dataset for novel attack p_j are derived as follows:

$$C_j \coloneqq \mathcal{P}(C \setminus \{n, p_j\}) \setminus \{\emptyset\}$$

The empty set is excluded from this powerset as there needs at least one cyberattack to be included in the training dataset. Suppose we take now a random cyberattack set $c \in C_j$. Let us define the set of instances having a label included in this set c as $\mathcal{C} \coloneqq \{s \in S | y_s \in c\}$. Then the training dataset of this third experiment $T_{3,c}$ is:

$$T_{3,c} = (B \cup \mathcal{C}) \cap S_{train}.$$

With this formulation, we can study which set of known variants $c \in C_j$ leads to the highest novel cyberattack p_j detection rate.

E. Evaluation Metrics

In our classification task, the *positive* class represents malicious instances while the *negative* class represents benign entries. Let us denote $y_s \in \{0, 1\}$ as the actual label of an instance s where 0 is the negative class and 1 represents the positive class. A confusion matrix is constructed by comparing the binary predictions $\hat{y}$ of a classifier with the actual labels y. This 2×2 dimensional matrix contains four base measures: the number of *true positives* (TP), the number of *true negatives* (TN), the number of *false positives* (FP), and the number of *false negatives* (FN). TP and TN show the number of instances that are correctly predicted, while FP and FN two show the number of mistakes. These four measures form the basis of any binary evaluation metric.

The performance of a binary classification model is quantified by one or more evaluation metrics, which are a function of one or more base measures. The considered evaluation metric to test the selected classifiers is the F_1 score, which is the harmonic mean between *recall* and *precision*. Recall is the ratio of intrusions the classifiers were successfully able to detect, while precision is the ratio between the *true positives* and the number of *positively predicted* instances. Hence:

$$Recall = \frac{TP}{TP + FN}, \quad Precision = \frac{TP}{TP + FP}.$$

Taking the *harmonic* mean between those two gives the F_1 score, which is defined as:

$$F_1 = \frac{2}{Recall^{-1} + Precision^{-1}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

For cyberattacks aiming to exhaust a resource, it is better to have a low false alarm rate than a high recall as it is not necessary to block all malicious traffic. We simply want to prevent the resource from being overloaded and prevent blocking legit HTTP requests. This makes the task at hand different in contrast to detecting intrusions in general as there the cost of a false negative is higher. Still, optimizing only precision is not desirable. Therefore, the F_1 score is an appropriate middle ground as it optimizes the harmonic mean of those metrics. When data is imbalanced, this score is more suitable than accuracy as it corrects this imbalance.

F. Dutch Draw Baseline

Stating the evaluation metric scores of the selected classifiers makes it possible to compare their performances. It does, however, not indicate what the scores themselves mean without some frame of reference. Baselines help interpret results as models can only be considered appropriate when outperforming them. Therefore, we have selected the Dutch Draw (DD) [9] as this baseline method helps in comparing the performances of the selected ML models. This method gives as a baseline value the score of the optimal random classifier that is input-independent. The selected evaluation metric to compare classifiers is the F_1 score and it follows from [9] that the corresponding *DD baseline* is given by:

$$\frac{2P}{2P + M}.$$

To construct a baseline for a test dataset E, the number of positives is given by $P = |E \setminus (B \cap S_{test})|$ and $M = |S_{test}|$.

V. Experimental Results

Now, we show the results of the three experimental setups performed in this research. The results of the experiments were gathered by testing the selected classifiers on 20 different train-test splits for the CIC-IDS-2017 and 10 different for the CIC-IDS-2018. Furthermore, as the CIC-IDS-2018 is very large and there was limited computational time, a subset of the data was used for hyperparameter tuning. For the DT and RF techniques, 10% was randomly selected for hyperparameter tuning. As the KNN model, with the selected hyperparameter options, is computationally very expensive, we were limited to only using 1% (randomly) of the training data for hyperparameter tuning. The same percentage of data was required in the training process to be able to evaluate this model in a reasonable time. For the CIC-IDS-2017, no subset sampling was required for training purposes. In our experiments, we have performed multiple hold-out-cross validation splits with each split an 80/20 split in a random manner. Before splitting the data, all redundant features (features with only 0 values) are removed as these features do not contain any new information. In the training set, a validation set (20%) is randomly selected to obtain the best hyperparameters for each model.

A. Detecting Known Attacks

ML classifiers were tested on whether they were able to distinguish benign HTTP interactions from interactions that were labeled with a predefined known cyberattack by training and testing on the same classes. We start with discussing the results for the Web attacks and afterward show the results for (D)DoS attacks. Table V shows the average F_1 scores for the selected ML models and the corresponding standard deviations. Here, each row corresponds to the selected cyberattack for the setup (P_k). The relatively lowest scores are highlighted in red. It can be observed that almost in all scenarios the classifiers outperform the Dutch Draw baseline, for which we have taken the average expectation over all train-test splits. For the CIC-IDS-2018, we observe that the GNB and KNN model outperform the Dutch Draw baseline, but the scores were relatively low. Here, the models were not able to detect any of the *SQL injection* instances in all train-test splits. All other setups indicate that the ML models were able to find patterns to distinguish normal traffic from a malicious variant.

TABLE V. EXPERIMENT 1 F_1 SCORES OF CLASSIFIERS DETECTING KNOWN WEB ATTACKS.

CIC-IDS-2017	**DD**	**GNB**		**DT**		**RF**		**KNN**	
Attack	**Exp**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
Brute Force	0.0536	0.5711	0.0034	0.9994	0.0008	0.9994	0.0003	0.9983	0.0005
SQL Injection	0.0001	0.9500	0.2236	0.8419	0.2690	0.8833	0.2484	0.0833	0.2059
XSS	0.0139	0.9044	0.0065	0.9975	0.0033	0.9950	0.0022	0.9901	0.0043
CIC-IDS-2018									
Brute Force	0.0042	0.8108	0.0059	0.9994	0.0003	0.9996	0.0002	0.9861	0.0018
SQL Injection	0.0000	0.0134	0.0006	0.8834	0.0591	0.8655	0.0834	0.0000	0.0000
XSS	0.0035	0.9977	0.0009	0.9998	0.0003	0.9999	0.0002	0.9927	0.0027

The same analysis is performed for (D)DoS attacks. Table VI shows the average F_1 scores if classifiers were tested on the task of detecting known (D)DoS attacks. It can be observed that in almost all scenarios the considered models were able to learn the relevant characteristics of the considered attacks. One exception is the GNB model that was trained and tested with the on the *SlowHTTPTest* attack. This model obtained a high recall (0.997), but a poor score on its precision (0.154). Even though the model is able to detect most malicious instances, there were many false positives.

TABLE VI. EXPERIMENT 1 F_1 SCORES OF CLASSIFIERS DETECTING KNOWN (D)DOS ATTACKS.

CIC-IDS-2017	**DD**	**GNB**		**DT**		**RF**		**KNN**	
Attack	**Exp**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
Botnet	0.0057	1.0000	0.0000	0.9971	0.0046	0.9998	0.0008	0.9909	0.0076
GoldenEye	0.0577	0.9972	0.0010	0.9997	0.0002	1.0000	0.0000	0.9983	0.0006
Hulk	0.5511	0.9990	0.0002	0.9999	0.0000	1.0000	0.0000	0.9999	0.0000
LOIC	0.4257	0.9999	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
SlowHTTPTest	0.0108	0.2339	0.2065	0.9955	0.0042	0.9956	0.0031	0.9874	0.0046
Slowloris	0.0171	0.9013	0.0078	0.9976	0.0016	0.9969	0.0023	0.9929	0.0035
CIC-IDS-2018									
Botnet	0.0437	0.9998	0.0001	1.0000	0.0000	1.0000	0.0000	0.9974	0.0011
GoldenEye	0.0087	0.9919	0.0006	0.9843	0.0010	0.9914	0.0004	0.9536	0.0051
HOIC	0.2558	0.9964	0.0001	0.9964	0.0001	0.9964	0.0001	0.9961	0.0002
Hulk	0.3658	0.9999	0.0000	1.0000	0.0000	1.0000	0.0000	0.9997	0.0000
LOIC	0.0847	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
Slowloris	0.0016	0.9876	0.0018	0.9982	0.0012	0.9986	0.0007	0.9586	0.0054

B. Detecting Novel Attacks with One Attack Learned

Let us relax the closed-world assumption: What if our trained algorithm sees a different variant of the learned attack? Figure 2 shows the average F_1 scores achieved by the classifiers in this experiment when detecting Web attacks. The diagonal of this matrix shows the F_1 scores of the closed-world assumption, also obtainable from Table V, while the off-diagonal values were the scores of detecting novel attacks. We can observe that the GNB model is not able to detect all *Brute Force* attacks. Surprisingly, a high score is obtained when this model is not trained on the *Brute Force* attack but on the *XSS* attack.

Figure 2. Experiment 2 average F_1 scores for the CIC-IDS-2017 to detect known and novel Web attacks.

If we now look at the results extracted from the CIC-IDS-2018 dataset, we see a difference in comparison to the CIC-IDS-2017. Figure 3 shows that the DT model is very useful to detect a *Brute Force* attack when trained on the *SQL injection* and is also able to detect the *XSS* when using the *Brute Force* attacks. The scores here were actually higher on the diagonal for the CIC-IDS-2018 than the CIC-IDS-2017.

Figure 4 shows the average F_1 scores achieved by the classifiers when detecting novel and known (D)DoS attacks. The diagonal of this matrix shows again the F_1 scores of the closed-world assumption, also obtainable from Table VI, while the off-diagonal values were the scores of detecting novel attacks. We observe in this open-world learning setting that *Botnet* attacks were hard to detect in this setting, and neither can they easily be used to detect other variants. However, there were situations where classifiers were able to detect novel variants. This is, however, not symmetrical: learning attack

Figure 3. Experiment 2 average F_1 scores for the CIC-IDS-2018 to detect known and novel Web attacks.

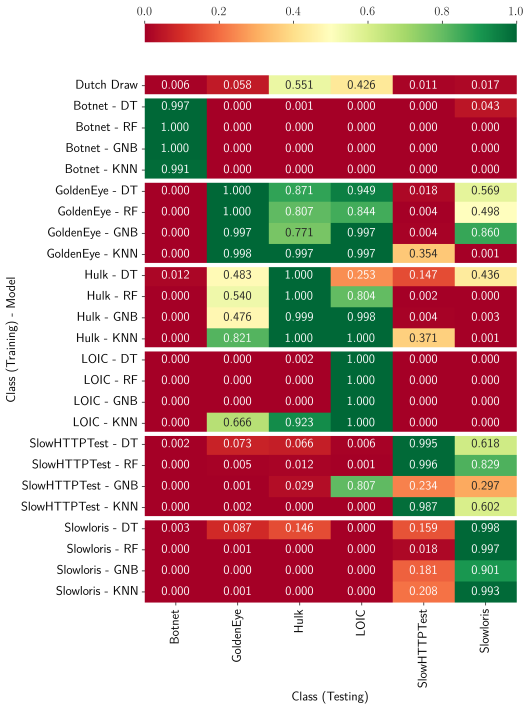


Figure 4. Experiment 2 F_1 scores averages for the CIC-IDS-2017 dataset to detect known and novel (D)DoS attacks.

A and finding attack B does not mean it works also the other way around.

Let us now look at the results of the CIC-IDS-2018 dataset containing *Benign* instances and (D)DoS attacks. Figure 5 shows the results of the same experimental setup performed on the CIC-IDS-2018. Similar results were observable on the diagonal: ML algorithms were indeed able to detect attacks it has trained on. In these results, it is less apparent that learning one (D)DoS attack leads to the model being able to detect another attack. Only a few combinations of train and test attacks were successful. For example, learning the *HOIC* (High Orbit Ion Cannon) with the KNN model results in high scores for testing on the *LOIC* and the *Hulk*. Results showed that classifiers such as DT and RF were not able to learn sufficiently from the training data as a striking class imbalance between benign and the attack led to low performance. Still, the same observation as in the CIC-IDS-2017 is apparent: when training on attack A and being able to detect B, it does not imply the reverse also holds.

Despite the 2017 and 2018 datasets having the same cyber-attacks, the HTTP interactions of the attacks are not identical. Figure 6 shows the results of the selected ML classifiers trained to detect a Web cybervariant of the CIC-IDS-2017 and tested whether the classifiers were able to detect CIC-IDS-2018 Web variants. We observe that there is no clear consistency between the ML models and whether they are able to detect known or novel attacks. Again, we see there are no symmetric results observable in the heatmap. It is not conclusive that the Web attacks in the CIC-IDS-2017 are identical to the CIC-IDS-2018.

Figure 7 shows the results of the selected ML classifiers trained to detect a (D)DoS variant of the CIC-IDS-2017 and tested whether the classifiers were able to detect CIC-IDS-2018 (D)DoS variants. It can be observed that in almost no situations the classifiers were able to do so. An exception here is the *Slowloris* attack, which has actually a high performance on all models except the GNB. This indicates that despite the datasets containing the same attacks, HTTP interactions were not necessarily identical.

C. Learning on a Set of Variants to Detect a Novel Variant

In our last experiment, we study which combination of cyberattacks in the learning phase results in the highest novel detection rate. Table VII shows the results when classifiers were trained on one or more Web variants to detect a novel Web variant. In bold is indicated the highest score obtained by the models and in the last column, the set of attacks that resulted in the bold score is stated. We observe that, for both datasets, the KNN is practically useless to detect novel Web variants. The *Brute Force* attack is always used in the training dataset to achieve the highest novel detection rate.

The same procedure and analysis were performed for the (D)DoS variant. Table VIII shows the results of the classifiers using a set of attacks to learn from and the corresponding combination of attacks that led to the highest performance. Despite the fact that models can use more attacks to detect

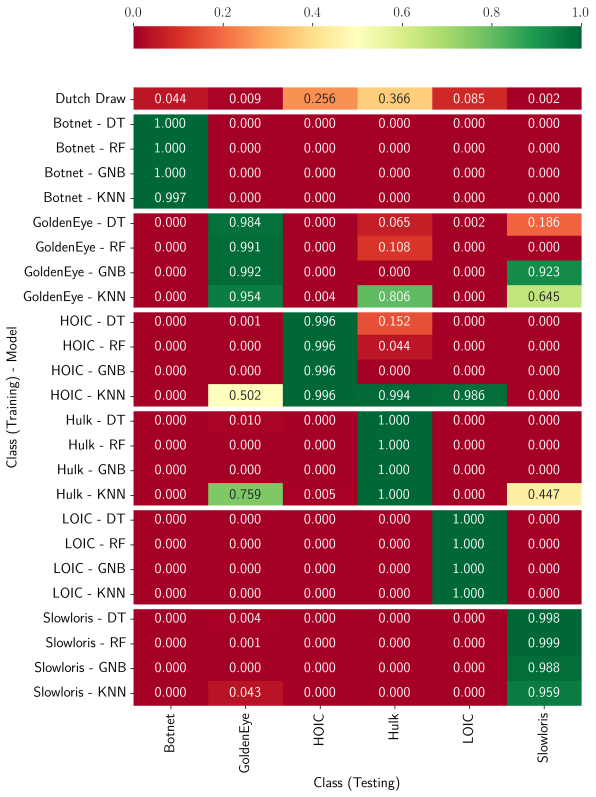


Figure 5. Experiment 2 F_1 scores averages for the CIC-IDS-2018 dataset to detect known and novel (D)DoS attacks.

Figure 6. Experiment 2 F_1 average classifier scores when trained on the CIC-IDS-2017 Web attacks and tested whether they were able to detect CIC-IDS-2018 Web attacks.

TABLE VII. EXPERIMENT 3 HIGHEST OBTAINED F_1 SCORE FOR EACH MODEL BY TRAINING THEM ON MULTIPLE INTRUSIONS TO DETECT A NOVEL WEB ATTACK.

CIC-IDS-2017	**DD**	**DT**	**GNB**	**KNN**	**RF**	**Train Set Opt Model**
Brute Force	0.054	0.202	**0.967**	0.000	0.100	{XSS}
SQL Injection	0.000	0.257	**0.400**	0.000	0.000	{Brute Force, XSS}
XSS	0.014	0.144	0.140	0.000	**0.327**	{Brute Force}
CIC-IDS-2018						
Brute Force	0.004	0.600	**0.767**	0.000	0.152	{SQL Injection, XSS}
SQL Injection	0.000	**0.366**	0.236	0.000	0.318	{Brute Force}
XSS	0.004	0.438	**0.735**	0.000	0.200	{Brute Force, SQL}

TABLE VIII. EXPERIMENT 3 HIGHEST OBTAINED F_1 SCORE FOR EACH MODEL BY TRAINING THEM ON MULTIPLE INTRUSIONS TO DETECT A NOVEL (D)DOS ATTACK.

CIC-IDS-2017	**DD**	**DT**	**GNB**	**KNN**	**RF**	**Train Set Opt Model**
Botnet	0.006	**0.460**	0.291	0.000	0.000	{Hulk, LOIC, Slowloris}
GoldenEye	0.058	0.664	0.476	**0.821**	0.782	{Hulk}
Hulk	0.551	0.870	0.986	**0.997**	0.833	{GoldenEye, LOIC}
LOIC	0.426	0.949	0.998	**0.999**	0.999	{Hulk}
SlowHTTPTest	0.011	0.240	0.181	**0.399**	0.100	{Hulk, Slowloris}
Slowloris	0.017	**0.878**	0.860	0.601	0.874	{Bot, Eye, Hulk, HTTP}
CIC-IDS-2018						
Botnet	0.044	0.000	0.000	0.000	0.000	-
GoldenEye	0.009	0.290	**0.862**	0.773	0.100	{LOIC, Hulk, Slowloris}
HOIC	0.256	0.000	**0.853**	0.500	0.000	{LOIC, Hulk}
Hulk	0.366	0.899	**0.999**	0.997	0.986	{GoldenEye, Slowloris}
LOIC	0.085	0.100	0.288	**0.985**	0.000	{HOIC}
Slowloris	0.002	0.539	**0.922**	0.837	0.000	{GoldenEye}

a novel variant, it is not necessarily the case that this yields the highest detection rate: even a few cyberattack classes were enough to obtain the highest performance. It can be observed that for the CIC-IDS-2017 the KNN model is dominantly getting the highest average F_1 scores, while for the CIC-IDS-2018 it is the GNB model. In neither case does the RF model outperform other models, which is unexpected as this model outperforms other models in detecting known attacks. For the CIC-IDS-2017 dataset, the *Hulk* attack is almost always used to obtain the highest scores with the least number of attacks required. The strong imbalance affects the learning process of the DT and the RF, similar to the results in experiment 2. These models could have been improved by downsampling benign entries so that the training classes were balanced.

VI. CONCLUSION

This research provides a procedure to construct intrusion detection datasets combining multiple layers with the tool Zeek. Zeek generates a set of extensive log files and two of them are selected to create an ML-admissible dataset for the detection of cyberattacks. This procedure to create such a dataset is not limited to only these protocols but can be extended to also combine other protocols, such as TCP connection with *File Transfer Protocol* interactions.

The aim of this research was to test to what extent ML

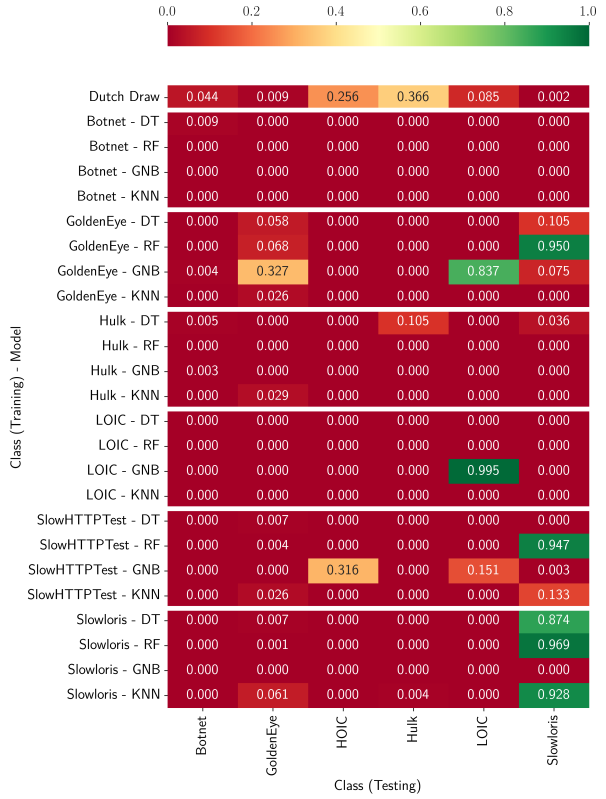


Figure 7. Experiment 2 F_1 average classifier scores when trained on the CIC-IDS-2017 (D)DoS attacks and tested whether they were able to detect CIC-IDS-2018 (D)DoS attacks.

classifiers are able to detect novel variants of known intrusions. A set of classifiers were applied in three different experimental setups and we studied their ability to detect variants. The focus of this research was to study the detection of variants of (D)DoS and Web attacks, but the same analysis can be performed on variants of another cyberattack. It has been shown in the first experiment that ML classifiers are to a great extent able to detect known (D)DoS attacks in a closed-world setting. For the Web attacks, the classifiers were not in all situations able to distinguish benign from malicious variants. Especially detecting *SQL Injection* instances with a GNB or KNN model was not accurate.

In the second experiment, it was observed that there are scenarios in which classifiers are able to detect a novel variant when trained on a different variant. Detecting novel variants is however not a two-way street: learning to detect attack A and being able to also detect attack B does not have the property that it is symmetrical. We have observed that for the CIC-IDS-2017 dataset the classifiers had a higher novel detection rate for (D)DoS variants than the results achieved on the CIC-IDS-2018. This was remarkable as the CIC-IDS-2018 contained similar attacks and more instances. It has been shown that the attacks are, however, not identical between the two datasets. Only the Slowloris seemed to have similar results between the datasets.

The third experiment showed that it is not necessary to use many malicious variants to detect a novel attack. Sometimes a few known attacks can already lead to the highest detection rate. Looking at the results of the (D)DoS attacks, DT and RF perform poorly in detecting novel attacks. The high imbalance in the training data caused this effect. The GNB model seemed more robust against this high imbalance in the training dataset and still achieved reasonable detection rates. For Web attacks, the results varied much between the model and cyberattack variant combination. The KNN model turned out to be effective in detecting known *Brute Force* and *XSS* attacks but was useless to detect novel Web attacks.

To sum up, this research shows that ML algorithms can, when sufficient training data is presented, detect cyberattacks almost as well as signature-based approaches, but also have the capability to detect novel variants. Selecting the right combination of an ML model with a (small) set of intrusion classes included in the training data can result in a higher novel intrusion detection rate.

References

[1] E. van de Bijl, J. Klein, J. Pries, R. D. van der Mei, and S. Bhulai, "Detecting novel variants of application layer (D)DoS attacks using supervised learning," in *IARIA Congress 2022: The 2022 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications*, pp. 25–31, 2022.

[2] S. Axelsson, *Intrusion detection systems: A survey and taxonomy*, 2000, unpublished, http://www.cse.msu.edu/~cse960/Papers/security/axelsson00intrusion.pdf, retrieved: December, 2022.

[3] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[4] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *2010 IEEE Symposium on Security and Privacy*. IEEE, 2010, pp. 305–316.

[5] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1–2, pp. 18–28, 2009.

[6] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[7] Y. Xin et al., "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[8] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, pp. 1–29, 2020.

[9] E. van de Bijl, J. Klein, J. Pries, S. Bhulai, M. Hoogendoorn, and R. D. van der Mei, "The dutch draw: constructing a universal baseline for binary prediction models," 2022, arXiv:2203.13084.

[10] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147–167, 2019.

[11] *NIDS*, December. 2022. [Online]. Available: https://github.com/etiennevandebijl/NIDS

[12] J. Zhao, S. Shetty, and J. W. Pan, "Feature-based transfer learning for network security," in *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*. IEEE, 2017, pp. 17–22.

[13] J. Zhao, S. Shetty, J. W. Pan, C. Kamhoua, and K. Kwiat, "Transfer learning for detecting unknown network attacks," *EURASIP Journal on Information Security*, vol. 2019, no. 1, pp. 1–13, 2019.

[14] P. Wu, H. Guo, and R. Buckland, "A transfer learning approach for network intrusion detection," in *2019 IEEE 4th International Conference on Big Data Analytics (ICBDA)*. IEEE, 2019, pp. 281–285.

[15] Z. Taghiyarrenani, A. Fanian, E. Mahdavi, A. Mirzaei, and H. Farsi, "Transfer learning based intrusion detection," in *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*. IEEE, 2018, pp. 92–97.

[16] M. Masum and H. Shahriar, "TL-NID: Deep neural network with transfer learning for network intrusion detection," in *2020 15th International Conference for Internet Technology and Secured Transactions (ICITST)*. IEEE, 2020, pp. 1–7.

[17] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*. SCITEPRESS, 2018, pp. 108–116.

[18] *A realistic cyber defense dataset,* Canadian Institute for Cybersecurity, December. 2022. [Online]. Available: https://registry.opendata.aws/cse-cic-ids2018

[19] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP 2017)*. SCITEPRESS, 2017, pp. 253–262.

[20] V. Paxson, "Bro: A system for detecting network intruders in real-time," *Computer Networks*, vol. 31, no. 23-24, pp. 2435–2463, 1999.

[21] M. Bijone, "A survey on secure network: Intrusion detection & prevention approaches," *American Journal of Information Systems*, vol. 4, no. 3, pp. 69–88, 2016.

[22] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[23] J. Klein, S. Bhulai, M. Hoogendoorn, R. Van Der Mei, and R. Hinfelaar, "Detecting network intrusion beyond 1999: Applying machine learning techniques to a partially labeled cybersecurity dataset," in *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE, 2018, pp. 784–787.

[24] *urllib* (3.9), December. 2022. [Online]. Available: https://docs.python.org/3/library/urllib.html

Design and Implementation of a Model-based Intrusion Detection System for IoT Networks using AI

Peter Vogl, Sergei Weber, Julian Graf, Katrin Neubauer, Rudolf Hackenberg
Ostbayerische Technische Hochschule, Regensburg, Germany
Dept. Computer Science and Mathematics
email:{peter.vogl, sergei.weber, julian.graf, katrin.neubauer, rudolf.hackenberg}@oth-regensburg.de

***Abstract*—The rising digitalisation introduces many useful features, as a result of, which the vulnerability for cyber attacks increases. Internet of Things devices and networks can be used to monitor and process sensitive data but at the same time they often are not hardened against threats. This can subsequently put personal data at risk. In this paper the capabilities of an approach to combine artificial intelligence and static analysis in an intelligent Intrusion Detection System for Internet of Things networks are evaluated. The development of static and dynamic methods for attack detection in networks is additionally discussed. The architecture follows a layer-based concept. Methods of classic security analysis and artificial intelligence are therefore deployed in a modular manner. For the extraction of important features a block-based approach has been developed, in which the calculated entropy of the network traffic is used in the extraction process. Detailed insights into the methodologies to analyse port and address information as well as used tools like Snort and Snorkel are given respectively. The metadata of the network traffic and extracted features are then used in combination to further improve the performance of anomaly detection and attack classification. The various models and algorithms utilised in this process are also shown in detail. This approach demonstrates that the security of Internet of Things environments can be enhanced with the deployment of an intelligent Intrusion Detection System that uses combined methodologies of static analysis and artificial intelligence.**

Keywords*—*Intrusion Detection; Network Security; Internet of Things; Artificial Intelligence; Machine Learning; Deep Learning.

I. Introduction

The methods and procedures presented are based on the publication "Design and Implementation of an Intelligent and Model-based Intrusion Detection System for IoT Networks" published at IARIA CLOUD COMPUTING 2022 [1] and are shown in more detail. The goal is to provide a deeper insight into the development of an intelligent Intrusion Detection System (iIDS) that goes beyond the original paper and thus also provides an extension to the latest research results.

Demographic change is a particular challenge worldwide. One consequence of demographic change is an ageing population. However, because of the now higher life expectancy, the risk of illness for each older person is also increasing [2]. For this reason, measures must be taken to enable the ageing population to live more safely.

Ambient Assisted Living (AAL) is used to improve the safety of people in need of assistance. AAL refers to all concepts, products and services that have the goal of increasing the quality of life, especially in old age, through new technologies in everyday life [3]. The iIDS described is part of the publicly funded research project Secure Gateway Service for Ambient Assisted Living (SEGAL). Within SEGAL, a lot of sensitive information such as heart rates, blood sugar or blood pressure are measured and processed. This kind of sensitive data is required in order to enable people in need of care to live in their familiar environment for as long as possible. To address this problem an AAL service is to be developed within the research project SEGAL. The purpose of the AAL service is to process the recorded data from Internet of Things (IoT) devices and send it via the Smart Meter Gateway (SMGW) to the AAL data management of the responsible control center from the AAL-Hub. The SMGW is a secure communication channel, as a certificated communication path is used for the transmission of the recorded data [4]. However, the exchange of data between IoT devices and AAL hub is not necessarily to be considered secure and can be seen as a target for attacks. Therefore, it is necessary to secure the communication between IoT devices and the backend system to prevent manipulation of the transmitted data. In this case, the iIDS is used to protect sensitive recorded data, as it is intended to detect possible attacks.

The increasing need for security is not limited to the healthcare sector. All IoT networks can be targets for various attacks. The Federal Criminal Police Office of Germany states in one of their reports that these networks can be used by malicious actors to amplify Distributed Denial of Service (DDoS) attacks [5]. The following case is a prime example mentioned in this report. In September 2019 Wikipedia's server infrastructure has been targeted with a DDoS attack that has likely been conducted by IoT devices and was therefore unreachable for several hours. These devices are not only used to perform attacks but can also be the target. In 2018 the number of attacks against Symantec's IoT honeypot has averaged to 5200 per month targeting mainly routers and cameras [6]. Their report also shows that approximately half of the usernames and passwords used in attacks are present in the Top 10 ranking. This greatly increases the attack surface and may further bring

the attention of threat actors to IoT devices.

The individual layers of the iIDS are designed to be easily integrated into cloud structures. This allows cloud to take advantage of flexibility and efficiency to monitor network security optimally. Therefore, security services can be scaled depending on the circumstances [7]. In addition, the cloud offers the possibility to improve new innovative Artificial Intelligence (AI) security analytics and adapt them to the supervision of different networks.

In [8] the architecture of the iIDS has been presented initially. The implementation of the intelligent and model-based iIDS, including the explanation of attack detection methods is further described in detail.

The structure is organised as follows: Section II describes the related work. Section III presents the architecture of the iIDS. In Section IV the rule-based modules of iIDS are described in detail. Section V deals with the Explorative Data Analysis (EDA), while Section VI describes data preprocessing, required for AI modules. The developed AI based modules are shown in Section VII, followed by a conclusion and an outlook on future work in Section VIII.

II. Related Work

In recent years, AI methods have been increasingly used in many different sectors including the healthcare sector. For the increasing number of IoT networks, possible cyber attacks needs to be detected reliably and conscientiously to guarantee security. Different approaches are used for the respective iIDS.

As Vinayakamur et al. [9] show deep learning approaches like self-taught learning can be an improvement for Intrusion Detection System (IDS). Also, Anomaly Detection (AD) approaches are commonly used. The usage of a bit-pattern technique for deep packet analysis is therefore one useful approach as Summerville et al. showed in [10]. McDermott et al. [11], on the other hand, use a deep learning approach to detect botnets in IoT networks. They developed a model based on deep bidirectional Long Short Term Memory using a Recurrent Neural Network. Burn et al. [12] are using a deep learning approach for detecting attacks, in which they use a dense random neural network.

The approach for the SEGAL iIDS differs in some aspects. On one hand, we use common network analysing methods further described as static methods and on the other hand we use state of the art AI approaches to detect anomalies and classify attacks. The previous mentioned approaches can detect anomalies, but none of them can classify attacks. Our goal is to achieve a false positive rate as small as possible by using AI algorithms and static based models.

Therefore, two major research questions are to be answered:

- **RQ 1:** Can AI and static analysis be sustainably implemented in practice-oriented iIDS in AAL environments?
- **RQ 2:** How can static and dynamic methods be developed and combined to improve network attack detection?

The goal is to answer the identified research questions by presenting procedures and techniques for achieving advanced network observation.

III. Architecture

The architecture of the iIDS consists of 5 layers, with an Observation Layer as its basis and an Action Layer as top layer. The organisation of the individual layers and their connections are shown in Figure 1.

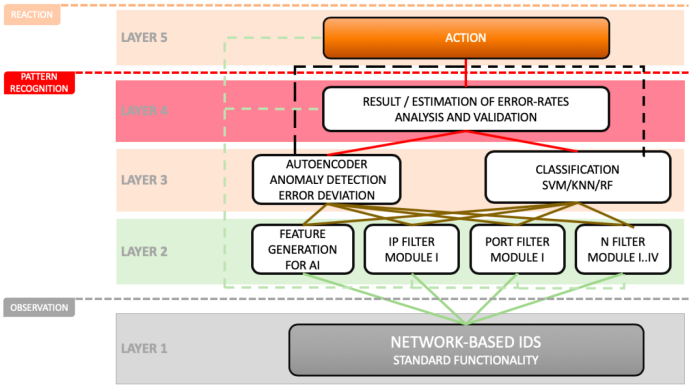


Figure 1. Architecture of the Intelligent Intrusion Detection System [8]

The Observation Layer, also called Data Collection Layer (DCL), implements the capturing and conversion mechanism to monitor network traffic and to extract the required transmission information. All data is also stored in a database for later usage. On top of the DCL, several rule-based modules are implemented to analyse and filter probably malicious traffic with static network observation methods. Also, the data preparation for the upcoming AI-based modules is part of this section. The third layer locates the different AI modules used to detect intrusions and to classify the type of attack. A deeper insight into these methods will be given in Section VII. All modules, rule-based and AI-based, are designed to return an assessment over their predicted outcome. In the penultimate layer, all the return values are evaluated and the probability of an intrusion will be calculated. Based on this calculation and through additional information for example, from the classifier in the third layer, the last layer can deploy dedicated security actions to prevent or limit damage to the system. Possible countermeasures could be notifications to an administrator, the shutdown of a connected device, or the interruption of the internet connection as a final action.

To get a lightweight and expandable system, all major components, like the iIDS itself, the AI-based modules, or the database, are deployed in their own Docker containers and can be managed independently.

IV. Rule-based Modules

As mentioned in Section III, the rule-based modules are part of the second layer in the presented architecture. They act as a first security barrier and are capable to give feedback on security issues based on observed network metadata of different ISO/OSI network model layer.

A. Analysing Port Information

Two modules are implemented to analyse the network's port information. The first one allows monitoring the individual

port usage. With an analysis of the network packages, the commonly used ports of the network participants can be discovered, which enables the ability to whitelist these ports. In reverse, packages, which do not have at least a whitelisted source or destination port number will be treated as a possible malicious package and an intrusion assessment value for the subsequent evaluation will be addressed to the next layer. The second module is designed to discover port scan attacks. The purpose of a port scan is to evaluate the open ports of a target system, which can be used to set up a connection. Despite a port scan may not be an illegal action, it is often used to get information about a target for later attacks. Because of this common intention and their easy execution with open-source software like Nmap [13], port scans can be treated in certain cases as threat indicators.

There exist multiple ways of conducting port scans. SYN scans, ACK scans and FIN scans use TCP packets with the according TCP flag set. Packets of NULL scans have no flags set while XMAS scans set the FIN, PSH and URG bits. Another possibility to determine open ports is to complete the three way handshake and close the connection immediately afterwards. There are also scans that use UDP and ICMP packets for reconnaissance [14].

One method to detect them is to monitor the network traffic and to define a threshold for the number of received packets from a source. Whenever within a specified time frame this threshold is exceeded by packets with the identically set TCP flags, it can be assumed that a port scan is conducted against the network. The source is determined by the source IP address of the packet. This threshold has to be adjusted to the regular network traffic in order to minimise false positives [14].

Another way of detecting port scans is proposed by Aniello, Lodi and Baldoni [15]. This approach combines three results of analysis into a rank. This rank is then compared to a defined threshold to determine if a scan is conducted. One step in this calculation is the determination of the entropy of failed connections from one source. For trustworthy sources this value will be near 0 while connection from scanners will be near 1. This is based on the way connections are established. A scanner will change either the destination IP address or the destination port with each request thus pushing the entropy value closer to 1. With this methodology the entropy value is calculated the following way where x = source IP address, y = destination IP address, p = destination port and $failures(x, y, p)$ = number of failed connections from x to y with destination port p [15].

$$N(x) = \sum_{y,p} failures(x, y, p) \tag{1}$$

$$stat(x, y, p) = \frac{failures(x, y, p)}{N(x)} \tag{2}$$

$$EN(x) = -\frac{\sum_{y,p}(stat(x, y, p)\log_2(stat(x, y, p)))}{\log_2(N(x))} \tag{3}$$

For the SEGAL iIDS both methods of these approaches are combined to get more consistent results while avoiding as many false positives as possible. The initial calculation of the entropy has been altered as described in equations (4) and (5) to fit the extended approach. First the number of packets from one source are categorised based on the layer 4 protocol of the ISO/OSI model and the set TCP flags. If the number of packets in one category exceeds the previously defined threshold, it is considered possible that the source is conducting a port scan.

Subsequently, the entropy value of the suspicious packets is determined with slightly adjusted calculations where $suspicious(x, y, p)$ denotes the number of suspicious packets. If this value is close enough to 1, the iIDS assigns these packets to a scan.

$$N(x) = \sum_{y,p} suspicious(x, y, p) \tag{4}$$

$$stat(x, y, p) = \frac{suspicious(x, y, p)}{N(x)} \tag{5}$$

$$EN(x) = -\frac{\sum_{y,p}(stat(x, y, p)\log_2(stat(x, y, p)))}{\log_2(N(x))} \tag{6}$$

B. Analysing Address Information

Part of the captured data from the Data Link Layer and the Network Layer is the address information. The data link layer and the network layer represent layers 2 and 3 in the ISO/OSI model and contain the necessary metadata to transmit network packets to a host in a destination-oriented manner. Based on the unique MAC-Address and the allocation of a static IP the trusted network members can be verified. A comparison of this information can be achieved by using whitelisting or blacklisting procedures. To further enhance security also the state of the dynamic host configuration protocol is analysed for violations of thresholds, such as IP range limits. The obtained information is also used to support the AI-based modules and provides important indicators for the Action Layer to defend against attacks.

C. Snort

Snort is a free network intrusion detection system (NIDS) and a network intrusion prevention system (NIPS) developed by Martin Roeasch. By using Snort it is possible to protocol IP packets and to analyse data traffic in real time [16]. The basis for pattern recognition is the Aho-Corasick algorithm [17].

Rules are the foundation of Snort's functionality. A distinction is made between two parts of the rule. These two parts are a general Rule Header and a more detailed specification by Rule Options. The header specifies the IP addresses and ports that are to be examined in more detail. In case of a detected signature, the Rule Header also defines the reaction to be performed. The Rule Options define further details of signatures and actions in case of detected intrusions. All rule options can be assigned to four different categories [18].

- **general:** In these options information about the rule can be found. However, these options have no effect on the detection.
- **payload:** These options are used to search for data within the payload. The options can also be linked with each other.
- **non-payload:** These options are used to search for non-payload data.
- **postdetection:** These options are rule-specific triggers. They are used after a rule was applied.

Snort is going to be used alongside the other static modules in the second layer of the SEGAL iIDS. To detect possible attacks, all recorded network traffic is compared against the configured rules. For the SEGAL iIDS, the community rules are used, which are a collection of rules submitted by members of the open source community or Snort integrators. Since the community rule sets are constantly maintained, Snort is able to quickly adapt to new attacks. This enables the SEGAL iIDS to react permanently and quickly to new threats. The rules are used to define malicious network activities. If a match is found against the rules, an alert is sent about the security issue. Snort can be considered one of the first security barriers for this reason.

V. Explorative Data Analysis

EDA provides a statistic insight into a given data set, enables the recognition and visualisation of dependencies, outliers and anomalies, and forms the basis for further feature extractions [19].

A. Data Insights

The used data set for training and testing the AI-based modules is based on a laboratory replica of a smart home (SHLab) that delivers network data from common IoT devices. Table I shows the scope of the used data set based on different labels. Two-thirds of the data are packages from normal daily data traffic, one-third are attack packages. Most of the malicious data are DDoS attacks, divided into SYN-, PSH-ACK-, FIN-, ICMP- or UDP-floods, but also WiFi-Deauthentication attacks are included.

TABLE I
Composition of the Used Data Set

Intrusion Class	Packages
Normal Data	**908355**
Wi-Fi-Deauthentication Attack	**32049**
DDoS Attacks	**468769**
— SYN-Flood	147849
— FIN-Flood	27408
— PSH-ACK-Flood	20971
— ICMP-Flood	185058
— UDP-Flood	87483
Combined Dataset	**1409173**

The SEGAL iIDS focuses on the analysis of network metadata. Since it is not always possible to collect information about the payload due to cryptography, meta information such as payload size and others are captured. Overall, 192 different data features are processed. This includes the address and port information mentioned in Section IV and, furthermore, data from the Transport Layer, for example, the TCP flags or checksums.

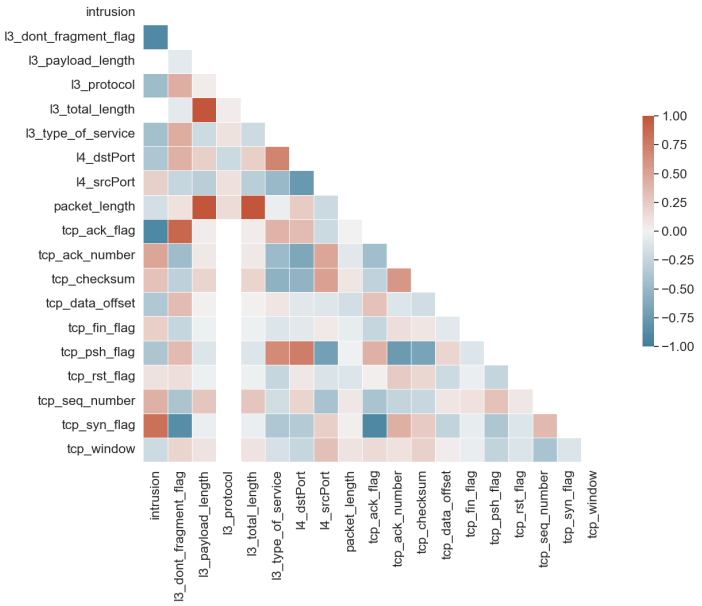


Figure 2. Feature Correlations

A correlation analysis allows a better insight into the correlations between important features. How well the features fit together is indicated by the correlation coefficient. The coefficient scales from – 1 to 1, whereby a value of 0 indicates no correlation between two features and -1 and 1 both indicate a strong linear correlation. Figure 2 shows a correlation matrix of the most important metadata.

One of these important features is the packet length. A comparison of normal data and attack data showed that attack packages have in average a significant lower packet length. Furthermore, the evaluation revealed that the trivial common DDoS attacks don't change the packet length over the attack time span. Another feature is the data offset of TCP packages, which is an indicator for the header size containing the position of the payload in a packet. In addition to a shorter packet length, a more detailed insight showed that attack packages also have a shorter header length, which leads to a smaller data offset. DDoS attacks aim to flood their target with a large number of packages. To achieve this, it is useful to have the bare minimum of packet size. This contains a small payload size and the least amount of header options, resulting in a small data offset. These and several additional network characteristics, such as port, flag, protocol information, are examined in order to be able to derive sustainable input for the iIDS modules.

B. Feature Extraction

For the extraction of new features from the data set two different concepts were developed. Both approaches derive additional information from the temporal context of the network

packets. However, a distinction is made here as to when or to what extent network packets are measured at the node. In both processing methods, the incoming network packets are aggregated into small blocks of different characteristics further called as block-based approaches. Hereby the data is ether combined to a specific number of packets measured on the order of arrival on the node or measured on passing the node in specific time windows.

Quantity-Blocks: Combines the data captured by the observation level of the iIDS to equal sized blocks calculated on a specific count value.

Time-Window-Blocks: Combines the data captured by the observation level of the iIDS to equal sized blocks based on fixed time frames.

In principle, both the quantity-blocks and the time-window-blocks approach can be used for networks of all sizes. Depending on the type of system, both methods are differently suited and thus have both advantages and disadvantages. In a low-volume network, the quantity-block approach may be too slow to trigger a timely response from the iIDS. Due to time differences in the arrival of network packets, long waiting times may occur until the desired block size is reached. In this case, the time-window-block approach would allow a more continuous analysis and a faster reaction. However, it can generally be said that the best results were achieved by a combination of both approaches. In the example just given, the time-window-block approach is able to compensate time delays. By combining the two approaches, the time-window-block approach can be used for a preliminary analysis until there are enough packages for a quantity-block analysis.

To detect network attacks the incoming packets of each local network member is separated by destination IP or MAC addresses. These packets are then processed by both methods and combined to quantity- and time-window-blocks. This allows a device-specific analysis of the network traffic (Figure 3). Through this combined concept time-based correlation can be used for each device to extract additional features to enhance AI-based attack pattern recognition. This modern, unconventional approach makes it possible to find new classification patterns and integrate them into the analyses of the iIDS.

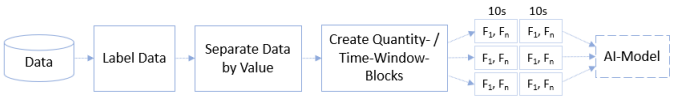


Figure 3. Feature Generation Process

These block-based classification features can be extracted through an analysis based on stochastic methods. Figure 4 shows for example the analysis of payload entropy based on the SHLab network device communication.

Entropy can be seen as a measure of uncertainty. The higher the distribution of values, the higher the value of entropy as a measure of this distribution. The highest value is reached when the dispersion of the values takes on an uniform distribution

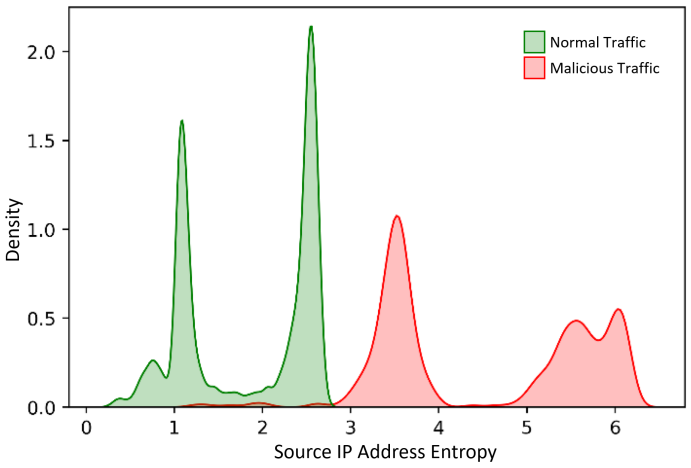


Figure 4. Source IP Address Entropy Comparison Between Normal and Malicious Network Traffic

over all possible outcomes [20]. The calculation of the entropy of a feature is based on the formula developed by Claude Shannon, also known as Shannon entropy [21]. The probability that a certain feature value occurs is specified by the parameter p_i. The individual calculated values are subsequently summed up. The parameter m equals to the number of packages in a quantity- or time-window-block. The result represents the entropy of a feature within a block.

$$H_1 = -\sum_{i=1}^{m} p_i \cdot \log_2 p_i \tag{7}$$

In the example shown in Figure 4, the entropy is the measure of the distribution of the source IP addresses. In more detail, the entropy indicates how scattered the source IP addresses of a block are. The possible outcome in this example is the scope of recorded source IP addresses. A high entropy value of a block indicates that the packets in a block originate from many different communication partners. Conversely, a low value indicates a limited variety. Figure 4 shows that in the case of an attack, the IP address entropy compared to the normal data traffic increases. To maximise the damaging effect, flooding attacks, such as the SYN flood shown here as an example, are usually carried out by different hosts simultaneously. During an attack, the source IP addresses are more evenly distributed over the entire recorded source IP addresses, and this leads consequently to an increased entropy.

VI. Data Preprocessing

Preprocessing the data is also performed on the second layer along the rule-based modules. Data preprocessing consists of cleaning, labelling, encoding, normalisation and standardisation of the captured data.

A. Encoding

The encoding of the captured package data is an important step for later usage. The captured network information like the MAC or IP addresses but also the different protocol types are

stored in a database. Another already mentioned reason for encoding parts of the data is to make it accessible for the AI modules. The majority of models require specific data types.

In order to ensure these requirements, two possible solutions were evaluated. The first one was to exclude this data from later usage. This is not a suitable option because of the importance of the information for the classification. To make the information usable for the AI-based modules a label encoding function is used. Label encoding replaces the distinct categories, i.e., the unique MAC addresses, with a numeric value. Through this, the specific value is lost, but the overall correlation is still valid, which is important for Machine Learning (ML) usage.

B. Cleaning

Missing data and NaN values are an additional problem for most AI models. Due to the huge variety of protocols used in network traffic the entries in the data set often contains empty fields. In example, an IPv4 package has no specific IPv6 information and vice versa. Features with more than 20% missing data entries are removed from the data set, because no meaningful statistical interpolation parameters can be calculated from the limited data stock, which can fill the missing gaps without significant errors. This doesn't apply for all network values. Therefore, for the other features, we use different interpolation methods based on the specification of the feature to deal with missing data. This includes the use of mean and median interpolation for the empty data fields.

C. Normalisation and Standardisation

Feature scaling is an often-done step in the data preprocessing phase of most AI-based models. It is not an essential requirement and not all algorithms benefit in the same way from this process. However, it can lead to better learning performance.

There are two major ways to perform feature scaling: Normalisation and Standardisation. The normalisation, also often called min-max-scaling, converts the original range of individual features to a general scale for all features. A common interval for this scale is [0, 1] [22]. Figure 5 shows an example of Min-Max-Scaling of ports.

The standardisation shapes the feature values in the proportion of a normal distribution. The mean value of the normal distribution is calculated over the elements of a feature. Unlike normalisation, the interval limits are not given values. However, the standard deviation from the mean value is used to set the scale for the feature rescaling. Standardisation is often used for data with a natural standard distribution [22].

D. Snorkel

Snorkel is a framework for labelling AI training data based on the work of Alex Ratner [23]. The proposed solution is to enable the developer to implement labelling functions, which programmatically imply rules to label the data.

The implementation process starts by aggregating the data over 10 second time intervals. As already mentioned in Section V-B this is necessary to improve the performance of the AI-based modules and Snorkel is able to benefit from this procedure too. Figure 6 illustrates how the size of the time interval affects both the runtime and the accuracy of Snorkel.

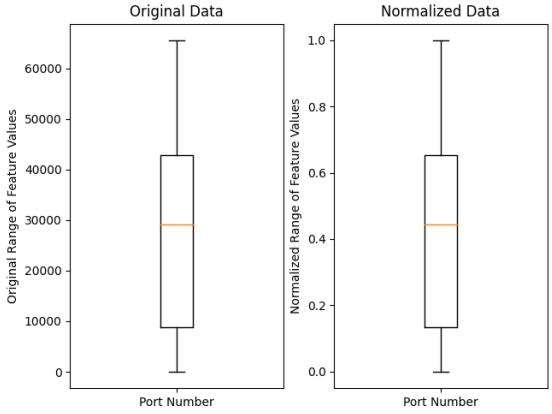


Figure 5. Comparison of Original and Normalised Data

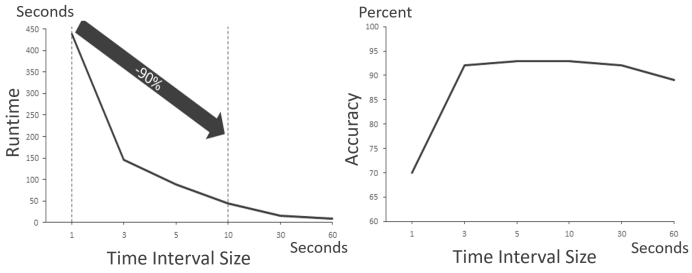


Figure 6. Impact of the Time Interval Size on the Runtime and Accuracy of the Snorkel Training Process

Figure 6 shows the average runtime of a training run over the entire data, when the recorded network packets are summarised over a 1 second intervals. Thereby, an average accuracy of approximately 70% is achieved. By increasing the time interval, the runtime of the Snorkel model can be reduced and the accuracy can be increased. With a time interval between 3 and 10 seconds, the accuracy of the Snorkel model stabilises at approximately 90%. However, longer intervals lead to a decrease in accuracy. Since the accuracy is almost the same with a time interval between 3 and 10 seconds, a 10-second interval is used in the further development. Another advantage of the 10-second interval is that the runtime of the Snorkel model can be reduced by approximately 90%.

The aggregated data is used to generate specific indices for each intrusion class. Most flooding attacks don't change their parameters during an attack, therefore the assumption is that the most common parameter subsets belong to flood packages. For a TCP flood, this leads to an index with the destination port and the packet length as parameters. This approach is also flexible enough to handle continuous new data from the SHLab without the need for changes. Trained on the data set mentioned in Section V-A Snorkel is able to classify all aggregated entries with an accuracy of 90-95%.

The difficulty is to find a classification model with two important properties. The first requirement is a good training result with the aggregated and labelled data delivered from Snorkel. The second requirement is a high accuracy on normal network data delivered from the SHLab. To test these requirements different classification models are used.

VII. AI-based Modules

AI enables the consistent analysis of complex data through the use of special architectures and deep learning techniques. In the following, the two architectures of the developed neural networks are presented. As shown in Figure 1, the AI-based modules are located in the third layer of the architecture. Three different modules are developed, whereby two are used to detect anomalies and one for attack classification. The first module for AD is implemented through the use of an neural network, which is based on our previous publication where we described the theoretical approach. The second module relies on the use of binary trees to isolate anomalies. The last module is trained to classify attacks and is based on a pretrained VGG19 [24] Convolutional Neural Network (CNN).

A. Anomaly Detection

To detect an attack there are two major ways, Signature Detection (SD) and AD. The advantage of SD is that known attacks can be detected very fast and with a high degree of precision. The downside is that this method needs a well-maintained database with historical and actual attack signatures. This leads to a higher administrative burden and consequently, the system would be more costly. The AD avoids this disadvantage by monitoring the network and building a reference for the usual daily traffic. This allows the AD to recognise new and unknown attacks, which would be overlooked by a SD-based system. But due to this characteristic, the AD is also prone to false-positive alarms because changes of the network traffic, for example, by bigger updates, can exceed the normal frame of reference.

1) Autoencoder-Anomaly-Detection-Model: To detect anomalies, a special neural network architecture called Autoencoder is used. Their specific process logic allows the neural network to learn without any supervision. Autoencoder are useful tools for feature detection and dimensionality reduction. Autoencoder reduce a given input to a lower dimensional space. This has the consequence that the most important network information is elaborated. From this point on a reconstruction process is started to extrapolate the original input from the so called bottleneck, as shown in Figure 7.

After the training phase, the Autoencoder has learned to reconstruct the input information based on the reduced information in the bottleneck. This means the reconstruction error of an extrapolated package compared to the input package on learned representations is small. In reverse, the reconstruction of an attack package, which is not part of trained behaviour differs compared to reconstruction error of normal network traffic. Based on the characteristics of the reconstruction error

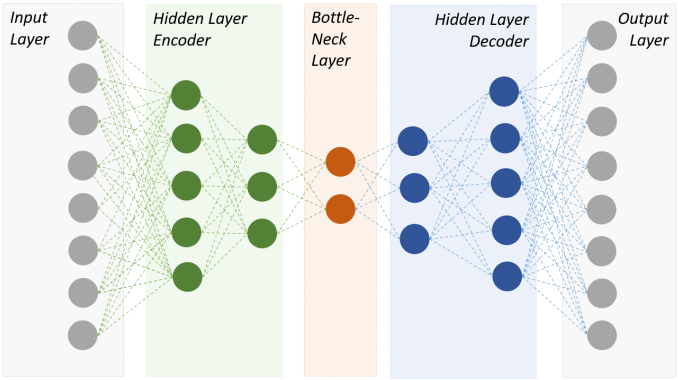


Figure 7. Basic Architecture of a Deep Autoencoder

we can calculate the probability of an network anomaly. However, the Autoencoder cannot specify the specific kind of attack. This means classification models are good enhancements.

2) Isolation Forest: In addition to the Autoencoder, an Isolation Forest (IF) Model is currently under development to extend the existing iIDS. The IF is another AI-based method for the unsupervised detection of anomalies in the monitored network. Combined with the results of the Autoencoder, this allows for potentially higher accuracy in detecting anomalies in network traffic. To detect these anomalies, the IF relies on two basic properties that must be present. The anomalies should only make up a small part of the data. Furthermore, the values of the anomaly should differ substantially from those of the normal data. Based on these properties, the anomalies are isolated from the normal data by the IF algorithm. Similar to a random forest, the IF is based on a set of decision trees. These decision trees are also called isolation trees. During the training process of the model, random subsets from the captured network traffic are passed to the different isolation trees. The partitioning of these subsets are then carried out by the usage of a random feature selection based on the captured network metadata. The result of the training process is a set of differently trained isolation trees, which together form the IF. The probability of whether a packet is an anomaly or not is expressed by an anomaly score. Due to the aforementioned property of anomalies that they differ substantially from normal data, the anomaly is usually isolated near the root of the isolation tree, as shown in Figure 8. A short path from the root node to the decisive leaf, indicates an increased probability that the examined package is an anomaly. Conversely, a longer path is more likely to indicate normal network traffic. The anomaly score can thus be derived from the path length and is calculated by averaging all path lengths of the different isolation trees [26] [27].

An advantage of IF is the performance of the AD. Especially in networks with high network traffic, a fast processing of the packets is a decisive criterion in order to be able to initiate appropriate steps in a timely manner. However, like the Autoencoder mentioned above, the IF can only detect

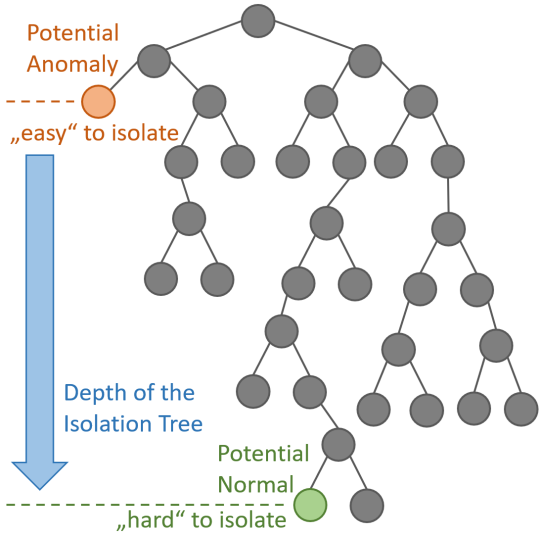


Figure 8. Architecture of an Isolation Tree [25]

malicious network traffic but not classify it in more detail. A more precise classification is therefore still necessary.

B. CNN-Classification-Model

Through a precise classification of threats, we gain additional information, which can be used to deploy countermeasures in the Action Layer. The used classification model is based on the VGG19 Model, which was developed by the Visual Geometry Group of the University of Oxford. A schematic illustration of the VGG19 architecture is shown in Figure 9.

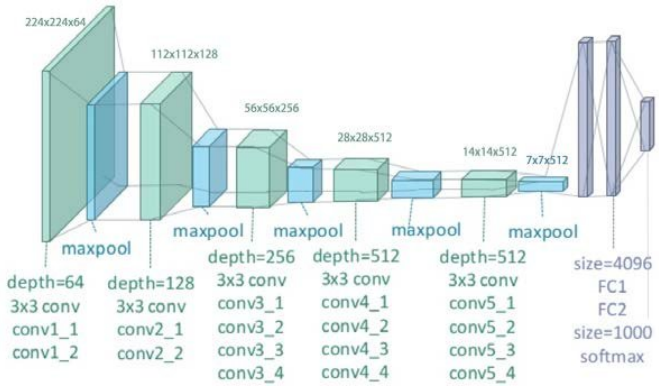


Figure 9. Architecture of the VGG19 Model [28]

The model consists of 16 convolutional layers with a filter size of 3x3 pixels. As shown in Figure 9, the convolutional layers are reduced by 5 maxpooling layers with a window size of 2x2 pixels. After the final reduction, the image data is further processed by 3 fully connected layers. The first two layers work with 4096 channels. For the third layer, the number of channels was reduced to 1000. The architecture of the CNN is completed by a final layer, which features a soft-max activation function [24].

1) CNN: The implementation of the classifier followes the assumption that the conversion of network packages into images can lead to better classification performance. Based on [29] there are 3 different approaches under development for the transformation process of the network data into RGB images. As Figure 10 illustrates, the transformation for all three approaches is based on the same data set. To obtain comparable results, regardless of the approach, the model is used for classification.

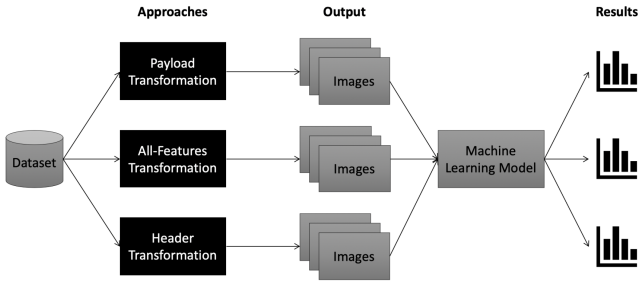


Figure 10. Overview of the RGB Image Transformation Process [29]

The first approach is based on the transformation of the payload data of the respective packet transmitted as an encrypted byte stream. The result of the transformation is a squared RGB image. For the transformation, the basic requirements of the VGG19 CNN for the image to be analysed must be observed. The first requirement is that the image must be in a square format. Also, all images need a minimum width of 32 pixels. Since the analysed picture is an RGB image with three separate colour channels, the minimum results is a 32x32x3 matrix with at least 3072 values. However, since payload data varies greatly in size, the transformation process would also result in images with different resolutions. Since this violates the requirements of the model, all fields of the matrix are initialised with 0. This can be seen as a representation of a completely black image with the necessary minimum dimensions. The individual colour channels, red, green and blue, are each represented by one byte for each pixel. Since the payload data is transmitted as a byte stream, these bytes can be written into the matrix without further processing to replace a portion of the fields previously initialised with 0. This matrix is then transformed into an RGB image that can be classified by the VGG19. Figure 11 shows three results of this transformation process.

Figure 11. RGB Images based on the Payload Transformation Approach [29]

In contrast to the first approach, the second approach in-

cludes not only the user data but also the corresponding header information of the packets in the transformation. The general procedure of the transformation does not change, but the header information must be converted in advance. The reason for this is that the string and integer values of most header information exceed the necessary value range of 0 to 255 for the transformation. As aforementioned, this range represents the 8 bits for each RGB colour channel. As described in Section VI-A all non-numeric values had to be transformed before rescaling. Based on this, a normalisation with an Min-Max-Scaler was performed. Different to the procedure in Section VI-C the range for the Min-Max-Scaler was set to 0 – 255. After rescaling the header information, all values are now within the necessary value range. The header information, each one byte in size, can now be written into the previously declared matrix together with the payload data using the same procedure as in the first approach. However, the images created after the transformation differ only minimally in the number of pixels from the RGB images shown in Figure 11.

The last approach for creating the RGB images for the CNN focuses exclusively on the use of the header information. As described in the second approach, the header information must first be rescaled. Afterwards, the header information can be transformed into an RGB image as also known from the first approach. In contrast to the transformation of the payload data, only a few pixels can be extracted from the header information for the RGB image. However, since the minimum dimensions still apply, the majority of the resulting image would be black. To counteract this problem, the initialised matrix is completely filled for each network packet by repeating the header information. Figure 12 shows three results of the third approach, which are significantly different from those of the payload transformation.

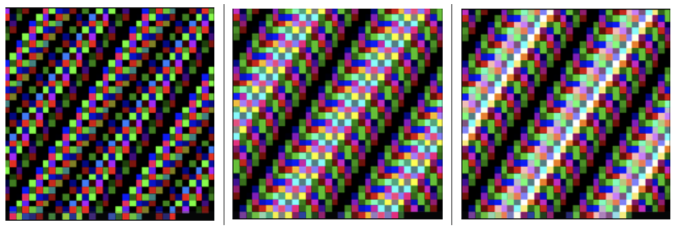


Figure 12. RGB Images based on the Header Information Transformation Approach [29]

First tests with this classification model delivered promising results. However, further tests with larger and more heterogeneous data sets are necessary to verify these results.

VIII. Conclusion and Future Work

The presented architecture provides the basis for the implementation of the static and AI-based methods for the SEGAL iIDS. The conceptual design of SEGAL iIDS is to combine different network monitoring methods in such a way that they can be operated both locally and in the cloud. The static methods are analysing information of the recorded network traffic. This allows the SEGAL iIDS to detect possible attacks in the network and alert about these security issues. The static methods also include the analysis of the port information. The analysis is done by categorising the number of packets. In the same layer as the static analysis, EDA and data preparation are performed. The gathered information of EDA derives the most relevant features for the subsequent AI modules. Data preprocessing prepares the data set for the usage in static and AI modules. Different AI algorithms are implemented. The first module is used to detect anomalies using a special architecture of neural networks. By dimension reduction of the input space and subsequent extrapolation from the smaller dimension space, network anomalies can be detected by analysing the reconstruction error expression. The development of an IF Model is expected to further improve AD. This approach is used to isolate anomalies from the normal data. Randomly sub-sampled network data is processed in a tree structure. Any samples that go deeper into the structure of the tree are less likely to be anomalies because multiple cuts are needed to isolate the samples. Otherwise, samples that end in shorter branches indicate anomalies because it was easier for the tree to separate them from other observations. The combination of these two AD methods empowers the SEGAL iIDS to detect anomalies in network traffic with increased accuracy. The second module is used for the classification of attacks. Here, data blocks are processed by time and number to create RBG image data. The CNN can then classify network attacks based on certain patterns within the image data. The data processing steps and data analysis methods described in the paper show that static and dynamic methods can be developed and combined in practice to provide better network monitoring. The presented iIDS differs from conventional IDS by the modular structure and also by the outsourced preprocessing. Due to the planned module layers, the iIDS to be developed can be used in different application areas without problems. The outsourcing of preprocessing allows the iIDS to be used on the different systems without performance loss.

In the future, a more detailed evaluation layer is to be developed. In order to achieve the desired improvement, an algorithm will be developed to enhance the aggregation of the static and AI-based module results. Due to these changes the SEGAL iIDS should be able to find even more appropriate countermeasures for detecting attacks. Furthermore, the already existing static and AI-based modules should be further expanded. In the case of the static modules, the detection logic is to be improved, whereby more attacks will be detected by the SEGAL iIDS. With regard to AI-based modules, the classification of the detected attacks should be improved. Thus, attacks detected by iIDS can be better classified. Deep package inspection is also to be used in the SEGAL iIDS allowing monitoring, analysing, filtering and marking of all data packets in the network.

References

[1] P. Vogl, S. Weber, J. Graf, K. Neubauer, and R. Hackenberg, ”Design and Implementation of an Intelligent and Model-based Intrusion Detection System for IoT Networks”, in Proc. CLOUD COMPUTING 2022 Special Track FAST-CSP, Barcelona, Spain, 2022, pp. 7-12.

[2] Robert Koch Institut, "Demographischer Wandel", 2020, [Online] Available at: https://www.rki.de/DE/Content/GesundAZ/D/Demographie_Wandel/Demographie_Wandel_node.html [retrieved: February, 2022].

[3] AAL-Deutschland, "Ambient Assisted Living Deutschland - Technik die unser Leben vereinfacht", 2016, [Online] Available at: http://www.aal-deutschland.de/ [retrieved: February, 2022].

[4] Bundesamt für Sicherheit in der Informationstechnik, "Smart Meter Gateway Dreh- und Angelpunkt des intelligenten Messsystems", 2022, [Online] Available at: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Smart-metering/Smart-Meter-Gateway/smart-meter-gateway_node.html [retrieved: March, 2022].

[5] Bundeskriminalamt, "Cybercrime Bundeslagebild 2019", 2020, [Online] Available at: https://www.bka.de/SharedDocs/Downloads/DE/Publikationen/JahresberichteUndLagebilder/Cybercrime/cybercrimeBundeslagebild2019.html [retrieved: October, 2022].

[6] Symantec, "Internet Security Threat Report", Vol. 24, 2019, [Online] Available at: https://docs.broadcom.com/doc/istr-24-2019-en [retrieved: October, 2022].

[7] M. G. Avram, "Advantages and challenges of adopting cloud computing from an enterprise perspective", Procedia Technology, Vol. 12, 2014, p 529-534.

[8] J. Graf, K. Neubauer, S. Fischer, and R. Hackenberg, "Architecture of an intelligent Intrusion Detection System for Smart Home", in Proc. 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Austin, Texas, USA, 2020, pp. 1-6.

[9] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System", IEEE Access, Vol. 7, 2019, pp. 41525-41550.

[10] D. H. Summerville, K. M. Zach, and Y. Chen, "Ultra-lightweight deep packet anomaly detection for Internet of Things devices", in Proc. 34th IEEE International Performance Computing and Communications Conference (IPCCC), Nanjing, China, 2015, pp. 1-8.

[11] C. D. McDermott, F. Majdani, and A. V. Petrovski, "Botnet Detection in the Internet of Things using Deep Learning Approaches", in Proc. 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 2018, pp. 1-8.

[12] O. Brun, Y. Yin, E. Gelenbe, Y. M. Kadioglu, J. Augusto-Gonzalez, and M Ramos, "Deep Learning with dense random neural network for detecting attacks against IoT-connected home environments", in Proc. First International ISCIS Security Workshop, London, United Kingdom, 2018, pp. 79-89.

[13] M. Shah, S. Ahmed, H. Khan, "Penetration Testing Active Reconnaissance Phase – Optimized Port Scanning With Nmap Tool", in Proc. 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2019, pp. 1-6.

[14] J. Gadge and A. A. Patil, "Port scan detection", in Proc. 16th IEEE International Conference on Networks, New Delhi, India, 2008, pp. 1–6.

[15] L. Aniello, G. Lodi, and R. Baldoni, "Inter-Domain Stealthy Port Scan Detection through Complex Event Processing", in Proc. 13th European Workshop on Dependable Computing, Pisa, Italy, 2011, pp. 67–72.

[16] snort.org, "SNORT - The Open Source Network Intrusion Detection System", [Online] Available at: http:www.snort.org [retrieved: October, 2022].

[17] B. Caswell, J. Baele, and A. Baker, "Snort Intrusion Detection and Prevention Toolkit (English Edition)", Amsterdam, Netherlands, 2007, p. 193.

[18] The Snort Project, "Snort User Manual", 2020, pp.182-185 [Online] Available at: https://docs.broadcom.com/doc/istr-24-2019-en [retrieved: October, 2022].

[19] S. K. Mukhiya and U. Ahmed, "Hands-On Exploratory Data Analysis with Python", Birmingham, United Kingdom, 2020.

[20] T. M. Cover and J. A. Thomas, "Elements of Information Theory", Hoboken, New Jersey, USA, 2005, pp. 29-30.

[21] C. E. Shannon, "A mathematical theory of communication", The Bell System Technical Journal, Vol. 27, 1948, pp. 379-423.

[22] A. Burkov, "The hundred-page machine learining book", Quebec City, Canada, 2019, p. 44.

[23] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Re "Snorkel: rapid training data creation with weak supervision", in Proc. 44th International Conference on Very Large Data Bases, Rio de Janeiro, Brazil, 2017, pp. 269–282.

[24] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", in Proc. 3rd International Conference on Learning Representations, San Diego, CA, USA, 2015, pp. 1-14.

[25] H. Rajeev and U. Devi, "Detection of Credit Card Fraud Using Isolation Forest Algorithm", In: G. Ranganathan, R. Bestak, R. Palanisamy, and A. Rocha, "Pervasive Computing and Social Networking", Lecture Notes in Networks and Systems, Vol. 317, Singapore, 2022.

[26] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation Forest", in Proc. Eighth IEEE International Conference on Data Mining, Pisa, Italy, 2009, pp. 413–422.

[27] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation-Based Anomaly Detection", ACM Transactions on Knowledge Discovery from Data, Vol. 6, 2012, pp. 1-39.

[28] Y. Zheng, C. Yang, and A. Merkulov, "Breast Cancer Screening Using Convolutional Neural Network and Follow-up Digital Mammography", in Proc. SPIE Commercial + Scientific Sensing and Imaging, Orlando, Florida, United States, 2018, p. 8.

[29] J. Ostner, "Usage of Image Classification for Detecting Cyber Attacks in Smart Home Environments", in Proc. Regensburger Applied Research Conference 2020 (RARC 2020), Regensburg, Germany, 2020, pp. 37-43.

A New Secure Publication Subscription Framework with Multiple Arbitrators

Shugo Yoshimura
Graduate School of Information Sci. and Electrical Eng., Kyushu Univ.
Fukuoka, Japan
yoshimura.shugo.822@s.kyushu-u.ac.jp

Kouki Inoue
Graduate School of Information Sci. and Electrical Eng., Kyushu Univ.
Fukuoka, Japan
inoue.kouki.882@s.kyushu-u.ac.jp

Dirceu Cavendish
Graduate School of Eng. Kyushu Institute of Tech.
Iizuka, Japan
cavendish@ndrc.kyutech.ac.jp

Hiroshi Koide
Research Institute of Info. Tech., Kyushu Univ.
Fukuoka, Japan
koide@cc.kyushu-u.ac.jp

***Abstract*— In this study, to make it easy for everyone to distinguish the right information from the wrong information, we suggest a new framework (Secure Publication Subscription Framework) that defines the reliability of publishers and provides it to subscribers. Nowadays, services like blogs and social media make available large amounts of information easily. On the other hand, there is a lot of unreliable information on the Internet. It is difficult to distinguish between true and false information. This problem is known as fake news and has become a serious problem. To solve this problem, we suggest a new framework for publishers and subscribers. The framework allows subscribers to easily confirm the authenticity of information by registering publishers and subscribers, and tracking publishers' reputation via a reputation score, guaranteeing the quality of the information that subscribers view. In this study, we show a proof of concept of a simple Secure Publication Subscription Framework and confirm that it is possible to implement a framework with the proposed functionality. We also confirm that the reputation score can be used as an indicator of the reliability of the information by using 1000 randomly generated articles within the framework. In addition, We also proposed three models of how to incorporate multiple Arbitrators to be considered when realizing this framework.**

Keywords-dissemination; publication; social networking; authenticity of information; reputation score.

I. Introduction

In our previous research [1], we proposed a Secure Publication Subscription Framework that allows subscribers to easily confirm the authenticity of the information and provides the publisher's reputation score. It consists of three parts, Publisher, Arbitrator, and Subscriber. The Subscriber can request the information challenge to the Arbitrator, and the Arbitrator verifies data truthfulness. A reputation score describes the Publishers' truthfulness and is increased or decreased according to the authenticity of the Publishers' information. We conducted experiments to confirm that the reputation score can be an indicator of the reliability of the Publishers. In this paper, we also include a model with multiple Arbitrators, considering the construction of a practical system. We propose three models for setting multiple Arbitrators. The merits, demerits, and conditions under which they should be used are discussed for each mechanism, reinforcing the realism of this framework.

In recent years, Internet technologies have made great progress, with the population of Internet users increasing rapidly. Thanks to services like blogs and social media, anyone can get a large amount of information easily. Nowadays, we can be aware of what is happening around the world, no matter where we are.

On the other hand, there is a lot of unreliable information on the Internet. It is difficult to distinguish between true and false information. This problem is known as fake news and has become a serious problem. Fake news is fabricated information that mimics news media content in form but not in organizational process or intent [2]. It is not just a prank, but a serious problem. As an example, during the 2016 United status presidential election, fake news was highly used and had a big impact on Twitter [3] [4].

To solve this problem, we suggest a new framework for publishers and subscribers. This framework allows subscribers to easily confirm the authenticity of information by registering publishers and subscribers, guaranteeing the publisher of the information that subscribers view, checking the information challenge from subscribers, and providing the publisher's reputation score that increases or decreases as a result of the authenticity of the information.

This framework consists of three parts, Publisher, Subscriber and Arbitrator. The main role of the Publisher is publishing articles or news. The Subscriber registers with the Publisher and subscribes for publications. The Arbitrator provides the Publisher's reputation and verifies the information challenge from the Subscriber.

The paper is organized as follows. Related work is in-

This study is supported by JSPS KAKENHI Grant Number 21K11888 and Hitachi Systems, Ltd.

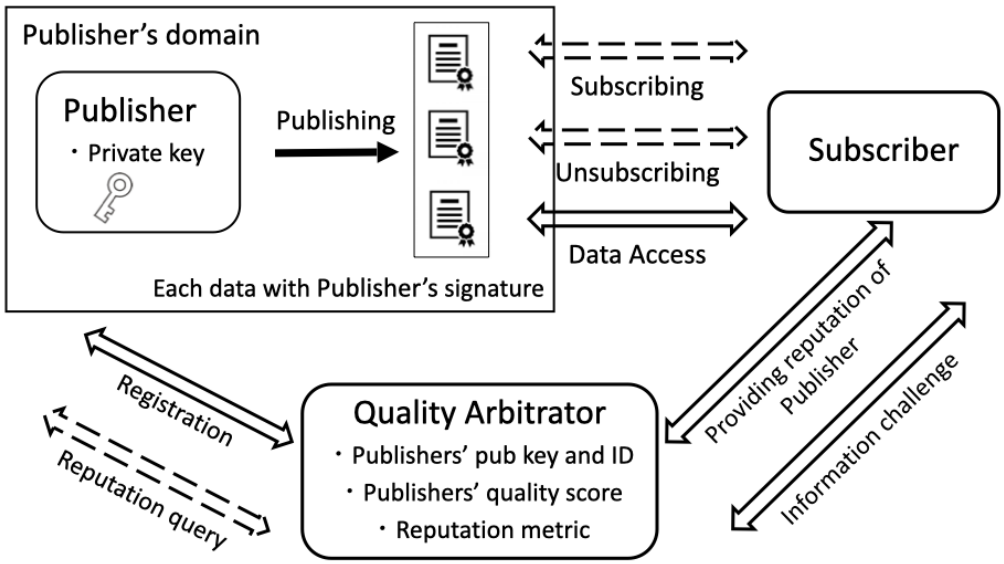


Figure 1. Secure Publication/Subscriber Architecture

cluded in Section II. Section III describes our proposed secure publication/subscription reference model. Section IV describes a proof of concept implementation of the reference model. Section V describes two experiments used to track the performance of the proposed publication/subscription model. Section VI presents the performance results and discussions. Section VII proposed three models for how to incorporate multiple arbitrators to be considered when implementing this framework, and discusses the advantages and disadvantages of each model. Section VIII summarizes our studies and addresses directions we are pursuing as follow up to this work.

II. Related work

Previous research on publication/subscription systems have covered various areas, such as security, confidentiality and scalability.

Nakamura and Enokido [5] focused on a peer to peer publication/subscription model where multiple topics are supported. In that work, they propose a subscription initialization protocol to ensure that peers not authorized to have access to topics do not have access to them. They do not address the quality of the information exchanged within topics. In contrast, our framework addresses information quality on a generic publication/subscription architecture, not necessarily requiring a peer to peer model.

Salem [6] addresses the problem of authenticating users of a pub/sub system containing a message broker in a privacy-preserving way. The proposal supports mutual authentication in a scalable way, and may be adopted by pub/sub systems with a broker. In contrast, our work does not focus on anonymity of publishers/subscribers, although our pub/sub model could be adapted to include a broker, if necessary.

In Srivatsa [7], a secure event dissemination protocol is proposed where encryption and authorization keys are used on top of an IP network that does not provide confidentiality nor integrity of data. In contrast, although our pub/sub model supports integrity verification of data, our focus is on the control of the quality of data published.

Bovet and Makse [4] describe an information ranking mechanism to fight unreliable (spam) data in a pub/sub system model with a broker reference architecture. They propose to rank information as a way to avoid blacklisting. However, their ranking system is still based on participants' voting. Although the purpose of the research is similar to ours, our solution to control quality of disseminated data is based on an arbitrator that is supposed to be able to verify data quality on specific domains, rather than relying on voting.

III. Secure Publication/Subscription

This section describes the operation of the Secure Publication Subscription Framework in detail.

Figure 1 describes our proposed secure publication/subscription system architecture. Multiple publishers provide signed data contents to consumers, or subscribers. Data content quality is tracked by an independent quality arbitrator. The quality arbitrator provides publishers' reputation to subscribers. Also, the arbitrator may receive data truthfulness challenges from subscribers.

A. Sec Pub/Sub Components

Figure 2 illustrates how Publishers provide signed data contents. Publishers also produce a digest of the data content using standard asymmetric cryptography, using their private key to ensure data integrity.

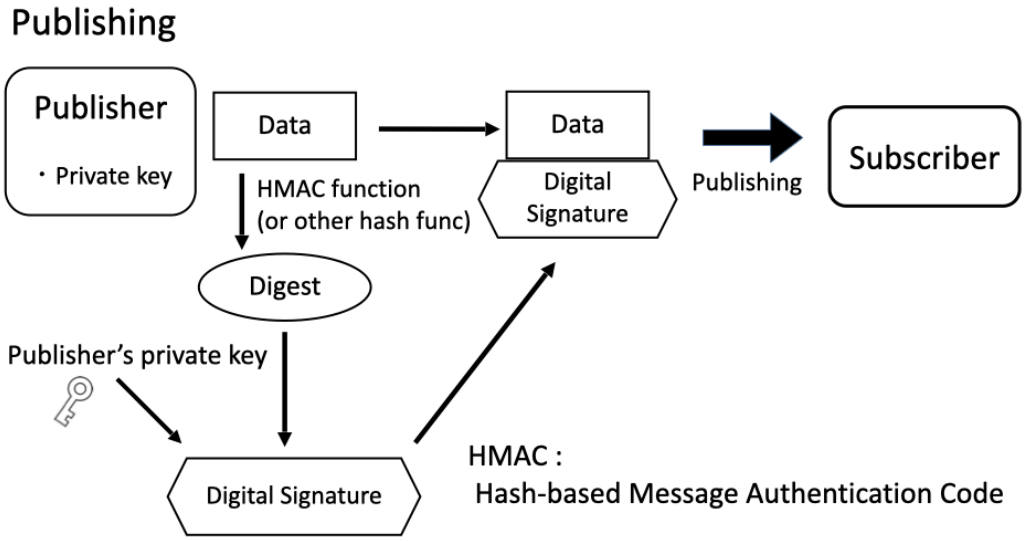


Figure 2. Signed publishing

Figure 3 illustrates publisher/subscriber interfaces. The subscriber requests subscription services from a publisher and receives the publisher public key used to verify data authenticity. Once the subscription service has been agreed upon, an information retrieval interface is used to request signed data from the publisher.

Figure 4 illustrates the subscriber's data processing of published data. Data processing includes data integrity verification and confirmation authorship. The subscriber verifies the digital signature and the digest of the data, using the publisher public key. In this process, the subscriber verifies the integrity of the received data and confirms the data's authorship.

Figure 5 illustrates publisher reputation tracking feature of the secure pub/sub framework. Each publisher registers first with the quality arbitrator, upon which its public key is passed to the arbitrator. The arbitrator then tests the publisher's possession of the corresponding private key as part of the registration. Each successfully registered publisher is associated with a reputation score metric, which can be queried by both the publisher itself as well as subscribers.

Figure 6 illustrates the subscriber/quality arbitrator interfaces. Subscribers can request publisher's reputation score from the arbitrator. In addition, subscribers can challenge publisher's trustfulness for each data received. The quality arbitrator, upon receiving the challenge, verifies data truthfulness, and adjusts the publisher reputation score according with data verification status.

B. Reputation Algorithm

The reputation score of a publisher is defined as

$$score = \frac{the\ number\ of\ correct\ data}{the\ number\ of\ all\ published\ data}.$$

However, as the quality arbitrator may not estimate correctly every and all data published, we introduce a noise model for data verification, as shown in Figure 7. In the model, p is the probability that a true piece of data be recognized as false, whereas q represents the probability of a false piece of information be admitted as true. In the experimental section, we exemplify the arbitrator score reputation tracking on two publisher scenarios: i- trusted publisher (all data is truthful); ii- untrusted publisher; Publisher produces up to 1000 data pieces (the data can be right or wrong).

IV. Implementation

In this section, we describe an overview of the implementation of Publisher, Arbitrator, Subscriber. We implemented the Publisher and the Arbitrator with Node.js and Express that is a JavaScript Web framework, and we implemented the Subscriber with Python3. The Publisher and the Arbitrator operate like a Web server, independently, and the Subscriber accesses them according to the scenarios. The versions used in the implementation are summarized in Table I.

TABLE I
Implementation

Application	Version
Node.js	12
MySQL	5.7
Python	3.9.12

A. Publisher

The Publisher is implemented with Node.js and Express, and it operates as a Web server. Figure 8 describes the im-

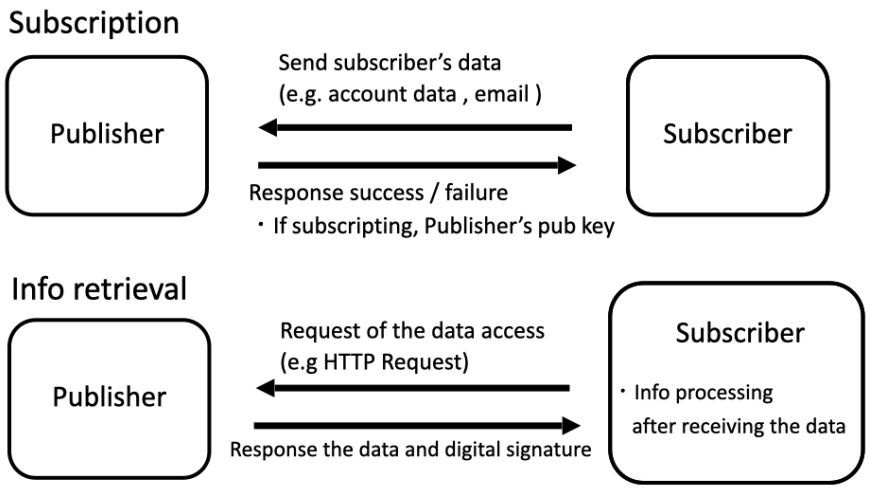


Figure 3. Subscription and Information Retrieval

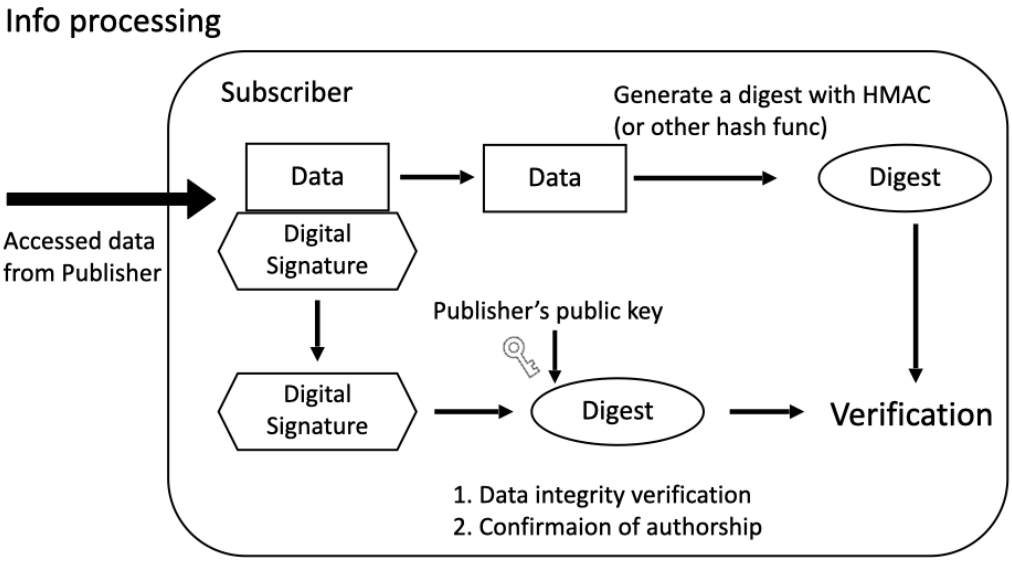


Figure 4. Data Integrity Verification

plementation. The Publisher has subscriber registration, login, some data pages and digital signatures. In addition, it has a MySQL database that saves the Subscriber's name and hashed password. If it receives an HTTP Request from the Subscriber, it replies with an HTTP Response and sends the data.

B. Arbitrator

The Arbitrator is also implemented with Node.js and Express, and operates as a Web server. Figure 9 describes the implementation of the Arbitrator. The Arbitrator receives the Publisher's registration, reputation query, as well as information challenge and request for publisher's public key. Additionally, the Arbitrator supports a MySQL database, which saves the Publisher's name, password, public key and Publisher reputation score. Firstly, the Publisher registers its name, password and public key. In our experiment scenarios, the Publisher's information is saved in initial state, so this step is omitted. If the Subscriber requests the Publisher's public key, the Arbitrator responds to it. If the Subscriber requests the Publisher's reputation score, the Arbitrator sends the Publisher's score. If the Arbitrator receives an information challenge from the Subscriber, it verifies data truthfulness, updates the score of the Publisher.

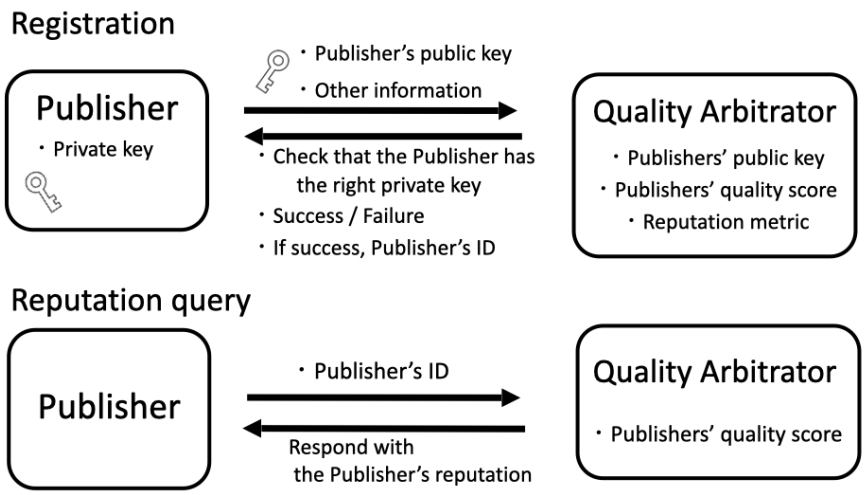


Figure 5. Publisher registration and Reputation Tracking

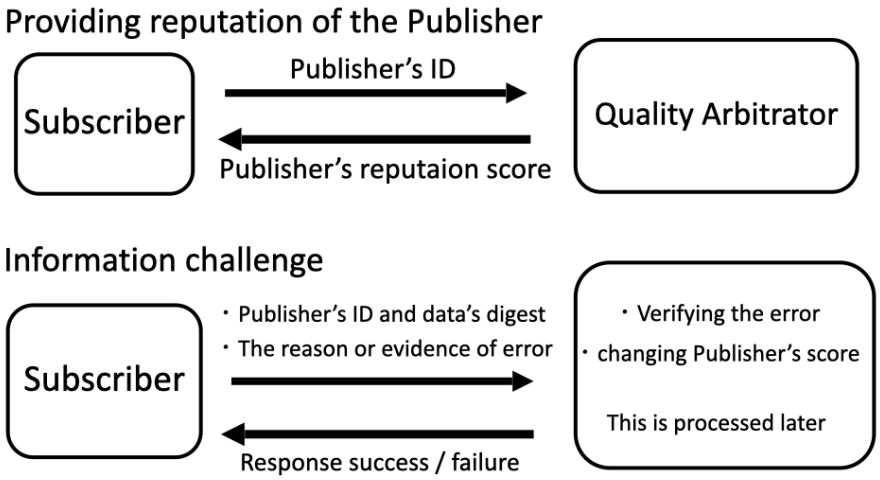


Figure 6. Reputation service interface

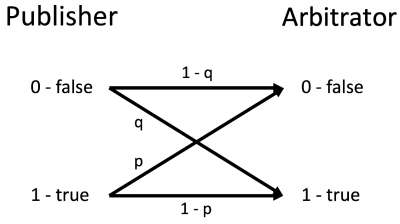


Figure 7. Noisy Channel Model

C. Subscriber

The Subscriber is implemented with Python3. It accesses the Publisher and the Arbitrator according to the different scenarios. During information processing, it verifies the integrity of received data and confirms data authorship (Figure 10).

V. Experiment

This section demonstrates the evolution of the reputation estimator and reputation score for the Secure Publication Subscription Framework using 1000 randomly generated true and false data.

The resulting graph shows 3 lines:

- Actual reputation score: the reputation score actually obtained after going through the Secure Publication Subscription Framework,
- Expected reputation score: the expected value of the reputation score obtained from the actual truth of the data, p and q,

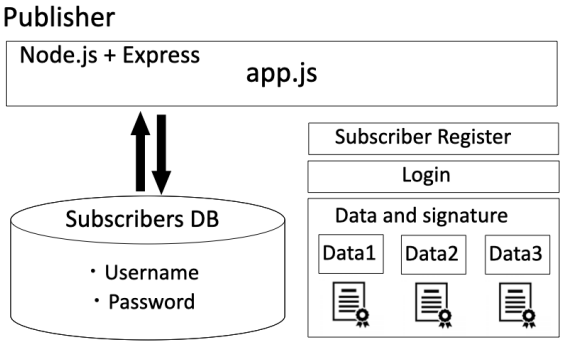


Figure 8. Publisher

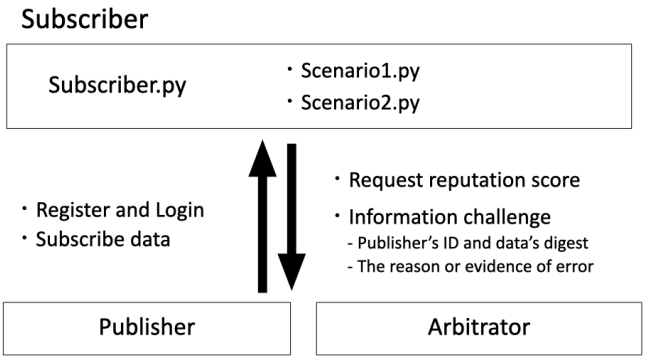


Figure 10. Subscriber

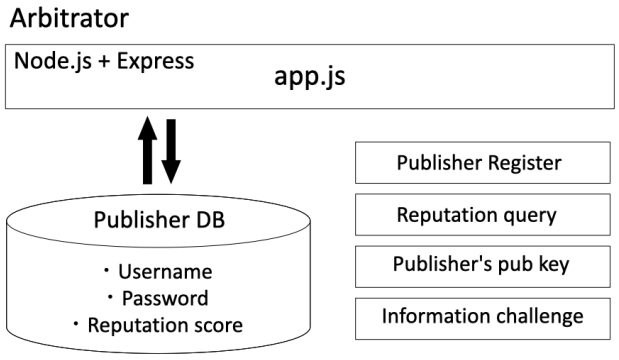


Figure 9. Arbitrator

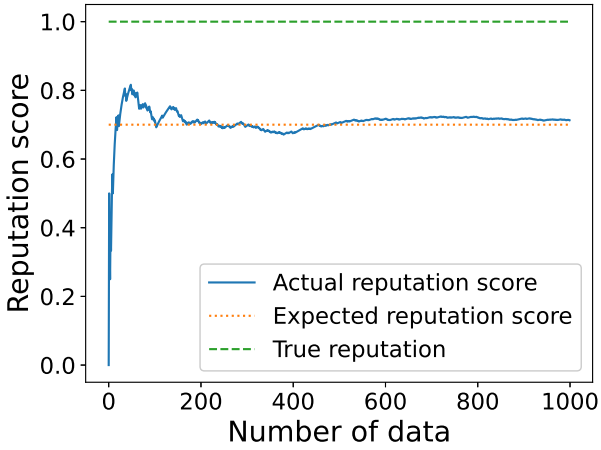


Figure 11. scenario 1

- True reputation: proportion of data that is actually true.

We illustrate the secure publication/subscription model with the following scenarios:

A. Scenario 1

1) Subscribers register and login in with the Publisher
2) Subscribers subscribe to data from the Publisher
3) Subscribers retrieve the data
4) Subscribers send a query about the Publisher's reputation to the Arbitrator

In Scenario 1, the credibility of the Publisher's data is 100%, hence the Publisher's true reputation is 1. However, the expected reputation score is

$$1 - p$$

because there is a possibility that the Arbitrator will judge it to be false. In this experiment, the values of the p and q are set to 0.3 to check the reputation scores. To show that the accuracy of the reputation score does not drop even if the accuracy of the true/false discrimination is not so high, p and q were set to fairly low values. We think that there is still room for further study on this value.

Figure 11 shows the graph of the results for Scenario 1.

B. Scenario 2

In scenario 2, Publisher's data is not always true.

1) Subscribers register and login in with the Publisher
2) Subscribers subscribe to data from the Publisher
3) Subscribers retrieve the data
4) Subscribers issue an information challenge
5) The Arbitrator decides the data as false, and updates the Publisher's reputation
6) Subscribers query the reputation of the Publisher from the Arbitrator

Let a be the probability that the publisher's data is false. Then, the expected value of the true reputation is

$$1 - a,$$

while the expected reputation score is

$$a * q + (1 - a) * (1 - p).$$

In Scenario 2, step 1, 2, 3 are the same as in Scenario 1. However, the Subscriber carries out an information challenge in steps 4 and 5. The probability of judging the data to be

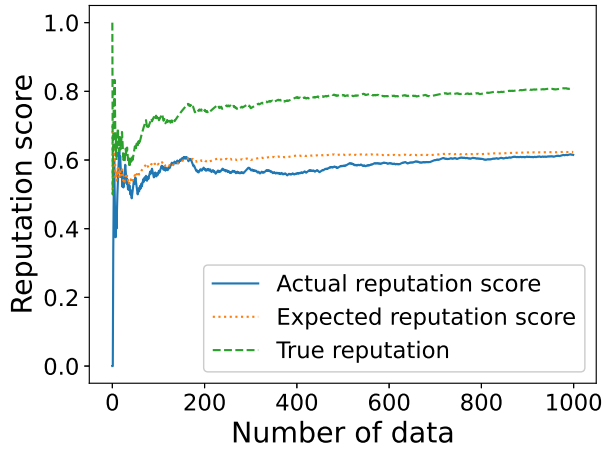


Figure 12. scenario2 data accuracy = 0.8

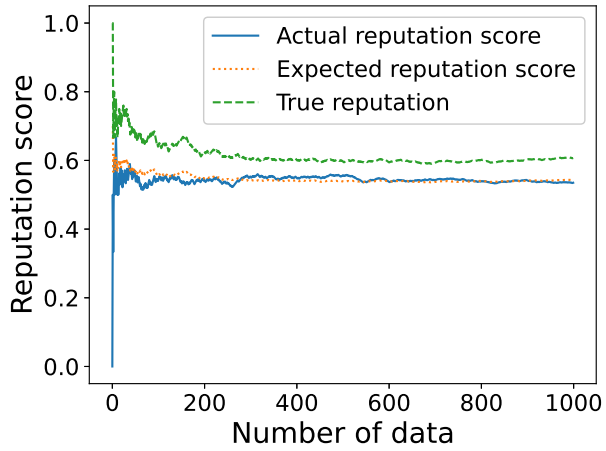


Figure 13. scenario2 data accuracy = 0.6

correct was varied between 0.8 and 0.6, and p and q were 0.3 to check the reputation scores for each case.

The experimental results are shown in Figures 12 and 13.

VI. Performance Analysis

In this section, we present the reputation tracking results of our secure pub/sub system. In scenario 1, the final three scores obtained from the 1000 data points are shown in Table II.

TABLE II
SCENARIO 1

Actual reputation score	0.713
Expected reputation score	0.700
True reputation	1.000

In scenario 2, the final three scores obtained from the 1000 data points are shown in Tables III and IV.

From these experimental results, with a sufficient number of data points and a certain degree of accuracy in determining the truth of the data, we see that the actual reputation score converges to the expected reputation score.

TABLE III
SCENARIO 2 DATA ACCURACY = 0.8

Actual reputation score	0.615
Expected reputation score	0.623
True reputation	0.808

TABLE IV
SCENARIO 2 DATA ACCURACY = 0.6

Actual reputation score	0.535
Expected reputation score	0.543
True reputation	0.607

Moreover, we use a noise model for data verification, and we define the expected reputation to be

$$a * q + (1 - a) * (1 - p).$$

So, if p and q are known, the Publisher's true reputation can be estimated from the actual score.

These results indicate that the reputation score is closely related to the probability of the correctness of the data (credibility) and that the actual reputation score can be calculated with considerable accuracy if p and q are known.

The result shows that the reputation score is a sufficiently reliable value for easily confirming the credibility of the Publisher.

VII. Incorporation of Multiple Arbitrators

Although we were able to confirm that the reputation score is related to the credibility of the publisher in the proposed framework, there are still some problems to be solved in actual operation. One of the problems is that it is not realistic for a single arbitrator to handle all of the enormous amounts of info challenges. To solve this problem, multiple Arbitrators can be used instead of a single Arbitrator to perform fact-checking. However, there are various problems associated with this method, such as the sharing of secret keys and reputation scores.

In this section, we propose three mechanisms for setting up multiple Arbitrators. The merits, demerits, and conditions under which they should be used are discussed for each mechanism.

A. Basic method

In this model, each arbitrator maintains the same database that contains the data of all the Publishers, and it is necessary to rewrite the information in the database in case of registration of a Publisher, information challenge from a Subscriber, etc. while synchronizing with the other Arbitrators. The overall diagram is shown in Figure 14. The explanation is based on the case of two Arbitrators, but the same operation can be performed even if the number of Arbitrators is larger.

The operation of Publisher registration is as follows.

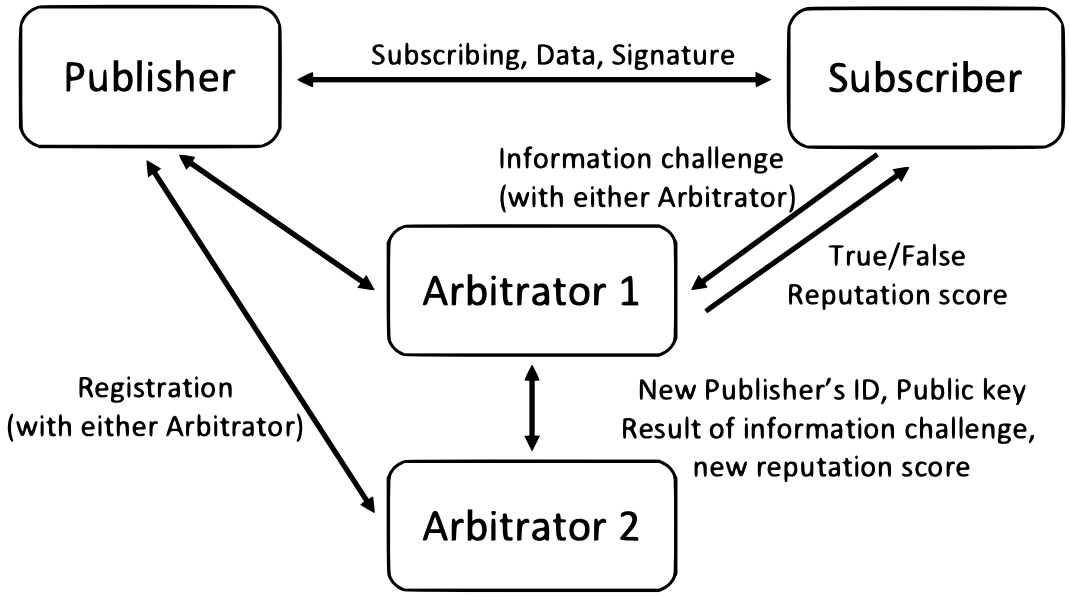


Figure 14. basic method

1) The Publisher selects one of the two Arbitrators and sends its public key and other information.
2) The selected Arbitrator verifies the key. If the key is invalid, it sends a message to the Publisher and terminates the operation.
3) If the key is OK, it shares the public key and other information with the other Arbitrator and updates the database.
4) The Publisher is notified that the registration has been completed.

This is how it works in the case of an information challenge.

1) The Subscriber selects one of the two Arbitrators to perform the information challenge.
2) The selected Arbitrator verifies the signature and performs a fact check.
3) The result of the fact check and the new reputation score is shared with the other Arbitrator, and the database is updated.
4) The results of the fact check and the new reputation score are sent to the Subscriber.

The advantage of this model is redundancy. If one Arbitrator becomes unavailable, another Arbitrator can be substituted and the entire system will not become unavailable. This model is suitable when availability at any time is important.

There are three possible disadvantages of this model.

- The application address for information challenge by the Subscriber when the Publisher registers
  In the past, there was only one Arbitrator, so there was no need to worry about where to submit applications, but in this model, there are two Arbitrators, so the Publisher and Subscriber must choose one or the other, or submit to both.
- Sharing of publisher information and reputation score
  For example, if one of the Arbitrator performs an information challenge and the reputation score of the Publisher changes, the other Arbitrator will be notified that the information challenge was performed and that the Publisher's reputation score has changed. The results of the information challenge and the new reputation score need to be shared with the other Arbitrator. When updating the database is necessary, it must be handled in such a way that it does not cause errors in the synchronization process.
- Sharing of publisher information and public keys
  Arbitrator needs to verify whether an article is written by the correct Publisher at the time of information challenge. Therefore, all Arbitrators must maintain the IDs and public keys of all Publishers, which is inefficient.

B. Combination of specific Arbitrator and Publishers

This model is a method that eliminates the need to share reputation scores and keys with other Arbitrators by linking the Publisher to a specific Arbitrator. The overall diagram is shown in Figure 15.

In this model, the Publisher selects which Arbitrator he/she belongs to and applies for registration to that Arbitrator. In addition, when making an information challenge, the Subscriber must send it to the Arbitrator to which the Publisher of the article belongs. Therefore, it is necessary to indicate which Arbitrator the Publisher belongs to in the article. In this model, Arbitrator 1 and Arbitrator 2 have different databases. Each Arbitrator keeps information only on the Publishers who belong to the respective Arbitrator.

The advantage of this model is that the load on the Arbitrator is well distributed. This makes it suitable for large-

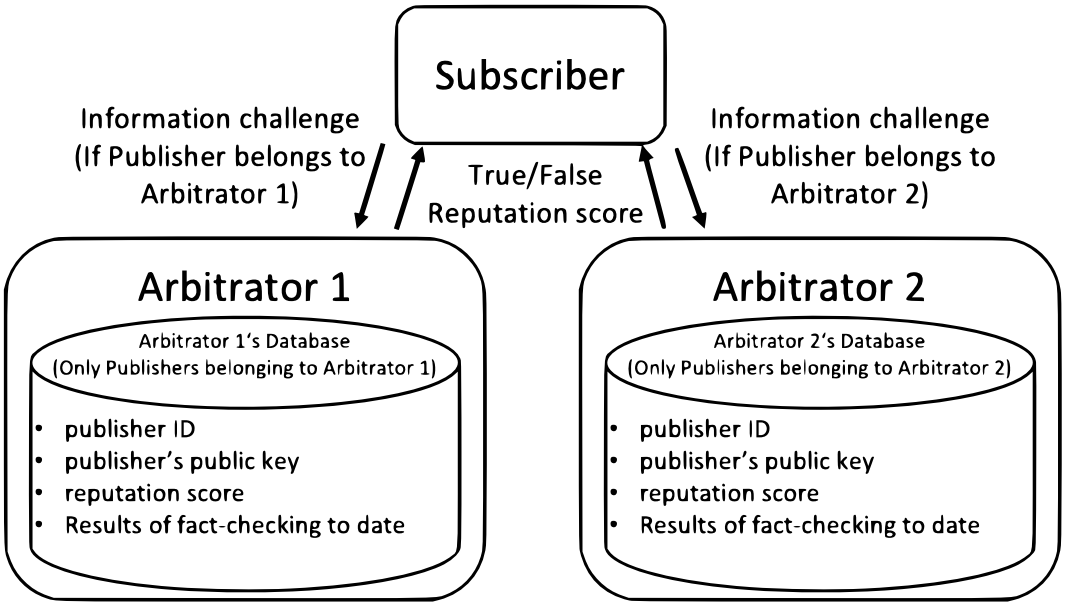


Figure 15. Combination of specific Arbitrator and Publishers

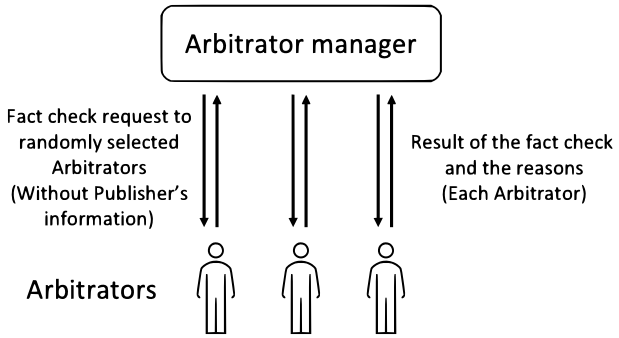


Figure 16. Arbitrator manager model

scale systems. The accuracy of the information challenge can be improved because the Publisher can select the appropriate Arbitrator according to the field of expertise and language used.

The disadvantage of this model is that although there are multiple arbitrators, only one arbitrator performs fact-checking, which may bias the judgment of credibility. It also has no redundancy. Therefore, it is not suitable for cases where the accuracy of fact-checking or stable availability at any time is important.

C. Arbitrator manager model

This model sets an "Arbitrator manager" that accepts access from publishers and subscribers. Arbitrator manager takes requests such as registration from Publishers, information challenge from Subscribers, confirmation of reputation score, etc. Only the fact check required for the information challenge is requested and distributed to multiple Arbitrators. The public key of the Publisher, reputation score, ID, and other information are kept by the Arbitrator manager. The configuration is shown in Figure 16.

Information challenge in this system is performed as follows.

1) The arbitrator manager receives information challenge from the Subscriber.
2) The Arbitrator manager verifies the signature and verifies that it is the correct Publisher.
3) The arbitrator manager requests a fact check from randomly selected arbitrators (the number of arbitrators is arbitrary).
4) Each arbitrator performs fact-checking and returns the results and reasons to the arbitrator manager.
5) The Arbitrator manager compiles the results of all fact-checking, returns the results to the Subscriber, and updates the Reputation score.

In step 3, the number of arbitrators to request fact checks can be considered according to the situation. Using numerous arbitrators may improve credibility, but it also increases the time and cost. In addition, the method of selecting Arbitrators could be not only random but also selecting appropriate Arbitrators according to their expertise in the language or field of study.

In step 5, there are several possible ways to compile the results of all fact checks. One is to simply ask how many people perform fact checks and reflect the number of people who judged the results to be true in the reputation score, another is to adopt the result of a majority vote, and another is to use a majority vote, but if the number of true/false votes are close, the final decision is made by the Arbitrator manager.

The advantages of this model are that the Subscriber does

not need to select an Arbitrator, but only needs to access the Arbitrator manager, that there is no problem of sharing and managing the Publisher's key and reputation score among multiple Arbitrators, and that the Arbitrator manager can make the final decision when there are multiple Arbitrators. In addition, the Publisher's key and reputation score can be shared and managed among multiple Arbitrators, and Arbitrators can be easily added or deleted. Therefore, this model is suitable for cases where high accuracy is important by having multiple Arbitrators perform fact-checking.

The disadvantage of this model is that it does not distribute the load of the Arbitrator manager itself and does not have redundancy. Therefore, it is not suitable for large-scale systems.

VIII. Conclusion and Future work

In this study, we proposed a new framework (Secure Publication Subscription Framework) that allows subscribers to check the accuracy of information based on the authenticity of the publisher's historical data by checking the reputation score. In this framework, subscribers can check the reputation score of the publisher and challenge data reliability if the information is suspected to be unreliable. We also conducted experiments on the publisher's reputation score, and found that the actual reputation score approximates the expected value calculated from the probability of correctly judging the reliability of information. In the actual operation of this framework, it will be necessary to incorporate multiple Arbitrators from the aspect of load distribution, etc. We have shown three applicable methods to support multiple Arbitrators, and discussed their technical feasibility. Each of them has different merits and can be applied to various situations.

The development of the Internet and social media has made it very convenient for anyone to easily disseminate information, but it has also caused a major problem: fake news. However, there is so much information that we see every day that it is practically difficult to check all of it to make sure it is not fake news. Moreover, some of the information is highly specialized and cannot be confirmed as true or false even if it is carefully read. Therefore, we believe that there is a demand for a framework that allows anyone to easily verify whether a Publisher is impersonating someone else, and to confirm the authenticity of that Publisher.

As future research, integration of AI(Artificial Intelligence) algorithms to automatically identify fake news with expert arbitrators is a promising path. Although the accuracy of discriminating fake news has been a challenge for AI technologies, our expert framework can aid by using AI algorithms to improve false positives/negatives. Combined with these technologies, we believe that a robust data reliability framework for publication/subscription platforms can emerge.

There are still some minor problems. For example, in the current reputation score algorithm, the score of publishers who publish a small number of articles is rated higher than the actual credibility of the articles. This problem can be improved by setting the score lower when the number of articles is below a certain level. We believe that improving the specification of these details will make this framework more realistic.

References

[1] S. Yoshimura, K. Inoue, D. Cavendish, and H. Koide, "Secure Publication Subscription Framework for Reliable Information Dissemination." *The 2022 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications*, 2022.

[2] D. M. J. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, M. Schudson, S. A. Sloman, C. R. Sunstein, E. A. Thorson, D. J. Watts, and J. L. Zittrain, "The science of fake news." *Science*, pp. 1094–1096, 2018.

[3] N. Grinberg, K. Joseph, L. Friedland, B. Swire-Thompson, and D. Lazer, "Fake news on Twitter during the 2016 US presidential election." *Science*, pp. 374–378, 2019.

[4] A. Bovet and H. A. Makse, "Influence of fake news in Twitter during the 2016 US presidential election." *Nature communications*, pp. 1–14, 2019.

[5] S. Nakamura, T. Enokido, and M. Takizawa, "Subscription Initialization (SI) Protocol to Prevent Illegal Information Flow in Peer-to-Peer Publish/Subscribe Systems," *19th International Conference on Network-Based Information Systems*, pp. 42–49, Sept. 2016.

[6] F. M. Salem, "A Secure Privacy-Preserving Mutual Authentication Scheme for Publish-Subscribe Fog Computing," *14th International Computer Engineering Conference*, pp. 213–218, Dec. 2018.

[7] M. Srivatsa and L. Liu, "Secure Event Dissemination in Publish-Subscribe Networks," *27th International Conference on Distributed Computing Systems (ICDCS '07)*, pp. 22–22, June 2007.

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.

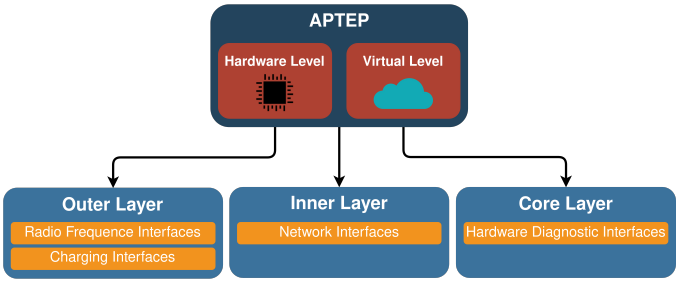


Fig. 1. APTEP Architecture

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

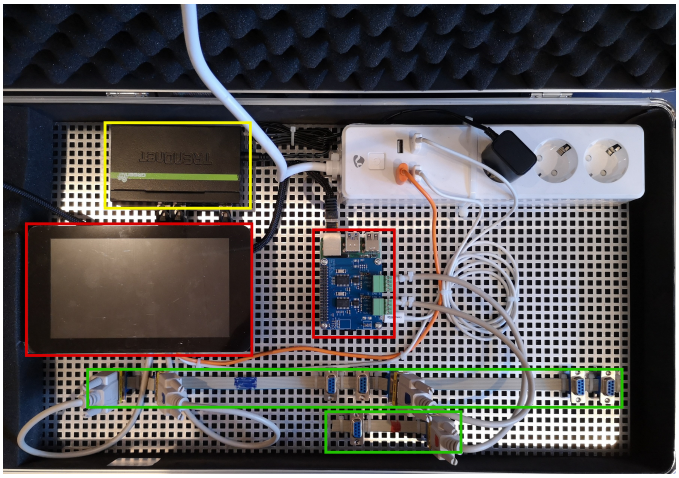


Fig. 2. ANSKo Hardware

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.

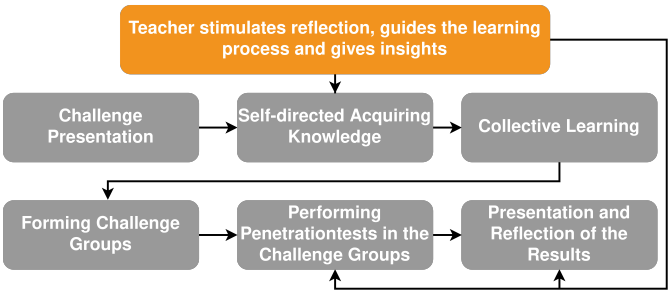


Fig. 3. Learning Concept

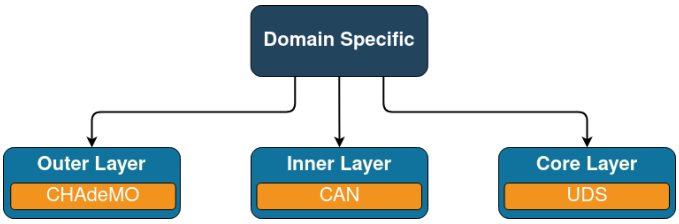


Fig. 4. Domain-specific Challenge

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

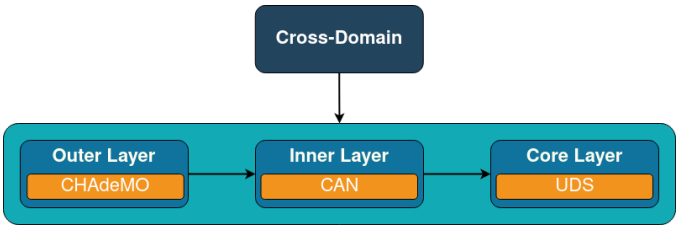


Fig. 5. Cross-domain Challenge

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).

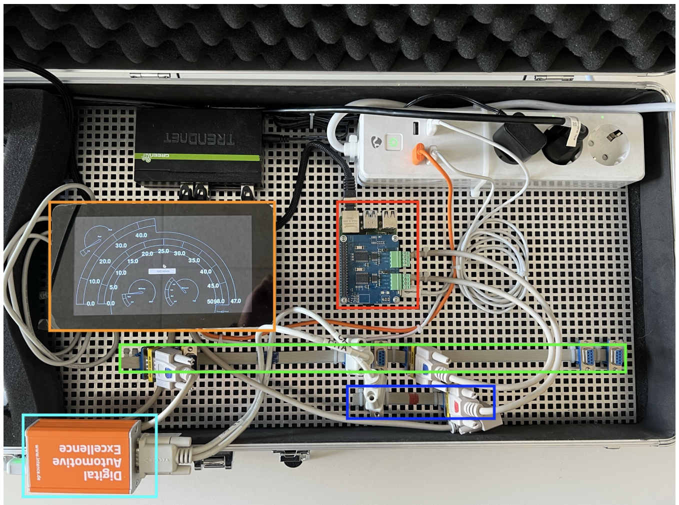


Fig. 6. ANSKo MITM Setup

- **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.

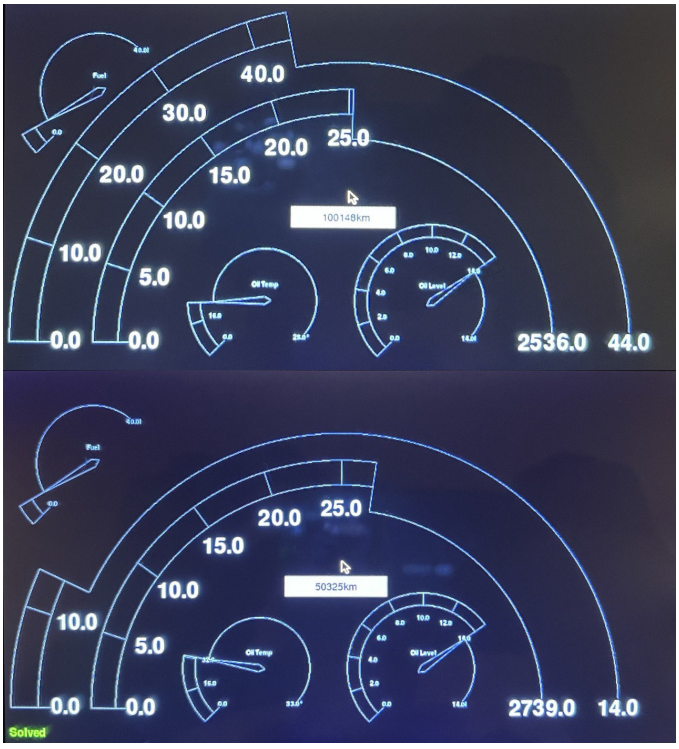


Fig. 9. ANSKo Tachometer (Top: Before; Bottom: After)

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.

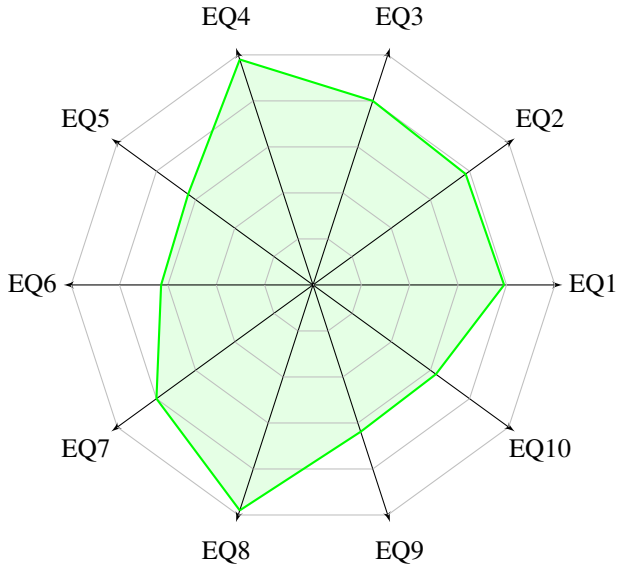


Fig. 10. Evaluation of the ANSKo

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).

Secure Authorization for RESTful HPC Access with FaaS Support

Christian Köhler
Computing
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen
Göttingen, Germany
E-Mail: christian.koehler@gwdg.de

Mohammad Hossein Biniaz
Computing
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen
Göttingen, Germany
E-Mail: mohammad-hossein.biniaz@gwdg.de

Sven Bingert
eScience
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen
Göttingen, Germany
E-Mail: sven.bingert@gwdg.de

Hendrik Nolte
Computing
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen
Göttingen, Germany
E-Mail: hendrik.nolte@gwdg.de

Julian Kunkel
Computing
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen/Universität Göttingen
Göttingen, Germany
E-Mail: julian.kunkel@gwdg.de

***Abstract*—The integration of external services, such as workflow management systems, with High-Performance Computing (HPC) systems and cloud resources requires flexible interaction methods that go beyond the classical remote interactive shell session. In a previous work, we proposed the architecture and prototypical implementation of an Application Programming Interface (API) which exposes a Representational State Transfer (REST) interface that clients can use to manage their HPC environment, transfer data, as well as submit and track batch jobs. In this article, we expand on this foundation by including a full Function as a Service (FaaS) interface which allows it to be a drop-in replacement for functions with high resource demands. In order to enable automated processes without any manual interaction while maintaining the highest security standards, a fine-grained role-based authorization and authentication system which facilitates the initial setup and increases the user's control over the jobs that services intend to submit on their behalf is presented. The developed *HPCSerA* service provides secure means across multiple sites and systems and can be utilized for one-off code execution and repetitive automated tasks, while adhering to the highest security standards.**

***Keywords*—HPC; RESTful API; OAuth; authorization; FaaS.**

I. Introduction

This work is an extension of [1] by the inclusion of the *Function as a Service* (FaaS) idiom, which becomes increasingly popular due to several advantages, including the cost effectiveness, fault tolerance, and ease of use. However, there are usually strict limitations on the execution time of a function, resource requirements and the size of input and output data [2]. Driven by the large success of data- and compute-intensive methods, there is an increasing demand for computing power in various scientific domains which are outside of these limits. Historically, HPC systems were used to satisfy those requirements in a cost-effective manner. Meanwhile it is similarly attractive to use a RESTful interface to easily deploy preconfigured functions and use those in an automated setup. This has led to the creation of different services which for instance expose a RESTful API with which users can remotely interact with an HPC system. There are numerous different use cases for such a requirement. One motivating example can be the ability to manage complex and compute-intensive workflows with a graphical user interface to improve usability for inexperienced users [3].

While, on one hand, there are these efforts to ease and open up the use of HPC systems, there is, on the other hand, a constant threat by hackers or intruders. Since users typically interact with the host operating system of an HPC system directly, local vulnerabilities can be immediately exploited. Two of the most favored attacks by outsiders are brute-force attacks against a password system [4] and probe-based login attacks [5]. These attacks, of course, become obsolete if attackers can find easier access to user credentials. Therefore, it is of utmost importance to keep access, and access credentials, to HPC systems safe.

In this context, services easing the use of and the access to HPC systems should be treated with caution. For example, if access via Secure Shell (SSH) [6] to an HPC system is only possible using *SSH keys* due to security concerns, these measures are rendered ineffective if users re-establish a password-based authentication mechanism by deploying a RESTful service on the HPC system that is exposed on the internet. Observing these developments, it becomes obvious that there is a requirement to offer a RESTful service to manage data and processes on HPC systems remotely which is comfortable enough in its usage to discourage spontaneously concocted and insecure solutions built by inexperienced users with the main objective of "getting it to work", but which adheres to the **highest security standards**.

In order to prevent those security risks by users, HPC systems are increasingly secured, including a two-factor authentication (2FA) for SSH connections [7], which is a problem for automated workflows since they need to run without any manual interaction. In this paper we present *HPCSerA*, a REST API which is compatible to the FaaS idiom, therefore allowing clients to use it for large, data and compute-intensive functions similar to *OpenFaaS* [8]. In order to enable this functionality, a detailed security analysis was done and a secure authorization method was developed to enable automated data processing without any manual intervention, while maintaining a similar security standard compared to a SSH access secured by 2FA. The *key contributions* of this article are:

1) discussion of the FaaS usage model and its capabilities on HPC systems using HPCSerA;
2) analysis of possible attack scenarios based on a RESTful service running on an HPC system;
3) presentation of a user-friendly and secure authorization method inspired by OAuth;
4) discussion of the usability, including the resolution of complicated dependencies.

The remainder of this paper is structured as follows: In Section II, the related work is presented, including state-of-the-art mechanisms to solve this issue. This is followed by a presentation of the fundamental idea of HPCSerA and its three basic components in Section III. Based on this, the FaaS functionality is discussed in Section IV. In Section V, existing security issues preventing a wide-spread application of HPCSerA are being discussed and an improved architecture with a security-based scope definition is presented. In the following Section VI, our implementation is presented. At the end, a diverse set of use cases are presented in Section VII, as well as a concluding discussion, which is provided in Section VIII.

II. Related Work

There is without question a general trend towards remote access for HPC systems, for instance in order to use web portals instead of terminals [9]. These applications actually have a long-standing history with the first example of a web page remotely accessing an HPC system via a graphical user interface dating back to 1998 [10].

Newer approaches are the *NEWT* platform [11], which offers a RESTful API in front of an HPC system and is designed to be extensible: It uses a pluggable authentication model, where different mechanisms like Open-Authorization (OAuth), Lightweight Directory Access Protocol (LDAP) or *Shibboleth* can be used. After authentication via the */auth* endpoint, a user gets a cookie which is then used for further access. With this mechanism *NEWT* forwards the security responsibility to external services and does not guarantee a secure deployment on its own. This has the disadvantage that *NEWT* is not intrinsically safe, therefore providers of an HPC-system need to trust the provider of a *NEWT* service that it is configured in a secure manner. Additionally, no security taxonomy is provided, which is key when balancing security concerns and usability.

Similarly, *FirecREST* [12] aims to provide a REST API interface for HPC systems. Here, the Identity and Access Management is outsourced as well, in this case to *Keycloak*, which offers different security measures. In order to grant access to the actual HPC resources after successful authentication and authorization, an *SSH certificate* is created and stored at a the *FirecREST* microservice. Although this is a sophisticated mechanism, there seem to be a few drawbacks. First of all, the *sshd* server must be accordingly configured to support this workflow, secondly it remains unclear how reliable status updates about the jobs can be continuously queried when using short-lived certificates, and lastly these certificates needs to be stored at a remote location, which might conflict with the terms of service of the data center of the user. A similar approach is used by *HEAppE* [13] where the communication is between the API server and the HPC system is done via SSH. To do so, for each project an SSH key is managed by the API server. Users are not supposed to connect to the system via SSH at all. However, in order to upload data via secure copy users obtain a temporary SSH key. To manage the exposure of a possible data breach of the API server, the developers recommend to use one instance of HEAppE per HPC account.

Additionally, HPC systems are often configured to allow logins from a trusted network only, which means that the *FirecREST* microservice can not serve multiple HPC systems at a time.

While the *Slurm Workload Manager* provides a REST interface that exposes the cluster state and in particular allows the submission of batch jobs, the responsible daemon is explicitly designed to not be internet-facing [14] and instead is intended for integration with a trusted client. Its ability to generate JSON Web Token (JWT) tokens for authentication provides an interesting alternative route for interaction with our architecture, provided both services are hosted in conjunction. Clients that shall execute Slurm jobs authenticate the trusted Slurm controller via the *MUNGE* service [15] that relies on a shared secret between client and server. If either of these is compromised, then it is assumed that the whole cluster is insecure. Slurm can be deployed across multiple systems and administrative sites and there are various options for Slurm to support a meta-scheduling scenario or federation. However, if the Slurm controller is compromised, it can dispatch arbitrary jobs to any of the connected compute systems. In addition, decoupling the API implementation from the choice of the job scheduler, as we propose, allows interoperation of multiple sites, possibly using different schedulers.

An alternative execution model popular with public cloud systems is *Function as a Service* (FaaS). In this model, a platform for the execution of functions is provided, i.e., code can be submitted by the user and execution of the function with parameters is triggered via an exposed endpoint. A runtime system executes the function in an isolated container and automatically scales up the number of containers according to the response time and number of incoming requests. Cus-

tomers are billed for the execution time of the function. The core assumption is that the function is a sensible unit of work, e.g., running for 100ms, running on a single core, side-effect free, and thus only suitable for embarrassingly parallel workloads. Authentication and security is of high importance for these systems as well. For example, OpenFaaS is a Kubernetes-based FaaS system that utilizes, e.g., OAuth to authorize users and to generate tokens that are verified upon function deployment or execution. While this mechanism has similarities to our approach, FaaS is for short-running (subsecond to several seconds) single node jobs, we provide different, security-derived authorization processes for the different available operations, while mitigating user impact via push notifications and solve the issue for long-running HPC systems including parallel jobs.

III. General Architecture

HPCSerA consists in total of three components which enable the access and remote control of an HPC system via a REST API.

These three components as well as their interactions are depicted in Figure 1: The main component is the API server, which at first glance looks like a simple message broker. Clients, shown on the left side in green, can use the REST API of the *API Server* to post a new HPC task. On the opposite side, there is a cronjob running, in the following called *Agent*, which periodically queries the API server for available tasks and pulls them if available. Once pulled, the agent will execute the task and will update the state of the task on the API server accordingly. This simple approach has several advantages:

- If the egress firewall rules allow access to the API server, which would be possible even for HPC systems which do not allow general internet access, the entire setup can be done in user space.
- The agent is independently configurable. This means that the agent does not require a fixed interface, like a certain resource manager, and can be customized to work with any kind of system.
- The agent can only do, what it is configured to do. Therefore, a user can configure what should be exposed. The highest form of exposure would be to allow arbitrary code ingest and execution, like sending a shell script and executing it. A smaller level of exposure would be to just allow the submission of preconfigured batch jobs to the resource manager.
- A user can hook an authorization mechanism into the agent in user space and therefore does not need to completely trust the administrators of the API server or HPCSerA. This mistrust allows a large exposure of the agent in a secure manner.

In the following the three components are presented in more detail.

A. The API Server

As a central component of the *HPCSerA architecture*, the *API server* handles HTTP connections from the *Client* and *Agent* (described below), maintains the internal state of all jobs and functions and resolves dependencies between functions. In addition it provides the necessary maintenance endpoints to allow configuration via the Web UI. It communicates with the database for persistence of the internal state and verification of any authentication tokens. Since every job has to be kept in the database until it is completed, the API server is not stateless. All other connections are initiated by other components, therefore the API server is the only part of the architecture that has to allow incoming connections. It is also the responsibility of the API server to ensure separation between jobs of different projects, i.e., these are only visible in response to requests which are authorized for the same project.

B. The Client

Any application or service that needs HPC as a back-end implements the *Client* component, which initiates HTTP connections to the API server in order to submit jobs, call functions (cf. Section IV) and retrieve information on the job state. Examples of use cases for the Client are given in Section VII.

C. The Agent

On the HPC system the *Agent* component regularly connects to the API server in order to retrieve jobs that are ready for execution. Depending on the function being called, the batch system might be involved and is regularly queried on the state of each pending or running cluster job. This information is used for further calls to the API server in order to keep the job state up to date. In the case of function calls which depend on each other only via the cluster jobs they need to run a corresponding set of jobs including the dependency information is being submitted to the batch system.

IV. Advanced execution models

Extending on this general idea, a more formal execution model can be defined. Generally one can observe that the execution model of predefined tasks triggered by a REST call is a well known concept in the cloud ecosystem known as Function as a Service (FaaS) [16].

A. FaaS for HPC

In FaaS it is typical that a user has a preconfigured task or function which is packaged into a container to be called with varying inputs. These functions are available by user-defined REST endpoints. Since in HPCSerA every user has a dedicated namespace on the API server, this expected behaviour can be replicated on an HPC system using the respective scheduling mechanism for batch jobs.

The basic mechanism for this is shown in Figure 2. It can be seen that the user can send REST requests to the API server resembling FaaS requests. For this, each user has their own namespace `/<username>/function/<functionname>`, where custom functions can be registered at their own discretion. It is important to state that the function name must be

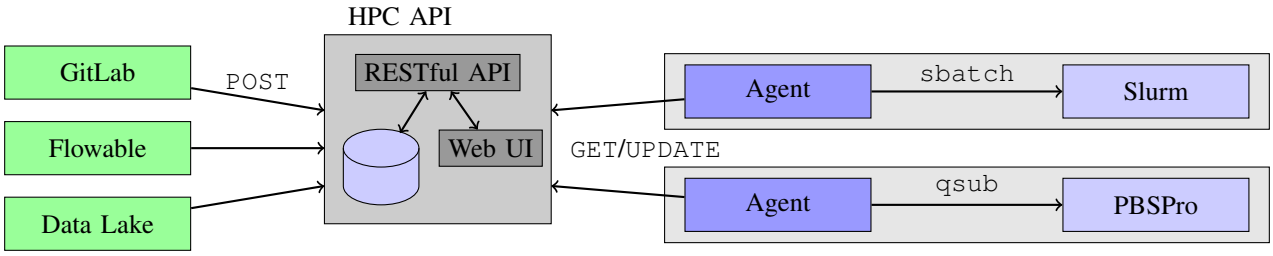


Fig. 1. Components of the architecture: external services, API server, HPC systems (in our use cases we show Slurm and PBSPro as examples, which are used in the Scientific Compute Cluster of GWDG and HAWK at HLRS, respectively).

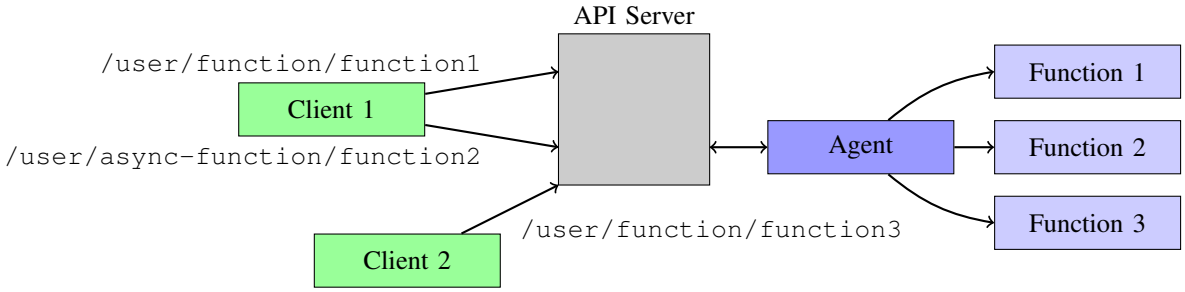


Fig. 2. Basic schema of the FaaS methodology.

unique within the namespace of each user and is not being further isolated by additional structures like the notion of projects. Once a client has posted a function call to the API server, it will be available for the agent to be pulled with the corresponding GET request. The agent then actively pulls these function calls and dispatches them by calling a starter script with the same `<functionname>` located in a preconfigured path, e.g., in the user's home directory in `~/hpcsera/functions/<functionname>`. These functions, which are then being executed, can be anything. It can be a Bash script which is being executed on the frontend, it can be a script fetching data from a remote source and staging it on the fast parallel filesystem of the HPC system, or it can be a simple job submission to the respective batch system, to give just a few examples. Since only executables are being executed, there are no inherent limits to the capabilities of these functions.

B. Long Running vs. Short Running Functions

Since HPCSerA does not enforce any boundary condition on the user-defined functions, it is important to differentiate between long-running and short-running functions from the beginning. The most important difference from the user's perspective is that long-running functions will be usually executed asynchronously, whereas short running-functions can be executed synchronously as well. The reason is that in this case a TCP connection between the client and the API server can stay established during the entire time. Therefore, the HTTP response will correspond to the output of the function (cf. Section IV-G), e.g., a response code `200` would directly mean that the function ran successfully. There might even be some payload data attached to the response, which can be immediately used by the client. The client process is blocking for the duration of the HTTP request. These functions, however, do not only need to have a short runtime, but also need to have limited resource requirements. In those cases, an oversubscribed queue (commonly used to enable interactive jobs) can be used, which can be created and managed by typical resource managers like Slurm.

In the other case, during an asynchronous function execution the client would get an immediate HTTP response from the API server. Here, the return code `202` would however only mean that the request to execute the function was successfully accepted from the API server. This allows for the established TCP connection between the client and the API server to be terminated. Therefore, the client process would only be blocking for the duration of the initial communication with the API server, but not for the entire time the function needs for processing. However, this leaves the client without the optional output data of the function, which might be required. This can be solved on the client side by providing a callback URL in the HTTP header when the initial REST request is made. The API server would in that case make a callback to the client once the function has finished. This is possible since the API server offers statefulness of the functions. Since the API server itself is not meant to handle large data transfers, usually S3 will be used for these cases. Therefore, it might be advantageous to implement some event handling using S3 rather than the API server.

About the difference between synchronous and asynchronous jobs which require access to the compute nodes that are managed by a dedicated resource manager like Slurm it can be stated from the HPC perspective that the synchronous

case is using `srun`, whereas the asynchronous function call is using `sbatch`.

For HPCSerA a single function configuration is enough to execute the same function synchronously and asynchronously. The client can then choose the mode of execution at runtime and just distinguishes between those modes by using a different Endpoint, i.e., either the `/<username>/function/<functionname>` for the synchronous execution, or the `/<username>/async-function/<functionname>` for the asynchronous execution.

C. Remotely Building Complex HPC Jobs

Offering a FaaS infrastructure based on HPC which enables long-running, data intensive and highly parallelized functions is a useful addition for those users who are already in the FaaS ecosystem. There is, however, also the HPC-native user group, where people would like to be able to access and use an HPC system as before, just with a RESTful interface. In order to combine these two scenarios, a closer look at the typical HPC usage is necessary. The usual workflow for users working on HPC systems can consist of several steps:

- The environment and binaries for the computation are prepared. This is mostly done interactively on the frontend.
- Input data for IO-intensive applications is staged on a fast parallel filesystem prior to the job submission.
- Last changes to the batch script are done and the job is submitted to the batch system.
- After job completion the results can be inspected and possibly backed up.

Since there are no restrictions on the capabilities of the functions, one can recreate the workflow described above under two conditions:

- The execution of a function can depend on conditions.
- Code ingest needs to be supported.

The first condition is derived from the requirement that a job can only be submitted to the batch system once its environment is built and its input data is staged. There can also occur other examples and more complex conditions. Since HPCSerA is not in any way supposed to replace a workflow engine it is also not supposed to handle complex conditions on its own. Therefore, one can only add the condition to a function that it should start only after one or more other functions have (successfully) finished. The logic to determine whether a function call has been successful or not has to be within the function itself.

In order to build complete end-to-end HPC jobs with this mechanism these function calls need to be embedded in a suitable data structure.

In Figure 3 it is shown that all function calls are organized within a data structure called Job. A user has to first create a new Job, which gets a unique `JobID` assigned by the API server. Afterwards, a user can call functions within the context of a Job. These function calls then get a Function-ID associated to them. Conditions can be assigned to these function calls, i.e., other functions within this Job structure have to be (successfully) finished before this function can start. This mechanism allows to build up a typical, multi-step HPC job as described above, by calling consecutively the exposed FaaS REST API. Independent function calls will be executed concurrently. In our example, this applies to both the *Prepare Binary* and *Stage Input Data* functions which have no dependencies. *Dispatch Batch Job*, on the other hand, can only be run once both previous function calls are completed. Finally, *Postprocess Data* is run once all other function are completed.

Alternatively, one can define a single HPC job in HPCSerA using a single `YAML` file. In this case all functions need to be known when submitting the job request to the API server. If not specified, the functions are executed in the order they appear in the `YAML` file, and an implicit dependency on the previous function is assumed. In the consecutive buildup where independent functions are called in the context of a job, additional function calls can be issued to the same Job-ID at a later time.

D. Virtual Function Calls

Since all dependencies that HPCSerA is supposed to resolve should only cover the exit status of a function, i.e., with or without error, a mechanism is needed to map more complicated conditions onto this boolean. One example for such a more complicated condition would be that a function should only start after some special resource, like a certain block device, was provisioned.

To cover these cases, one can define *Virtual Functions*. These functions differ from normal functions in that they do not need to be pulled by the agent and then need to be executed. Instead these functions are only existing on the API server. There they expose a REST endpoint, where from an external source the state of the Virtual Function can be changed. This means that some external program can make a REST call to that endpoint to set the state of the function to (successfully) finished. This is an alternative, similar to the call-back URLs provided by clients when triggering an asynchronous function. When an external REST call is used, the necessity to execute functions which do busy waiting to check if a certain condition is met, can be avoided. However, functions which do busy waiting can also be used in a straightforward manner, as long as the necessary logic is implemented to differentiate between the waiting state since the condition is not yet satisfied and the failed state where the condition will be never satisfied. In the latter case the function should be terminated with the corresponding unsuccessful state. If a proper failure condition can not be formalized, when a condition failed and will not be met in the future, a final wall-clock time should be specified after which the function is terminated and the state of the function is unsuccessful.

E. Function Configuration

There are two different ways to configure and deploy a function. The first option is to connect to the HPC system

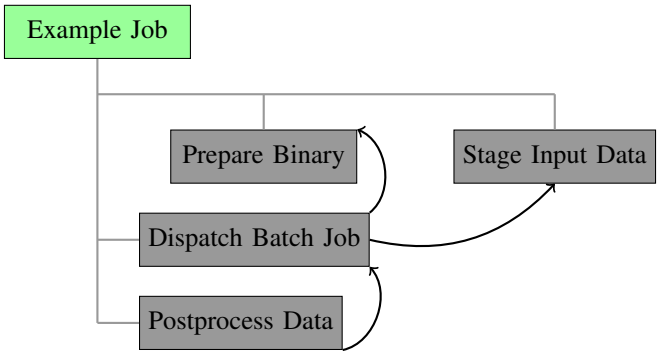


Fig. 3. Sketch of a Job data structure in which three different functions with corresponding Function-ID's are organized.

via SSH and prepare the executable which is called when the function is triggered. This executable can be a binary, or a shell script, for instance. In case a binary should be executed within a certain environment, one can wrap the call of the binary within a shell script. If more complex environments are required, the executable can be packaged togteher with its dependencies into a container, e.g., Singularity/Apptainer [17] can be used. Once the function is configured within such a SSH session, it can be called afterwards by the agent, therefore it is also immediately available for the client on the API server.

The alternative is to configure a new function via the API server, i.e., completely within the context of HPCSerA. For this, all necessary files can be zipped or tarred, and sent along with the configuration request, which is available on a dedicated REST endpoint. The agent will then accept the archive, unpack it into a temporary directory, and will execute a preparation executable. This preparation executable can be as simple as to just copy the the function executable into the function directory of the agent. More complicated examples may include some code which needs to be compiled. For large files, like a Singularity/Apptainer container, it is recommended to upload those via REST to an S3 Bucket. Then just passing a preparation executable to the agent, which fetches this large file from the remote bucket and places it in the necessary path on the HPC system, is enough to configure such a function.

F. Passing Arguments to Functions

Some or even most functions will require that some arguments are passed to them when calling them. These can be passed to the API server of HPCSerA either as Uniform Resource Locator (URL) query parameters, or as a JSON file. In the first approach an arbitrary list of key value pairs can be passed with the calling REST endpoint, e.g., `/<username>/function/<functionname>?k=val`. This call would forward the key `k` with the value `val` to the function in two possible ways: Either the agent would export an environment variable `<PREFIX>_k` with the value `val` (where `<PREFIX>` can be set by the user) before calling the executable corresponding to the called function or alternatively, these key value pairs can be formatted into a single command line string which is appended to the binary call, as it is common when executing an executable on a Linux shell.

In case a function requires more extensive arguments, this previous discussed method is not handy anymore. Instead, one can use a JSON file which is send along with the REST request to trigger the function. This JSON file is then simply forwarded from the API server to the agent which accepts the JSON file and stores it locally. The file path can then be passed as an argument to the function. Neither the API server, nor the agent will in any way process the content of the JSON file. If a function requires this kind of complex input data, the logic needs to be implemented by the function itself or a wrapper script.

G. Returning Output Data to the Client

When a function call is completed, the method of returning its results depends on the mode of execution and the volume of the produced data:

- For synchronously executed functions (cf. Section IV-B) the results are available by the time of completion and can be included in the HTTP response. If applicable, the results can be completely included in the form of a JSON structure produced by the function, e.g., for scripts that query the status of the system, such as custom calls of the batch system or storage CLI tools. Binary data, such as base64-encoded BLOBs can be included, although for the purpose we recommend, especially in the case of a high volume of produced data, that the JSON structure merely contains information about the location of the output data, for example a file system path on shared storage or the URL of an S3 bucket.
- If the function call is asynchronous, only information about the data structures created on the API server can be relayed, in particular the `JobID`. This has to be kept on the client side and used for later status requests.

H. Error Handling

Since the functions have a state, which is managed by the API server, and for instance distinguish between a successful and an unsuccessful exit, the user-defined functions are ideally able to distinguish between those states. However, some error in the code execution itself is not the only error which can

occur. It could also happen that during the function execution the process is unexpectedly killed, or the host is suddenly turned of, for instance due to a power outage. When the agent is able to detect those interruptions, where a function stopped processing without sending a proper exit code, it assumes that a crash unrelated to that particular function has happened and will trigger the function execution again. In order to support this behaviour a function should be idempotent, i.e., it should be possible to execute a function multiple times with the same input parameters and it will always produce the same output.

I. External Job Dependencies

With a slight modification in the data structure describing a job and the included functions, our architecture can support medium-term storage of campaign data as well: The set of `subJobTypes` was originally designed to be run in order, but a more generic solution is given by implementing a directed acyclic graph (DAG) of dependencies. Hereby each function can define one or more dependencies on another function, which can either exist in HPCSerA or represent an external event via a virtual `FunctionID`. The latter is marked as done via an external source, for example when campaign storage or a data source is ready to be used as job input data. This workflow is typically used for research projects and can include dependencies between compute jobs, storage provisioning and data migration. However, the conventional linear chain of `subJobTypes` is still included as the special case where each step depends on predecessor. As shown in the first half of Figure 4 this variant is implemented via dependencies between functions. However, in the more general case, as depicted in the second half, multiple functions could depend on the same prerequisite, in this case `B1`. If a subset of the function calls is implemented via batch jobs and all other dependencies pointing outside of the set are already fulfilled, the corresponding subgraph can be submitted in one step, thereby delegating further resolution of the remaining dependencies to the batch system.

V. Security Architecture

We first analyze the potential security issues from our initial architecture and describe an approach to address them via an updated authorization and authentication process. Finally, each step of the revised workflow is discussed individually.

A. Problem Statement

In the original architecture, static *bearer tokens* were used for user authentication. There was one *bearer token* per user, which means that each client, as well as each agent, authenticated towards *HPCSerA* with the same token, compare [18, III. B.]. Although considered state-of-the-art, this approach has different security flaws which prevented a public deployment. These security problems become apparent when particularly taking into account that an access mechanism for an HPC system is provided. One problem is that this single *bearer token* can be used to access all endpoints, which means that it can be used to perform any possible operation. This can be maliciously exploited in two different ways:

- If that token is not properly guarded, an attacker can use it to post a malicious job, to gain direct access to the HPC system.
- If an attacker has escalated their privileges, the token used by the agent is left vulnerable. If the user has authorized that token to get access to more than one HPC system, the attacker has immediately gained access to another cluster.

There are two different conclusions one can deduce from these observations: First, it is a highly vulnerable step to allow code ingestion via a RESTful service into an HPC system and one has to take the chance of a token loss into account, when designing such a system. Second, the agent sometimes only needs the permissions to read queued jobs and to update the state of a job, e.g., from *queued* to *running*. Therefore, it is an unnecessary risk to allow a job ingress from the token of an agent.

B. Improved Architecture

The separation of access tokens by the user who created them and the services (clients and HPC agents) to which they are deployed, as described in [18], already enables revoking trust in a setup with multiple services and multiple backend HPC systems easily. However, during operation, there is global access to the entire state, i.e., in-flight jobs, to all parties involved. In order to segment trust between groups of services and HPC backends, our revised architecture (cf. Figure 5) resolves this issue by introducing a dedicated tag field into the design of the database for access tokens. Based on this information, client services and HPC agents can be authorized individually. Moreover, each token can be assigned one or multiple roles that restrict the combination of Hypertext Transfer Protocol (HTTP) endpoints and verbs which can be used for all entities that have been created using the same tag. The token's individual lifetime is implied by the granted role.

User control over each individual task and job that is allowed to be run or submitted, respectively, is enforced by introducing an intermediate authentication step that requires user interaction via an external application. This could be run on a mobile device or hardware token, like the ones being used for 2FA or integrated into the web-based user interface used for token and device management for fast iterations on the workflow configuration. Metadata about the action to be authorized is included in the user prompt in order to allow an informed decision. However, the measure is restricted to this most critical step of the process, while non-critical endpoints, such as retrieving the state of pending jobs, can continue to respond immediately. For submitting a new job, the necessity of individual user confirmation is also determined by whether new code is ingested or an already existing job is merely triggered to run on new input data.

From the user's perspective, setting up the workflow would start with logging into the web interface and creating tokens for each service to be connected to the API and configuring them in each client and agent, respectively. In order to acquire

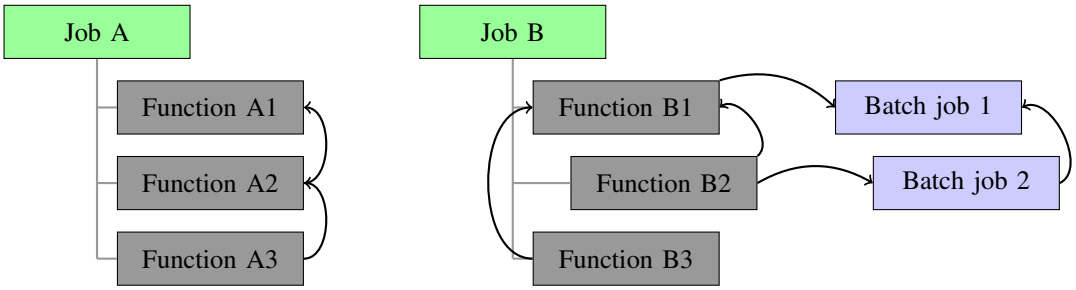


Fig. 4. Jobs with implicitly defined dependencies and a custom dependency DAG.

a minimal working setup, at least one token for the client service and one for the agent communicating to the batch system on the HPC backend system would be required. OAuth-compatible clients could initiate this step externally, thereby sidestepping the need for the user to manually transfer the token to each client configuration. As soon as each client has acquired the credentials either way, HPC jobs can be relayed between each service and the HPC agent.

While the OAuth 2.0 terminology [19] allows a distinction between an authorization server, which is responsible for granting authorization and creating access tokens, and a resource server, which represents control over the entities exposed by the API, in our case the tasks and batch jobs to be run, both roles are assumed by our architecture, so the design can be as simple as possible and deployed in a single step. However, since the endpoints for acquiring access tokens and the original endpoints that require these access tokens are distinct, a separation into microservices (which again need to be authenticated against each other) would also be compatible with the presented design.

The steps necessary for code execution are illustrated in Figure 5. As a preliminary, we assume that the HPC agent is set up and configured with the REST service as an endpoint. The arrows indicate the interactions and the initiator. The individual steps are as follows:

1) The workflow starts by a user logging into the web interface. The Single sign-on (SSO) authentication used for this purpose has to be trusted, since forging the user's identity could allow an attacker to subsequently authorize a malicious client to ingest arbitrary jobs.
2) The user can create tokens for the the REST service in the WebUI.
3) The tokens are stored in the Token database (DB), along with the granted role, project tag, and token lifetime.
4) The retrieved tokens can then be used by a client, e.g., to run some code on the HPC system or have an automatic process in place, provided the code is already present on the system, rendering manual authentication unnecessary.
5) The request is forwarded to the REST Service, which verifies the information in the Token DB. On success, the code to execute is forwarded to the HPC agent.
6) If the client chooses to use the OAuth flow instead in order to avoid manual token creation, the authorization request is forwarded to the Auth app instead.
7) The user can choose to confirm or deny the authorization request. In the former case, the generated token is stored (cf. step 3) in the Token DB. Again, further requests can then in general proceed via step 5 without further user interaction.
8) Like any other client, the HPC agent uses a predefined token or alternatively initiates the OAuth flow in order to get access to the submitted jobs.
9) For the most critical task of executing code on the HPC frontend or submitting batch jobs, the agent can be configured to get consent from the user by using the Auth app for authentication.
   This request is accompanied by metadata about the job to be executed, such as a hash of the job script, allowing an informed decision by the user. This step also avoids the need for trust in a shared infrastructure, since the authentication part can be hosted by each site individually.
10) Once the user has confirmed execution, the HPC agent executes the code, e.g., by submitting it via the batch system. In this case, information about the internal job status is reported back to HPCSerA.

We assume that the HPC agent is secure as otherwise the system and user account it runs on are compromised and, hence, could execute arbitrary code via the batch system anyway. The Web-based User Interface (WebUI), HPC agent, HPCSerA Service and Client are all independent components. For example, a compromised REST Service could try to provide arbitrary code to the HPC agent anytime or manipulate the user's instructions submitted via the client. However, as the user will be presented with the code via the authenticator app and can verify it similarly to 2FA, the risk is minimized.

There are multiple approaches to deploy HPCSerA across multiple clusters and administrative domains:

a) Replication: Each center could deploy the whole HPCSerA infrastructure which we develop (cf. Figure 5) independently, maximizing security and trust. By adjusting the endpoint URL, a user could connect via the identical client to either the REST service at one or another data center – this is identical to the URL endpoints in S3. Although the user now has two independent WebUIs for confirming code execution

on the respective data center, the authenticator and the identity manager behind it could be shared. An additional advantage of this setup would be that the versions of HPCSerA deployed at each center could differ.

b) Shared Infrastructure: The maximum shared configuration would be that for each HPC system a user has to deploy a dedicated HPC agent on an accessible node but all the other components are only deployed once. As the HPC agents register themselves with the REST service, now the user can decide at which center they would like to execute any submitted code. While using a single WebUI for many centers and cloud deployments maximizes usability, it requires the highest level of trust in the core infrastructure: If two of these components are compromised, arbitrary code can be executed on a large number of systems. However, authentication for access to the WebUI via the user's existing account from their HPC center can be implemented as SSO using OpenID Connect Federation.

VI. Implementation

In the following, more details about the technologies chosen for our implementation are provided. Due to the conceptualized architecture in Section V, this section has a focus on the current scope definition and the authentication/authorization scheme employed. Generally, the OpenAPI 3.0 [20] specification, which is a language-agnostic API-first standard used for documenting and describing an API along with its endpoints, operations, request- and response-definitions as well as their security schemes and scopes for each endpoint in *YAML* format, was used to define the RESTful API. This API is backed by a *FLASK*-based web application written in *Python*. The token database is in a SQL-compatible format, thus SQLite can be used for development and, e.g., PostgreSQL for the production deployment. The database schema contains only the user (`user_id`) and project (`project_id`) that the token belongs to as well as the individual permission-level (`token_scope`).

A. Definition of Access Roles

In order to give granular permissions for accessing each of the endpoints, OpenAPI 3.0 allows to define multiple security schemes providing different scopes to define a token matching to the security level of each of the endpoints. Eight different roles have been identified, which are listed and described in Table I.

These roles are entirely orthogonal, which means they can be combined as necessary. If, for instance, on one HPC system only parameterized jobs needs to be submitted, the *agent* can be provided with a token which has only the permissions of role 2 and 3, thus lacking role 5, which is required to fetch new files. Similarly, if a token is provided to a client which is not 100% trustworthy, one can choose to only provide a token with the role 6, i.e., to only allow to trigger a predefined job. Important to understand is the difference in mistrust between the role 3, 4, and 5. The security mistrust in role 4 comes from the admins of the *HPCSerA*, who want to ensure that a code ingestion is indeed done by the legitimate user. Therefore, in order to allow code ingestion, the possession of a token with the corresponding permission is not enough, the user has to confirm the code ingestion via a 2FA. The mistrust in role 3 and 5 comes, however, from the user, who wants to ensure that only jobs s/he confirmed are being executed. This is, again, completely orthogonal, to the enforced 2FA in role 4 and can be optionally used by the user. This fine-grained differentiation between the different security implications of the discussed endpoints minimizes user interference while providing a high level of trust.

B. Providing Tokens via Decoupled OAuth

The introduction of OAuth-compatible API endpoints has several advantages: Access tokens can be created on demand in a workflow initiated by a client or HPC agent, respectively. In addition, while there is a default API client provided, a standard-compliant API enables users to easily develop drop-in replacements.

It is important to note here that we modified the usual OAuth authorization code flow, where a client gets redirected to the corresponding login page to authorize the client. This "redirect approach" has two problems:

- The client is a weak link, where the Transport Layer Security (TLS) encryption is terminated and therefore becomes susceptible to attacks and manipulation.
- It does not support a headless application, like the HPC agent, which is not able to properly forward the redirect to the user.

Due to these shortcomings, a modified OAuth flow was developed to enable the usage of headless apps and improve security. This modified version decouples the user confirmation from the client, which means that the client is not being redirected but that the confirmation request is being sent out-of-band, e.g., via the WebUI or via notification on a smartphone device.

Starting with the case that the script does not already come equipped with a token, analogous to the usual OAuth flow, the generation of a token is requested. Since our use case was initially built as an instance of machine-to-machine interaction, i.e., headless, the issue of a lack of user interface is encountered; the usual OAuth flow - implemented in the browser - would redirect the user to an authorization server where the user could actively provide their username and password to the authorization server. The authorization server would then return a code, in the case of the authorization code flow, in the redirect URI, which would be posted in a backchannel, along with a client secret assigned at the time of registering the client to attain an access token.

In order to circumvent this headless-app problem, this work has implemented a synchronous push notification system analogous to the Google prompt where a notification is pushed to a user's device awaiting a confirmation to proceed. In the Minimum Viable Product (MVP), we have implemented this in the SSO-secured WebUI in order to have a more integrated interface. Eventually, the final product will see an Android

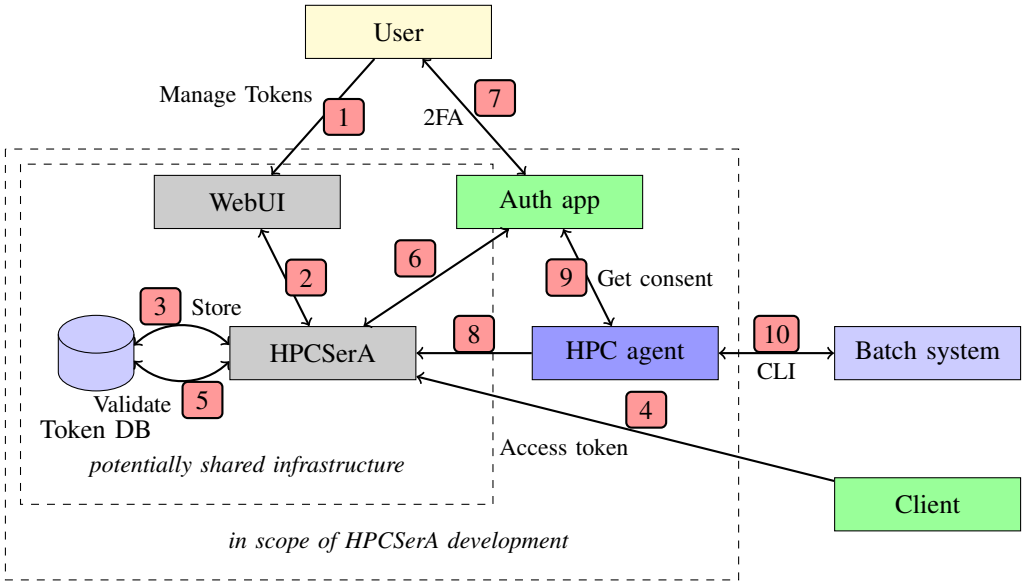


Fig. 5. A sketch of the proposed token-based authorization flow. The following parts are shown: 1) WebUI login 2) Connection to the HPCSerA service 3) Storage of access tokens 4) Client connecting to the API 5) Validation of access tokens 6) Authorization request 7) User interaction with the Auth app 8) HPC agent connecting to the API 9) Authentication request for code execution 10) Interaction with the HPC batch system.

TABLE I
DEFINITION OF THE EIGHT ROLES. OPERATIONS MARKED IN RED HAVE TO BE CONSIDERED SECURITY CRITICAL FROM THE ADMIN POINT OF VIEW, WHEREAS THE ORANGE MARKED OPERATIONS FROM A USER POINT OF VIEW.

Role Number	Role	Description
1	GET_JobStatus	Client can retrieve information about a submitted job
2	UPDATE_JobStatus	Used by client/agent to update the job status
3	GET_Job	Endpoint used by the agent to retrieve job information
4	POST_Code	Client to ingest new code to the HPC sytsem
5	GET_Code	Agent pulls new code. Might be necessary to run new job
6	POST_Job	Client triggers parameterized job
7	UPDATE_Job	Client updates already triggered job
8	DELETE_Job	Client deletes already triggered job

and iOS app that receives such notifications. This flow then grants the permission to execute a security critical operation, compare Table I.

This confirmation via push notification cannot solely rely on time-synchronicity since it would be susceptible to an attacker requesting tokens and/or 2FA confirmation for carrying out a security-critical operation in the same approximate time frame. Therefore, a sender constraint has to be implemented. This is done in a similar way to the original authorization code flow: The access code is signed with a client secret which was configured with *HPCSerA* prior to the execution of this workflow, and then sent to *HPCSerA*. *HPCSerA* verifies the secret and only then sends the actual token. This secret is implemented using public-private key pairs, where the public key is uploaded to *HPCSerA* in the initial setup to register a new client (or agent).

Alternatively, in the case that a token is supplied along with the software or script that is submitting a job to the HPCSerA API, the permissions are validated against a token database. In the case that the token provided contains permissions for accessing a sensitive endpoint, the second factor check is triggered through the WebUI and the notification / confirmation process is once again undergone. It is important to note that this is not a hindrance since already-running jobs and non-sensitive endpoints proceed without user-intervention.

C. Mapping of Roles to Functions

In order to provide the user with a FaaS interface which is capable of handling automated machine-to-machine communication of headless apps the previously defined roles need to be mapped on the FaaS endpoints. The most important differentiation is still between the `POST_Job` role and the `POST_Code` role. The latter is required, when a user wants to configure a new function via the API server. Here, the user can upload new code either directly as an archive, or via an external storage. Therefore, the configuration of a new function corresponds to the `POST_Code` role. The client making that request needs to have this elevated access rights.

On the other hand, simply triggering the execution of an already configured job, for instance on new input data, corresponds to the `POST_Job` role. As shown in Table I, this role is not security sensitive. Thus, it can be used by a client without any manual interaction as long as the client has a

token with the corresponding role. Therefore, HPCSerA can support automated FaaS functionality towards its client.

The agent side was not considered critical for the admins, but optionally critical for the users if they distrust the API server, if they want to implement their own 2FA mechanism here. To support the previously discussed endpoints, the client needs to either use the `GET_Job` role to receive the request to execute a function, or the `GET_Code` role to pull some new code and to configure a new function. The agent would then also require the `GET_JobStatus` role and the `UPDATE_JobStatus` role to manage the state of the functions, which is maintained on the API server. Via the `UPDATE_JobStatus` role the output data of a function can be send to the requesting client. The client would then also need the `UPDATE_JobStatus` role in order to be allowed to receive the output data.

D. Assigning Roles to Clients and Agents

The fine-grained distinction between those different roles, as discussed in Section VI-C, is an important part to provide the highest level of security while enabling a high degree of automation. This means especially that users should only use tokens with the minimum roles attached to them. For instance, if a user will configure functions always manually within a SSH session, the token of the agent should not have the `GET_Code` role enabled. Users define the roles of the tokens in the current setup within the WebUI. Here, users can either check the needed roles, create a token, and copy it out of the WebUI or they can, upon request from a client or agent via the presented detached OAuth flow, choose which roles should be associated to the token which will be created. Once a token has been created, it can also be revoked if it is not needed anymore or a potential breach is assumed.

VII. Use-Cases

Due to the previously stated changes in the architecture, there are certain adaptions in the previously presented use cases [18]. These changes will be discussed in the following and serve as the basis for a broader user impact analysis.

A. GitLab CI/CD

Since the *GitLab* Runner can be configured to run arbitrary code without including secrets in the repository, thanks to GitLab's project Continuous Integration and Integration Development (CI/CD) variables [21], the required tokens can be made available to the CI/CD job so it can in turn access the API endpoints required to transmit the current repository state to an HPC system where the code can be tested using the HPC software environment or even multiple compute nodes.

A new commit might of course introduce arbitrary code to the HPC environment, therefore it is advisable to enforce the extra authentication step (cf. Section V-B), when code from a new commit is submitted to the HPC system. The corresponding hash, available by default via the `GIT_COMMIT_SHA` variable, would be a helpful piece of information to display to the user when asking to authorize the request.

B. Workflow Engine

In the workflow use case, HPC jobs should be fully automated without user interaction. Due to multiple repetitions and time dependencies, interactions severely limit the functionality and practicability of the workflow. One possibility is to prepare the workflow in such a way that only parameterized jobs are called and thus only safe endpoints of HPCSerA are used. Another possibility is to use dedicated (legacy) endpoints that are only accessible through firewall regulations and fixed network areas. The latter can also be regulated via an additional proxy server, such as a *nginx*.

There are various levels where dependencies between jobs can be managed. The following descriptions and examples refer to Figure 6:

1) Dependency resolution can be completely handled by the *workflow engine*. In this case, workflow tasks are submitted as individual jobs via HPCSerA. If there is a dependency between two jobs that require a batch job to finish, on completion of the first cluster job the agent updates the job state on the API server from which the workflow engine eventually obtains the new state. In our example, this is the requirement to proceed from Task I to the dependent Task II. Only then can the second job be submitted to the API and if finally retrieved by the agent and submitted to the batch system. In conclusion, this variant is the easiest to implement but involves a high amount of latency for resolving job dependencies.
2) For jobs that are submitted with multiple Function-IDs, the *API Server* will handle dependencies by only providing function calls to the HPC agent for which all function calls on which they depend have been successfully completed. Comparing to the previous scenario, once the agent has marked the last batch job of Job A (A2 in our example) as completed, the function status of A2 on the API server is updated and the next one (function A3) can be immediately retrieved and run. While the dependency chain has to be implemented by building more complicated calls to the REST interface, there is no back and forth communication with the client contributing to the latency.
3) In view of Section IV-I the most low-latency resolution of job dependencies occurs when multiple Function-IDs which contain batch jobs are presented by the API server to the *Agent*. In this case, the completed first batch job (A1) directly leads to the scheduling of the second batch job (A2) by the batch system without interference from any *HPCSerA* components.

C. Data Lake

In order to provide high-performance computing capabilities to a data lake [22], *HPCSerA* is used to submit jobs on behalf of the data lake users. A user sends a so-called *Job Manifest* to the data lake, where the software, the compute command, the environment, and the input data are unambiguously specified. By transferring the responsibility of scheduling the job from

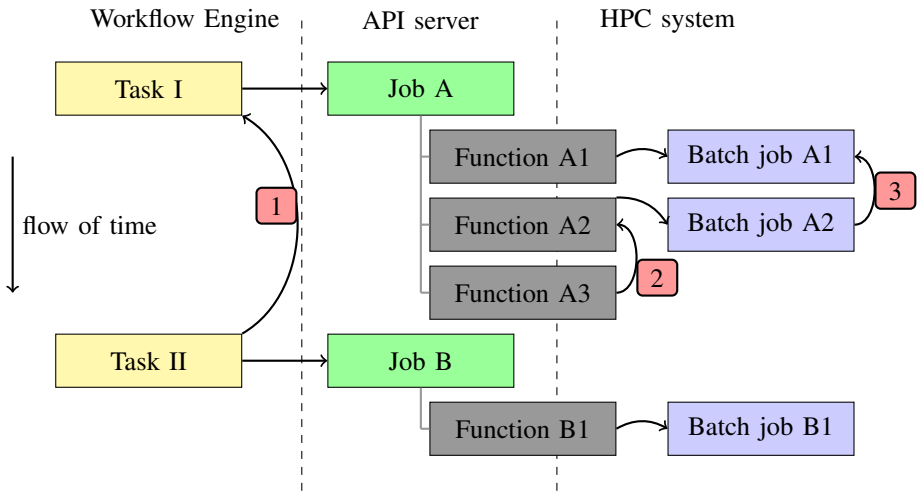


Fig. 6. Overview of the levels at which function dependencies can be resolved.

the user to the data lake, it has the control about it. This allows to reliably capture the data lineage and to foster reproducibility. The added benefit of the newly implemented security measures in *HPCSerA* is that users had to trust the data lake, and hereby the admins, with their *bearer tokens* before. By introducing *OAuth* and enforcing 2FA for code ingestion, this is not necessary anymore, since users now need to confirm each submission. Since users submit jobs actively, for instance via a *Jupyter Notebook* using a PythonSDK, the requirement to confirm each submission does interrupt the workflow too much.

VIII. Conclusion and Future Work

In the paper presented here, we have examined the issue of security in accessing HPC resources via a RESTful API. The initial situation with a very simplified token model does not meet the requirements. Therefore, a fine-granular token model, coupled with interactive user consent and OAuth flows, was proposed. With this new model, particularly critical interactions, such as code transfer, can be secured. User consent is requested in a prototype via a WebUI, which in turn uses a central Identity Management (IDM) for authentication. This means that no critical user-specific data needs to be managed.

Moreover, we presented an extension on the execution models that are possible in our architecture by supporting a *Function as a Service* (FaaS) idiom. Here users can define dependencies between function calls and choose between synchronous and asynchronous execution in analogous way to how HPC jobs can be immediately run on an oversubscribed queue vs batched for running with guaranteed resources.

Compared to the related work discussed in Section II, this paper presents a RESTful API which does not require the users to provide full SSH access to a potentially untrusted API server, as it is required in [12], [13]. Instead, using an agent which pulls from user space the incoming tasks guarantees that the user alone stays in full control the entire time. This approach is paired with a fine-granular role-based access model, and a novel authorization flow to enable OAuth-like authorization for headless applications. This mechanism allows automated workflows to access an HPC system to execute pre-configured tasks on new input data while still enforcing a similar security level to an SSH access with 2FA enabled.

In future work, the possibilities for obtaining user consent will be further analyzed. The development of mobile apps is planned, which will greatly simplify the consent workflow for the user. This is supposed to extend the currently used consent mechanism based on a WebUI. So far, the focus has been on the transmission and execution of code. However, there is also a requirement to transmit data objects that are necessary for execution. Therefore, it is examined to what extent the current implementation is suitable for such tasks and where possible limits are reached in terms of data quantity and transmission speed.

In addition, the FaaS approach lends itself well to collecting statistics about the frequency of function calls as well as metrics about their runtime behaviour. The most convenient way of presenting this information to the user would be inside the Web UI that is already required in our architecture for project and token management.

It would be advantageous to ease the configuration process, ideally to a degree where a user can just insert some code in the web interface of the API server. The agent would need to be extended to automatically work with predefined templates for different languages. For example, if a user inserts Python code in the interface, the agent would prepare a virtual environment with the necessary modules and insert the Python file in the correct place and transparently manage the corresponding environment information.

Acknowledgments

We gratefully acknowledge funding by the “Niedersächsisches Vorab” funding line of the Volkswagen Foundation and “Nationales Hochleistungsrechnen” (NHR).

REFERENCES

[1] M. H. Biniaz, S. Bingert, C. Köhler, H. Nolte, and J. Kunkel, "Secure Authorization for RESTful HPC Access," in *INFOCOMP 2022, The Twelfth International Conference on Advanced Communications and Computation*, C.-P. Rückemann, Ed., 2021, pp. 12–17.

[2] J. Decker, P. Kasprzak, and J. M. Kunkel, "Performance evaluation of open-source serverless platforms for kubernetes," *Algorithms*, vol. 15, no. 7, p. 234, 2022.

[3] Z. Wang et al., "RS-YABI: A workflow system for Remote Sensing Processing in AusCover," in *Proceedings of the 19th International Congress on Modelling and Simulation*. MODSIM 2011 - 19th International Congress on Modelling and Simulation - Sustaining Our Future: Understanding and Living with Uncertainty, 2011, pp. 1167–1173.

[4] A. K. Singh and S. D. Sharma, "High Performance Computing (HPC) Data Center for Information as a Service (IaaS) Security Checklist: Cloud Data Governance." *Webology*, vol. 16, no. 2, pp. 83–96, 2019.

[5] J.-K. Lee, S.-J. Kim, and T. Hong, "Brute-force Attacks Analysis against SSH in HPC Multi-user Service Environment," *Indian Journal of Science and Technology*, vol. 9, no. 24, pp. 1–4, 2016.

[6] T. Ylonen, "SSH - Secure Login Connections Over the Internet," in *Proceedings of the 6th USENIX Security Symposium (USENIX Security 96)*. San Jose, CA: USENIX Association, Jul. 1996, pp. 37–42, [accessed: 2022-03-21]. [Online]. Available: https://www.usenix.org/conference/6th-usenix-security-symposium/ssh-secure-login-connections-over-internet

[7] J. Buchmüller *et al.*, "Extending an open-source federated identity management system for enhanced hpc security."

[8] OpenFaaS. (2022) Invocations. [accessed: 2022-12-13]. [Online]. Available: https://docs.openfaas.com/architecture/invocations/

[9] P. Calegari, M. Levrier, and P. Balczyński, "Web portals for high-performance computing: a survey," *ACM Transactions on the Web (TWEB)*, vol. 13, no. 1, pp. 1–36, 2019.

[10] R. Menolascino *et al.*, "A realistic UMTS planning exercise," in *Proc. 3 ACTS Mobile Communications Summit 98*, 1998.

[11] S. Cholia and T. Sun, "The newt platform: an extensible plugin framework for creating restful hpc apis," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 16, pp. 4304–4317, 2015.

[12] F. A. Cruz *et al.*, "FirecREST: a RESTful API to HPC systems," in *2020 IEEE/ACM International Workshop on Interoperability of Supercomputing and Cloud Technologies (SuperCompCloud)*, 2020, pp. 21–26.

[13] V. Svaton, J. Martinovic, J. Krenek, T. Esch, and P. Tomancak, "Hpc-as-a-service via heappe platform," in *Conference on Complex, Intelligent, and Software Intensive Systems*. Springer, 2019, pp. 280–293.

[14] SchedMD. (2022) Slurm REST API. [accessed: 2022-03-18]. [Online]. Available: https://slurm.schedmd.com/rest.html

[15] C. Dunlap. (2022) MUNGE Uid 'N' Gid Emporium. [accessed: 2022-03-21]. [Online]. Available: https://dun.github.io/munge/

[16] J. Decker, P. Kasprzak, and J. M. Kunkel, "Performance evaluation of open-source serverless platforms for kubernetes," *Algorithms*, vol. 15, no. 7, 2022. [Online]. Available: https://www.mdpi.com/1999-4893/15/7/234

[17] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," *PloS one*, vol. 12, no. 5, p. e0177459, 2017.

[18] S. Bingert, C. Köhler, H. Nolte, and W. Alamgir, "An API to Include HPC Resources in Workflow Systems," in *INFOCOMP 2021, The Eleventh International Conference on Advanced Communications and Computation*, C.-P. Rückemann, Ed., 2021, pp. 15–20.

[19] D. Hardt, "The OAuth 2.0 Authorization Framework," RFC 6749, Oct. 2012, [accessed: 2022-03-21]. [Online]. Available: https://www.rfc-editor.org/info/rfc6749

[20] OpenAPI Initiative. (2017) OpenAPI Specification v3.0.0. [accessed: 2022-03-21]. [Online]. Available: https://spec.openapis.org/oas/v3.0.0

[21] GitLab. (2022) GitLab CI/CD variables. [accessed: 2022-03-18]. [Online]. Available: https://docs.gitlab.com/ee/ci/variables/

[22] H. Nolte and P. Wieder, "Realising Data-Centric Scientific Workflows with Provenance-Capturing on Data Lakes," *Data Intelligence*, pp. 1–13, 03 2022. [Online]. Available: https://doi.org/10.1162/dint_a_00141