HArMoNICS: High-Assurance Microgrid Network Infrastructure Case Study

Gianluca Roascio, Gabriele Costa, Emmanuel Baccelli, Lukas Malina, Raimundas Matulevicius, Marius Momeu, Nerijus Morkevicius, Enrico Russo, Branka Stojanovic, Aimilia Tasidou

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.DOI

HArMoNICS: High-Assurance Microgrid Network Infrastructure Case Study

GIANLUCA ROASCIO[1], GABRIELE COSTA[2], EMMANUEL BACCELLI[3], LUKAS MALINA[4], RAIMUNDAS MATULEVIČIUS[5], MARIUS MOMEU[6], NERIJUS MORKEVIČIUS[7], ENRICO RUSSO[8], BRANKA STOJANOVIĆ[9], and AIMILIA TASIDOU[10]

[1]Department of Control and Computer Engineering, Polytechnic of Turin, 10129 Turin, Italy (email: gianluca.roascio@polito.it)
[2]Systems Security Modelling and Analysis Group, IMT School for Advanced Studies, 55100 Lucca, Italy (email: gabriele.costa@imtlucca.it)
[3]National Institute for Research in Computer Science and Automation, 75012 Paris, France (e-mail: emmanuel.baccelli@inria.fr)
[4]Department of Telecommunications, Brno University of Technology, 60190 Brno, Czech Republic (e-mail: malina@vut.cz)
[5]Department of Computer Science, University of Tartu, 50090 Tartu, Estonia (e-mail: raimundas.matulevicius@ut.ee)
[6]Department of Informatics, Technology University of Munich, 80333 Munich, Germany (e-mail: momeu@sec.in.tum.de)
[7]Department of Computer Science, Kaunas University of Technology, 44249 Kaunas, Lithuania (e-mail: nerijus.morkevicius@ktu.lt)
[8]Department of Computer Science, Bioengineering, Robotics and Systems Engineering, University of Genoa, 16145 Genoa, Italy (e-mail: enrico.russo@unige.it)
[9]Institute for Information and Communication Technologies, Joanneum Research Digital, A-8010 Graz, Austria (e-mail: branka.stojanovic@joanneum.at)
[10]Télécom SudParis, 91000 Courcouronnes, France (e-mail: aimilia.tasidou@telecom-sudparis.eu)

Corresponding author: Gianluca Roascio (e-mail: gianluca.roascio@polito.it)

This paper is supported by European Union's Horizon 2020 research and innovation programme under grant agreement No. 830892, project SPARTA.

ABSTRACT Modern Intelligent Infrastructures (II) are highly complex, interconnected systems that are now emerging. For instance, II can integrate technologies and processes to provide citizens with faster services and better goods. An average II can include many technologies, e.g., Cloud applications and IoT devices, under different environments, e.g., industry 4.0 production plants and smart buildings. Although II bring concrete benefits to all of these contexts, they also carry security concerns.

Reasoning about threats and security exposures that might affect II is non trivial. This is only partially due to their inherent complexity. As a matter of fact, real II are typically in charge of some critical operations that cannot be interrupted or compromised for experimental purposes. An alternative solution is to rely on digital replicas which can provide a good trade off between realism and usability. These assets represent a strategic and highly demanded resource for the security community.

In this paper we present HArMoNICS, a case study infrastructure meant to provide a playground for security experts interested in II security. HArMoNICS revolves around a digital replica of a real Smart Polygeneration Microgrid (SPM) located in Italy. Although most of the components are based on or inspired to the real system, HArMoNICS has been enriched with further security-relevant features. As a result, the case study includes vertical uses cases focusing on specific security topics. Security researchers can use it to assess the effectiveness of new methodologies, to carry out security training activities, or even to extend it with new elements.

INDEX TERMS security, cybersecurity, embedded systems, intelligent infrastructure, case study

I. INTRODUCTION

The development of modern Intelligent Infrastructures (II) promises to raise the bar for several aspects of our everyday life. Nowadays, computationally-enabled devices include televisions, watches, alarm clocks, and many others surrounding us. These devices are progressively outnumbering other types, e.g., personal computers and smartphones, and they form the well-known Internet of Things (IoT). The number of connected devices is expected to skyrocket from 8.74 billion (2020) to more than 25.4 billion (2030) [1]. IoT is an enabling technology for II, yet not the only one. As a matter of fact, other technological, e.g., Fog and Cloud computing [2] [3], and infrastructural pillars, e.g., 5G networks [4], are directly involved.

As it often happens, along with opportunities, these technologies also bring new risks. Serious concerns exist that latent vulnerabilities may pave the way for attackers. Needless to say, security violations would have a dramatic impact, e.g., on the privacy of citizens and the continuity of critical services. Also, existing security mechanisms might not be

applicable to II. A reason is that these infrastructures are extremely complex and heterogeneous, made of myriads of objects using hardware and software of many different manufacturers. Such an extreme diversity, together with the quick development of modern technologies, requires appropriate countermeasures to ensure the security of the next-generation II.

A growing trend is that of using *Security-by-Design* (S×D) [5] as the leading principle. In short, S×D enriches the traditional development lifecycle with security-specific tasks. These tasks take place at every stage, from the very early design to the final deployment. By integrating these tasks in the development workflow, the S×D approach creates a security management process where every critical event triggers effects along the entire lifecycle. As a practical example, consider the case of a new vulnerability discovered in an existing II. Such a vulnerability may occur due to a design weakness, which one might want to evaluate against the initial security specifications. At the same time, the vulnerability may trigger the development of countermeasures, e.g., an emergency patch, as well as new security tests.

Although the ideas behind S×D and its potential benefits are clear, implementing it is still a major challenge. As a matter of fact, all the security tasks previously mentioned must be implemented and populated with actual security tools and procedures. For each of them, many alternatives exist and new ones appear over time. As a consequence, the entire security workflow must be continuously revised and maintained. Even worse, assessing the actual effectiveness of the involved security procedures is hardly feasible. For instance, most of these procedures require discontinuing the operations of the II, which is typically infeasible. Hence, the actual effectiveness of the processes it often only accessed when a real security event occurs and, if they fail, when it is too late.

A possible solution for the assessment of S×D frameworks is to rely on II replicas. For instance, computer simulations can accurately reproduce the behavior of a real system. However, virtual II are rarely available. A reason is that, although simplified, they typically bring part of the complexity of a real II. Thus, they are often considered a valuable asset by private actors who may prefer not to be shared. Furthermore, when they are created to mimic a real II, they may lack the security aspects of interest, e.g., vulnerabilities, for assessing the effectiveness of security solutions.

In this paper, we present our *High-Assurance Microgrid Network Infrastructure Case Study* (HArMoNICS), developed within the EU project SPARTA.[1] Briefly, HArMoNICS revolves around a zero-emission building scenario in the context of a smart microgrid. All the elements appearing in our case study are taken from or inspired by actual systems that reside inside the original infrastructure.

The main motivation behind our proposal is the strategic role of intentionally vulnerable environments, which provide a shared setting for research and development of security techniques. Indeed, such environments stimulate and favor innovation by providing a touchstone for new methodologies. The main goal of HArMoNICS is to be a valuable asset for the community interested in the security and privacy of critical and intelligent infrastructures. Researchers can use it both as a *playground*, e.g., for hosting training exercises, and as a *benchmark*, e.g., for systematically assessing the effectiveness of a certain security mechanism under different scenarios. These activities can be carried out by taking advantage of the built-in use cases. Furthermore, new use cases can be added to model specific security concerns, e.g., related to new technologies. Finally, through a VPN-based integration mechanism, physical devices can be plugged in, and dually, HArMoNICS can be used to extend existing environments, such as simulators or digital twins.

In summary, among the features of HArMoNICS, the following ones are those we consider highly relevant for the security community.

- The case study is designed and implemented in order to mimic a real II, and therefore it includes a number of technologies that may appear within the perimeter of a smart infrastructure.
- Our case study includes 8 security and privacy use cases revolving around specific threats and weaknesses common to most II, including but not limited to: (i) software integrity and updates for end-point devices; (ii) privacy-preserving data management and processing; (iii) intrusion detection; (iv) protocol verification, and; (v) fog computing orchestration and hardening.
- The case study blueprint and implementation are open-source, with the possibility of extending and modifying them.
- The entire case study can be executed inside a publicly-available virtual machine, with minimal computational resources.

The rest of this paper is organized as follows. Section II describes the general features of architecture and networking related to the case study infrastructure. Section III details every single scenario included, with the relative security and privacy problem statement. Section IV offers a glimpse of the current state of the art in such a kind of digital replicas. Finally, Section V concludes the paper.

II. ARCHITECTURE OVERVIEW

The HArMoNICS case study is based on a smart building scenario, composed of both IT and OT elements. The scenario is inspired to the *Zero-Emission Building* (ZEB), which is hosted inside the Genoa University Campus, located in Savona (Italy). Figure 1 shows a sky view of the Campus, with the ZEB location highlighted in red. Among the research infrastructures and facilities that the Campus hosts, there is a smart grid, called *Savona Polygeneration Microgrid* (SPM), consisting of several nodes for the generation of power. Power generation nodes rely on different sources, e.g.,

[1] www.sparta.eu

solar panels and gas turbines, and their production partially supplies the internal energy demand.

FIGURE 1. The Savona Campus and the Zero-Emission Building (in red).

ZEB is a smart building where innovative technologies and materials are adopted in order to optimize the energy consumption with the ultimate goal of nullifying the carbon footprint. It contains several laboratories, offices and a gym. Some servers and networks reside in the building and they host the services which contribute to the IT infrastructure. Standard and Fog-enabled access points provide wireless connectivity to network devices, including IoT ones. Also, sensors and actuators have been deployed to monitor the environment (e.g., room temperature) and reconfigure it (e.g., by opening a window).

A. NETWORK INFRASTRUCTURE

The network infrastructure of HArMoNICS is here described in more details. Figure 2 shows the reference network scheme of the case study. Briefly, the smart building hosts a three-segment network. The three segments are `dmz`, `intranet` and `iot`. Servers hosting public services, i.e., those that are accessible from outside the network perimeter, are connected to `dmz`. Other servers, instead, and hosts are connected to `intranet`. Finally, `iot` is used for connecting various field devices, e.g., sensors, actuators, and Fog nodes. The smart building network resides behind a router firewall that delimits the perimeter with the external network, i.e., the public Internet.

Networks and nodes are labeled with their *symbolic* names, e.g., `ns` and `www`, and IP addresses. Node names are managed by the DNS service running on node `ns`. Network address space represents the interval of IP addresses that can be assigned to connected devices. For brevity, statically assigned IPs are only represented by the last address segments. For instance, the IP address of node `www` is `198.51.100.3`. If a node has no address label, its IP address is dynamically assigned. Finally, when relevant, connections are labeled with a specification of the used channel, e.g., IEEE 802.15.4.

The colored and numbered circles indicate that the nearby devices belong to one of the 8 security and privacy-related scenarios described, which will be detailed in the next Section.

The network topology described above is implemented by means of Docker containers and networks [6]. Roughly speaking, Docker containers are lightweight virtual machines running inside isolated Linux processes. Container connectivity is granted by means of virtual networks which are emulated by the host machine.

Figure 3 highlights the implementation details of the network infrastructure of HArMoNICS. Below, the core aspects are discussed.

- *Public network simulation*. Several devices connect with the scenario infrastructure by means of the public Internet (see Figure 2). Some of them are actual, remote entities (e.g., web servers), while others must be deployed within the scenario. To support this hybrid structure, the infrastructure relies on a *simulated internet* (Figure 3). The simulated internet is implemented by means of a router (`rt-simint`) which connects three networks, i.e., `ext`, `simint` and `outside`. Briefly, `ext` is connected with a virtual interface of the host platform. By means of a virtual bridge, the host interface provides direct connectivity with the real Internet.
- *VPN access*. To make the infrastructure extensible, direct access to each network is provided, by means of a virtual private network (VPN) server. The VPN server is connected to `ext`. In this way, the VPN can be accessed from software running on the host platform and even from the Internet. The server accepts connections on different ports, in the range [`8886 - 8889`]. Each port is uniquely mapped into one of the network in the infrastructure. In Figure 3, port numbers are used to label (in red) the connection with the corresponding network. For instance, establishing a session with `172.16.255.100:8886` will connect the VPN client to the intranet.

HArMoNICS has been designed with a standard *Infrastructure-as-Code* (IaC) framework called TOSCA [7]. Docker Compose [8], the default Docker orchestrator, has been chosen as scenario deployment tool. Finally, the scenario code is hosted on a Github repository at https://github.com/enricorusso/spartawp6. In this way, by deploying HArMoNICS from the Github repository, continuous integration is also supported.

III. USE CASES

This Section provides a glimpse of the main use cases included in HArMoNICS. Each of the scenarios focuses on a different security concern and refers to a subsystem

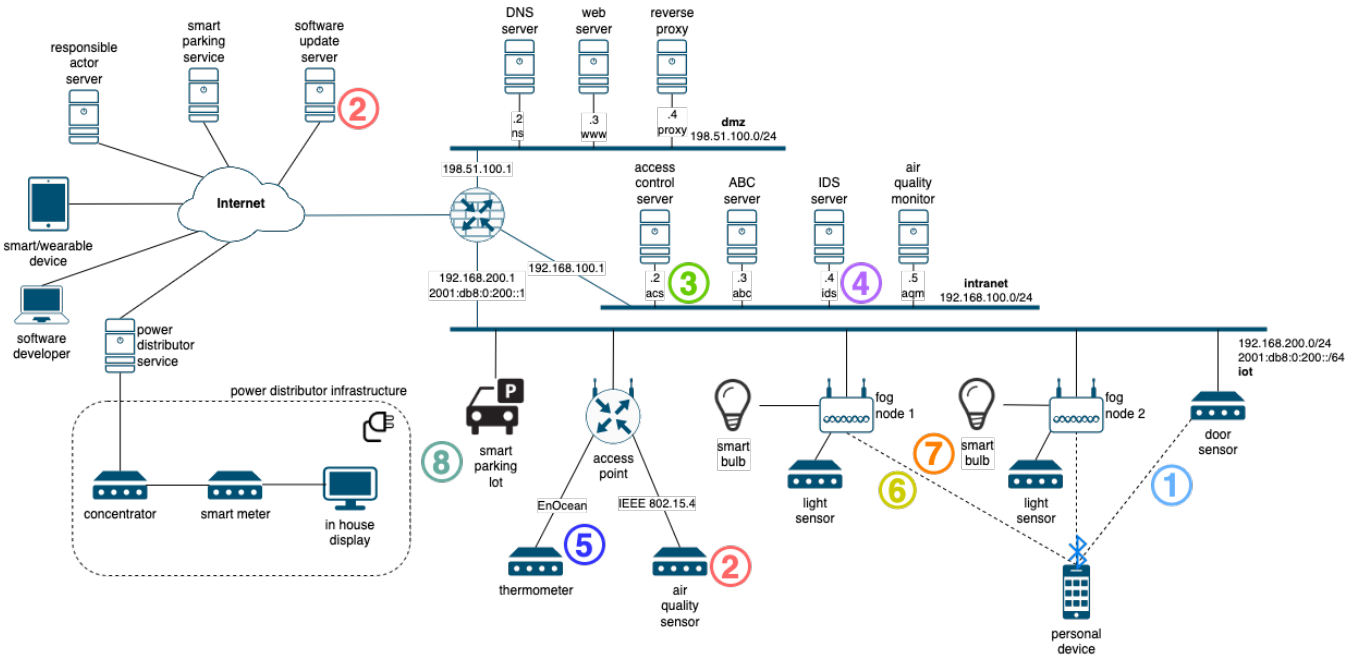


FIGURE 2. HArMoNICS network infrastructure overview.

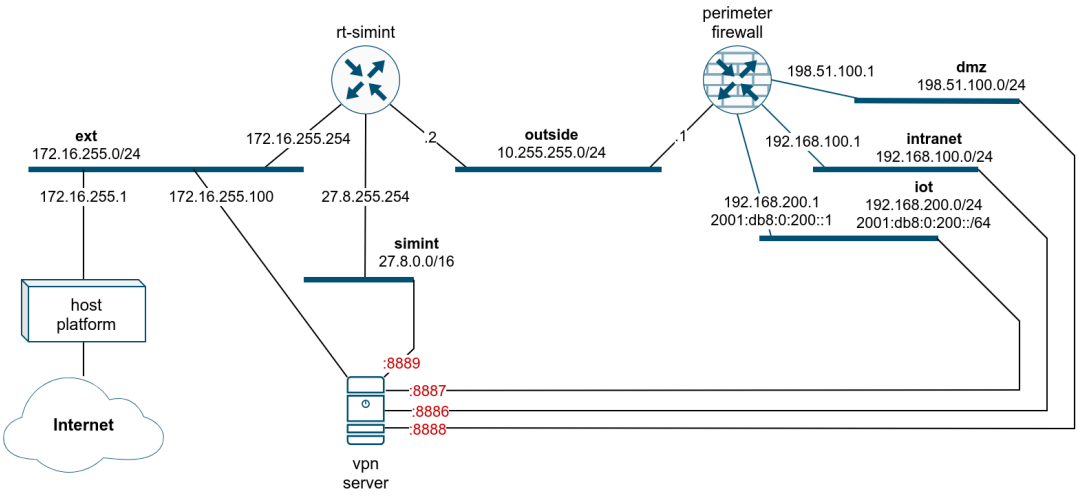


FIGURE 3. Scenario implementation overview.

within the case study infrastructure. Furthermore, for each scenario, we carry out a discussion of open challenges and possible countermeasures that are being investigated in the context of the SPARTA project. The discussed techniques are relevant to highlight how security researchers can leverage on HArMoNICS for assessing their methodologies in a realistic setting.

A. SOFTWARE INTEGRITY FOR SMART SENSORS

One of the main threats to the security of II is the limited, or even absent, protection of the sensing devices that often form a significant part of the perimeter of the entire infrastructure. For example, the smart door sensor, placed within the *iot* network, has the task of monitoring the access of individuals inside the building (item 1 in Figure 2, in cyan). To do this, it pairs over Bluetooth with the mobile device of the person accessing, and requests the transmission of a unique identifier. An app is installed on the personal device, capable of interacting with the door sensor to pass it the person ID.

Given the reduced operation complexity and the low activation frequency, the door sensor is an embedded device on which a firmware written in C language runs on bare-metal. C language permits low-level control without losing the advantages of high-level statements and data structures. Still, the manual management of data structures and memory pointers is often a source of vulnerabilities. The lack of mem-

ory safety capabilities (such as strong typing, present in other modern languages) enables attackers to exploit these flaws by maliciously altering the program behavior by partially or entirely hijacking control-flow.

Reasonably, the most famous vulnerability for this scenario is *buffer overflow* [9], which is caused by increasing or decreasing a pointer without proper boundary checks on the data structure that is being accessed. This results in out-of-bounds writes that corrupt adjacent data areas, e.g., stack or heap. Similar problems may arise when indexing bugs are present in the code, i.e., boundary checks over an index for a given data structure are missing or incomplete. Indexing bugs are often caused by integer-related errors like an integer overflow, truncation or signedness bugs, or incorrect pointer casting.

In the present use case, the firmware may mistakenly consider the mobile app as a trusted actor. In particular, since the transmitted information has a fixed size in bytes, the sensor may not check the incoming messages, but just read bytes until the string termination symbol is received. An attacker can exploit this vulnerability in various ways, e.g., to steal data or enter the network. Moreover, this vulnerability is an enabling factor for, e.g., bypassing a possible *non-executable-stack* defense [10], and mounting a *Return-Oriented Programming (ROP)* attack [11] [12] [13].

Discussion

A possible solution to preserve control-flow integrity in unprotected devices is by using enforcement mechanisms, e.g., relying on security extensions that consist of adopting a runtime Policy Decision Point (PDP), also called *control-flow monitor*, and several Policy Enforcement Points (PEPs), inserted before runtime into the code. Because of the small capabilities of these devices, this extension must be as tiny as possible in terms of additional computational resources, and must not compromise the real-time nature of the execution. An example is given by [14], where a lightweight solution is presented to protect firmware running on a microcontroller through the use of an external FPGA, which implements a checker. Checks are triggered at critical branch points by a single instruction that invokes the FPGA. Here, the legitimacy of the control-flow transfer is checked through a completely parallel execution, and if necessary, CPU activity is interrupted if a violation is detected.

B. SOFTWARE UPDATE FOR IOT END DEVICES

Low-power indoor sensors (*AirMonitors*) continuously collect data about the air quality inside the ZEB, aiming to detect and prevent bad air quality situations, which could lead, e.g., to a higher risk of COVID transmission.

From the hardware point of view, the AirMonitor prototype present in HArMoNICS bundles a typical COTS System-on-Chip (SoC): an ARM Cortex-M microcontroller communicating with IEEE 802.15.4 low-power radio. This SoC is connected via I2C/SPI bus on-board to a variety of sensors, for humidity, gas, and dust particles. The AirMonitor is networked via a low-power wireless access point, through which it communicates via IP protocols (6LoWPAN and CoAP) with a remote software udpate server (items 2 in Figure 2, in red). The software embedded on AirMonitors is based on RIOT [15], a popular open-source general-purpose operating system for low-power IoT devices.

The idea behind this scenario is to offer a test case for the security of software updates for low-power IoT devices. Over the last few years, the research community has been working on the definition of several IoT update processes [16], among which secure software updates for resource-constrained devices is a challenging research topic [17].

Discussion

The Internet Engineering Task Force (IETF) is currently on defining a new standard for firmware updates called Software Updates for Internet of Things (SUIT) [18] [19]. The main goals of SUIT are interoperability and end-to-end security. The SUIT information model [20] defines a collection of security threats for the update process. Threats associated with this type of scenario are manifold. As discussed in [21], an attacker might update the IoT device with a modified and intentionally flawed firmware image. Alternatively, an attacker may replay a valid, but old (known-to-be-flawed) firmware, or a firmware update that is authentic, but for an incompatible device.

Although the SUIT model suggests a set of security requirements and countermeasures, it is worth noticing that all these threats are related to the integrity and the confidentiality of the update process only, while the content of the update is assumed to be trusted. Therefore, the SUIT workflow allows an Information System Management (ISM) service to upload a firmware image containing security vulnerabilities or malicious behaviors. Furthermore, as demonstrated by some recent work, the SUIT workflow is flexible in that it allows not only pre-quantum, but also post-quantum security [22], and does not only cater for full IoT firmware updates, but also for securing modular software updates on low-power IoT devices [23]. Last but not least, SUIT allows the ISM to transfer its authority to another entity, e.g., a third-party developer, that can deliver to the ISM some components of a software update (e.g., the executable of the application to be updated) or trigger the update process directly. In this case, the ISM has no mechanism to assess the content of the external software components, and must trust the external entity.

C. PRIVACY-PRESERVING DATA PROCESSING

HArMoNICS includes an access control system that is managed by a dedicated server in the `intranet` subnetwork (item 3 in Figure 2, in light green). The system secures the access of persons inside the smart building. It also works as the access control system for drivers who want to use the parking lot of the campus. Access control is the process of mediating every request to data and resources owned by a system and determining whether the request should be

granted or denied. In general, access control is a necessary condition to build privacy in IoT solutions and to comply with the EU General Data Protection Regulation (GDPR) [24].

The are several risks associated with such a scenario, the following being the main ones.

- *Consent unawareness.* This risk relates to a user being unaware of the information disclosed to the system. The user could either provide too much information, e.g., allowing a malicious agent to retrieve her identity, or, on the contrary, inaccurate information, which can lead to wrong decisions or behaviors.
- *Policy and consent non-compliance.* This threat means that, even though the system shows its privacy policies to the users, there is no guarantee that the system actually complies with the advertised policies [25]. Therefore, although policies claim to prevent it, the user's personal data might still be revealed.
- *Information disclosure due to wrong design and/or implementation of access control.* The information disclosure threats expose personal information to unauthorized individuals [25]. This can happen in case the access control mechanisms in place are wrongly designed or implemented. Some modern approaches decompose access control in three main components: policy language, model, and enforcement [26]. Each of them may carry design/implementation errors.

A good privacy-preserving authentication system is an authenticated protocol that does not fully disclose the user identity to a verifier. Only necessary pieces of the user identity (e.g., age, gender, membership etc.) are provided during the verification phase. Furthermore, the authentication sessions should be mutually unlinkable, that is the protocol protects user identity and avoids profiling and tracking of the users.

Another aspect that needs to be taken into consideration within HArMoNICS is how data produced during the use of the II is being handled. The analysis of the usage and transaction logs of the II can provide valuable information on the characteristics and needs of the systems and their users, e.g. services demand, peak hours, and resources sufficiency. However, this type of analysis needs to be performed in a privacy-preserving way, so that the collected data is utilized while remaining protected both from data breaches and unauthorized uses.

Discussion

A promising privacy-enhancing technology that can be used for this purpose is searchable encryption (SE), supporting the storage of usage logs in encrypted form, while data remains available for processing [27]. This data processing system could be applied to the HArMoNICS smart building, as well as the smart parking lot and vehicle charging services. Desired properties of the SE service include:

- *Query expressiveness.* Support for complex, multi-keyword queries is required, in order to enable deriving useful conclusions from the data analysis.
- *Efficiency.* The query functionality needs to be efficient, in order to be applicable and practical in a real-world setting such as HArMoNICS.
- *Dynamic dataset support.* Dynamic updates of the encrypted dataset need to be supported in the system, to respond to the dynamic nature of the II.
- *Multi-client search support.* Enabling external entities to perform authorized queries on the encrypted dataset extends the utilization potential and value of the data processing service.

Searchable Symmetric Encryption (SSE) is a practical variation of searchable encryption providing balance between efficiency, functionality and security, while supporting the aforementioned requirements [28]. An open-source SSE implementation such as Clusion [29], based on the IEX SSE scheme [30], can be used for the implementation of the HArMoNICS privacy-preserving data processing system.

D. INTRUSION DETECTION

Intrusion detection is one of the main components of a global security strategy. In particular, it is typically considered the second line of defense against attacks. Reasonably, a complex II supported by a large-scale network, made of multiple and heterogeneous devices, is unlikely to be completely secure. Despite all the upstream efforts that are made during the design and development of a critical information system, malicious activities may succeed and compromise the confidentiality, availability, or integrity of the system during its life cycle. An Intrusion Detection System (IDS) aims at detecting such attacks against computer systems and networks. To deal with latent threats, an IDS continuously monitors the running system and analyses the gathered information to detect if an attack occurs or not. When the monitoring mechanism suspects that an attack has occurred (or is in progress), an alert is raised.

The presence of an IDS server within `intranet` network of HArMoNICS (item 4 in Figure 2, in purple) is useful to allow the investigation of currently trending challenges for the community. In particular, a crucial challenge is related to the availability of an appropriate dataset which is critical in the development of most IDSs. A bulk of state-of-the-art research does not provide reliable performance results since they rely on either the KDD99 or NSL-KDD benchmark datasets, which is concocted of traffic being over 20 years old. In this way, it does not represent recent attack scenarios and traffic behaviors. Obtaining traffic from simulated environments can help overcome this issue when merged with testing more recent datasets, such as the CICIDS 2017 [31]. Published datasets are available for different domains, such as industrial control systems (ICS) [32]. HArMoNICS can be used for assessing the effectiveness of datasets against real attack scenarios.

Discussion

An interesting direction for dealing with this scenario is to consider solutions using intrusion models that assume events

to belong to a partially ordered set. For each participant of the distributed computation, the input trace describes the sequence of events that occurred locally in the participant's process. An event is a performed action, like sending or receiving a message, but also an internal activity such as a system call. Clearly, for this model to apply, monitoring and logging code must be supported by all the participants, which is often the case.

Based on these traces, each distributed computation can be observed as a partially ordered set of events and be represented by a lattice of consistent cuts. This intermediate representation of learned normal behavior is potentially very large in size. Thus, it is used to infer smaller models that characterize the acceptable sequences of events. These models can take the form of an automaton or a list of temporal properties that have to be satisfied (likely invariants). Several types of models are constructed and used in parallel, as different models are often complementary during the detection phase. Also, using different types of models is a key to reduce false negatives. Furthermore, to reduce false positives during the detection phase, the training phase must consider multiple distinct correct executions: the resulting models are obtained by combining the intermediate models defined during the learning of each normal execution.

E. IOT PROTOCOLS FLAWS

HArMoNICS includes two different scenarios related to the verification of IoT protocols. In the first scenario, the protocol EnOcean [33], mainly used in smart building domain, has been considered (item 5 in Figure 2, in blue). Briefly, EnOcean is used to implement the communications between IoT devices interacting with a smart HVAC (Heating, Ventilation and Air-Conditioning) system. Since IoT devices are provided by different manufacturers, design flows in the EnOcean protocol may have dramatic effects on the correct behavior of the HVAC system. The HVAC hub server is located in the building area network (or local area network) and communicates through a gateway to the Internet and outside users. Via smartphone or tablet the user can get access to the building network and she can monitor or configure the system. In case of an unusual event the user will be notified immediately. Thus, an attacker could exploit flaws in the EnOcean protocol to carry out the following operations.

- Eavesdropping, i.e. spying on system.
- Replay attack, where (parts) of messages are recorded to use it at a later stage.
- Man-in-the-middle attack, where the communication between two communication partners is intercepted, and potentially changed during transmission (modification attack).
- Denial-of-Service attack, i.e., preventing legitimate users from accessing the system.

The second scenario, instead, focuses on a risk analysis of two different smart building system configurations through threat modeling. This scenario carries three elements. The first one is modeling a system, e.g., with a threat modeling tool, in order to identify potential threats and vulnerabilities. The second element includes the attack scenario definition through the exploitation of selected vulnerabilities. The last one amounts to a risk analysis of the attack scenarios.

Discussion

Formal verification allows proving the correctness of a target protocol with respect to a certain specification or property *with mathematical rigor*. Possible checks include verification or falsification of security properties, functional correctness, qualitative and quantitative analysis of protocol specifications or implementations [34] [35], in presence of an attacker.

In the first scenario describe above, the formal verification of EnOcean can be based on ProVerif [36] [37], i.e., a protocol model checker which considers the well know Dolev-Yao attacker model [38]. The first step is the creation of an input model, e.g., based on the protocol specification. Since, in general, model checking the whole protocol specification is computationally expensive, often models include only the most critical parts in the protocol (e.g., initial authentication between the participants) to be verified against the relevant security goals (e.g., authentication happens correctly). Usually, the output of the formal verification process is either a proof of correctness or a potential vulnerability (a.k.a. a counterexample). Since ProVerif also handles unbounded protocol sessions, which are undecidable in general, it may return false positives, i.e., counterexamples/attacks that are not actually executable. For this reason, ProVerif also gives an attack derivation, which helps a human analyst to manually reconstruct an attack.

Instead, the second scenario can be treated via a probabilistic risk analysis of the two system configurations (see above) through model checking [39]. Since the problem to be modeled for this scenario is probabilistic, one may consider the Prism model checker [40]. Briefly, Prism models are specified through various types of Markov chains, where each transition occurs with a giver probability. Assigned probabilities can correspond to the risk/likelihood score of a certain threat in the threat model.

F. FOG ORCHESTRATION SECURITY

The main idea behind this scenario is to check the placement of Fog and Edge devices and services for possible QoS and security-related issues, and find the non-optimal distribution of services between Fog nodes. Two Fog nodes are physically placed in two different locations and both are connected to the `iot` network (item 6 in Figure 2, in yellow). These Fog nodes are capable of running communication services to connect with related edge devices. Fog nodes host services that monitor the lighting characteristics in the rooms using light sensors, and are capable to adjust the lighting according to the preferences of the human by activating smart bulbs. Location and presence services are running on the Fog nodes to sense humans in the room and to monitor their exact position. A decision service "knows" the lighting preferences

of the particular human, and controls smart bulbs according to the position of that human inside the room. For example, if the human sits on the couch near the TV, the lighting should be dimmed, etc.

The goal of this scenario is to provide a testbed for Fog orchestrators, and for measuring their ability to make decisions on controlling the services according to the QoS and security requirements. Each decision of the orchestrator on starting/stopping/suspending/moving Fog services should be checked for the satisfaction of the minimal requirements imposed by various hardware and software restrictions of the involved physical devices, as well as requirements arising from the specifics of the area of application (e.g., health-related data should be protected better than environment monitoring data).

Orchestrators should follow three main steps:

- Each orchestrator should apply requirements on latency, bandwidth, security, and range imposed by the application area and hardware/software capabilities of each Fog node and decide if it is possible to start all required services without violating these requirements;
- Orchestrators should find the optimal distribution of the available services between different Fog nodes: this is useful for saving energy and computation resources in cases when some services may be stopped, suspended or moved to another Fog nodes;
- Dynamic service allocation should be carried out. Dynamic allocation happens when the situation changes at runtime, and orchestrators must change the distribution of the services between available Fog nodes according to the new conditions.

Discussion

The technique for a Dynamic Service Orchestration, which addresses the issues discussed above, is presented in [41]. The control loop is developed on three main consecutive steps: Monitoring, Optimization, and Execution. Optimization aims to place find which placement of n available services in k Fog nodes makes a set of chosen QoS parameters is optimal. The problem can be solved through a system of inequalities for parameters such that objective functions are minimized. Although, the objective functions are contradicting to each other so there is no single solution to this multi-objective optimization problem that optimizes all the objective functions at the same time.

In fact, the process resolves into two further substeps. Integer Multi Objective Particle Swarm Optimization (IMOPSO) method is used to find a set of Pareto optimal solutions. All service placements in this set are non-dominated (Pareto optimal), which means that each of them is better than all the other ones by at least one criterion. The second step is to choose the best solution from the Pareto optimal set by using the Analytical Hierarchy Process (AHP) [42]. AHP uses only a pairwise comparison of all alternatives by all objective functions, is easy to implement and gives consistent results.

G. FOG HARDENING

As previously stated, Fog nodes play an important role in the IoT network of HArMoNICS. As a matter of fact, they serve edge computation for end-users in their proximity, sharing the load on the resources provided by cloud servers. However, this network topology optimization exposes the security of the user- and kernel-space software running on Fog nodes, as they directly interact with the end-user device, which may be malicious (item 7 of Figure 2, in orange). Such an exposure poses security risks which threaten the confidentiality of the data processed by user-space Fog applications, the integrity of the kernel-space Fog operating system, and, inherently, of the whole Fog layer of the ZEB.

Attackers leverage memory corruption vulnerabilities to establish primitives for reading from or writing to the address space of a vulnerable application. These primitives form the foundation for code-reuse and data-oriented attacks [43]. The security enhancement should ensure the confidentiality of sensitive data, such as personal user information or user authentication material, that user-space application running on Fog nodes process. Moreover, the security extension should harden the underlying operating system kernel against data-oriented attacks, preventing an attacker from taking over the Fog node, which would allow her to have a foothold in the II network.

Discussion

With respect to the highlighted issues, virtualization extensions of modern CPUs could be leveraged to establish selective memory protection (xMP) primitives [44], that have the capability of thwarting data-oriented attacks. Such extensions, like Intel's Extended Page Table pointer (EPTP), offer the possibility to manage different views on guest-physical memory from inside a VM, without any interaction with a hypervisor. Therefore, selective protection of sensitive data in user or kernel space is obtained by isolating sensitive data in disjoint xMP domains.

In the specific use case scenario, such a solution can be used to harden the user-space decision service that the fog orchestrator deploys on fog nodes, which adjusts the light level based on user preferences received from their end device. Specifically, the end-user data that the service processes and stores in its address space can be isolated in a dedicated xMP domain. This way, the user information is prevented from being leaked in case an attacker exploits a memory corruption vulnerability that may emerge in the decision service.

H. MANAGING PERSONAL DATA IN VEHICLE RECHARGE PROCESS

HArMoNICS includes a vehicle recharge facility. Its main components are the user's personal device, the smart parking lot, and the power distributor infrastructure (item 8 of Figure 2, in dark green). The driver using her personal device initiates the charging process in the smart parking lot. Once

the charging process is done, the power distributor (based on the smart meter) sends the charged energy amount to be paid.

The scenario focuses on the payment details and on the related security risks. The payment details may include (i) the driver's name, (ii) bank account and/or credit card information, and (iii) authorization to debit/credit the bank account for the service. The process is as follows: the driver submits her payment details to the parking lot, where the charging is happening. Once done, the parking lot initiates to charge by sending an initiation command to the power distribution infrastructure. Then, the infrastructure charges the car, sends information about the charged energy amount for the payment. Also, the power distribution infrastructure informs the parking lot that the charge is completed. After the parking lot sends the payment details to the smart building, the central management allows performing the payment transaction for the charged energy amount. Then, the lot is informed about the success of the payment and it sends the payment transaction receipt to the user's personal device.

A security issue related to the above scenario is that it is not clear how the driver's personal data, i.e., payment details, are handled during the transaction and whether the treatment of the personal data respects the principles of the GDPR. Verification tools and methodologies can help when designing and implementing data management processes as the one described above. For instance, the DPO tool [45] can help to assess how much the described process is compliant with the GDPR and recommend means to achieve this compliance [46]. For example, in the scenario above, it is possible to determine that the data owner is the driver, and that her personal data, i.e., the payment details, are processed by the recharging infrastructure. Analysis of the process using the DPO tool results in a list of non-compliances, such as:

- Consent is missing (GDPR [24], Art. 7)
- Privacy policy is missing (GDPR [24], Art. 13, 14)
- No security measures are present (GDPR [24], Art. 25)
- Processing task is not being recorded (GDPR [24], Art. 30)

Discussion

The various non-compliances with regulations can be resolved through the adoption of several measures. For instance, to solve the issue of consent, the driver must provide consent for processing payment details to the infrastructure, and the infrastructure keeps consent for them. If it is not valid, the infrastructure informs the driver about the invalid consent; otherwise, it proceeds with sending the permission to charge the vehicle.

The process can be made compliant with security and privacy requirements by adopting TLS protocol to send sensitive personal information. Then, public-key encryption must be applied so that payment details are encrypted before sending to the parking lot. The infrastructure decrypts them before performing the transaction.

Last, a specific task in the process must be allocated after each processing task to log details.

IV. RELATED WORK

HArMoNICS represents a case study that collects, within a single infrastructure, technologies that are related to some major security concerns in real environments. In fact, our proposal integrates both emulated technologies and real components, that replicate some attack vectors of interest. Furthermore, the assets of HArMoNICS are grouped within a virtual machine, open and downloadable by the community. For all these reasons, we believe HArMoNICS to be a distinguished and useful asset for the security community.

Although HArMoNICS is not meant to be a *digital twin* (DT) framework, it has a few similarities with this kind of systems. Briefly, according to [47] [48] and [49], a DT is a digital replica of a real infrastructure, whose simulated execution is capable of generating the same amount of information as the original system [50]. In this respect, some of the scenarios of HArMoNICS can be seen as DTs of real systems. However, HArMoNICS is not designed to be generic and reconfigurable, but only for being extended with new scenarios.

Among the existing digital twins and security testbeds, EPICTWIN [51] is a major proposal focusing on smart grids, where the users can deploy real-world attacks and countermeasures. It includes SCADA workstations, PLCs, end devices, and smart meters executed through a combination of simulation/emulation technologies, including VMs and Matlab-Simulink real-time models. Intuitively, even if EPICTWIN is not an alternative to our proposal, it may be used to implement an infrastructure similar to that of HArMoNICS. To the best of our knowledge, such an implementation does not exist. Furthermore, DTs based on EPICTWIN can be composed with HArMoNICS through the technologies discussed in Section II. Beyond EPICTWIN, the reasoning discussed above also applies to other testbeds based on smart grids. Among them, some prominent examples are PRIME [52], the National SCADA testbed [53], and the infrastructure by the Mississippi State University SCADA Security Laboratory [54].

Some other authors have proposed systems for assessing smart infrastructures from a mainly functional point of view. As a consequence, security aspects are often neglected. Remarkably, [55] puts forward a demonstrator staged in a Campus of the West Cambridge university, which resembles the context of HArMoNICS. The work presents a detailed taxonomy for the layers on which an intelligent civil infrastructure must be based, sided by a data systematization model. Another example is [56], where a reference architecture for smart cities, also including a framework for responding to disastrous incidents, is given. Similar reference architectures are given in [57], [58], and [59]. Since they have a different target, none of these systems puts emphasis on IT security aspects and, thus, they cannot be directly compared with our work. In fact, they might even be composed with HArMoNICS to build larger case studies.

Finally, from the point of view of the formalization of security issues, [60] and [61] propose a systematic literature

review on smart buildings and cities, respectively. Although these works do not provide an implementation, the security concerns gathered there partially overlap with those of HArMoNICS. Again, other security scenarios presented in these works can be integrated with HArMoNICS when implemented.

V. CONCLUSION

In this paper, we presented HArMoNICS, an open-source case study based on a virtual replica of a real intelligent infrastructure located in the Savona Campus of the University of Genoa. HArMoNICS provides a series of vertical scenarios related to major security concerns of intelligent infrastructures, e.g., software integrity and upgradeability, privacy-preserving computing, and intrusion detection. The main goal of HArMoNICS is to become a strategic resource for the security community, which can rely on it for setting up experiments and benchmark activities. The infrastructure is currently used as an environment for the demonstration of security techniques (e.g., see [62]) developed in the context of the High-Assurance Intelligent Infrastructure Toolkit program of EU-funded project SPARTA.

In future work, we plan to leverage HArMoNICS for assessing the effectiveness of new methodologies aimed at increasing the security level of the intelligent infrastructures. Moreover, we will consider further security scenarios to enrich HArMoNICS.

VI. ACKNOWLEDGMENTS

Heartfelt thanks go to all colleagues not included in the list of authors but decisive for the activity: Katharina Hofer-Schmitz (Joanneum Research, Austria), Manon Knockaert, Jean-Marc Van Gyseghem (University of Namur, Belgium), Petr Dzurenda (Brno University of Technology, Czech Republic), Tewodros Beyene (Fortiss Research Institute, Germany), Mari Seeba, Jake Tom (University of Tartu, Estonia), Joaquin Garcia-Alfaro, Jean-Max Dutertre, Jean-Luc Danger, Maryline Laurent (Mines-Télécom Institute, France), Michel Hurfin, Ludovic Mé (National Institute for Research in Computer Science and Automation, France), Paolo Prinetto, Alessandro Armando (National Interuniversity Consortium for Informatics, Italy), Giorgio Bernardinetti, Francesco Mancini, Gabriele Restuccia (National Consortium for Telecommunications, Italy), Uwe Roth, Qiang Tang (Luxembourg Institute of Science and Technology), Marek Pawlicki (ITTI Institute, Poland).

REFERENCES

[1] A. Holst, "Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030." https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/, 2021. [Online; accessed June 07, 2021].

[2] N. Antonopoulos and L. Gillam, Cloud computing. Springer, 2010.

[3] C. Matt, "Fog computing," Business & information systems engineering, vol. 60, no. 4, pp. 351–355, 2018.

[4] A. Dogra, R. K. Jha, and S. Jain, "A Survey on Beyond 5G Network With the Advent of 6G: Architecture and Emerging Technologies," IEEE Access, vol. 9, pp. 67512–67547, 2021.

[5] J. C. S. Santos, K. Tarrit, and M. Mirakhorli, "A catalog of security architecture weaknesses," in 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), pp. 220–223, 2017.

[6] Docker, "What is a Container? - App Containerization." https://www.docker.com/resources/what-container, 2021. [Online; accessed December 6, 2021].

[7] P. Lipton, C. Lauwers, M. Rutkowski, C. Noshpitz, and C. Curescu, "TOSCA Simple Profile in YAML Version 1.3," tech. rep., OASIS, February 2020.

[8] D. Documentation, "Overview of Docker Compose." https://docs.docker.com/compose/, 2021. [Online; accessed December 6, 2021].

[9] "CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer." https://cwe.mitre.org/data/definitions/119.html, 2019. [Online; accessed November 25, 2021].

[10] M. Docs, "Data Execution Prevention - Win32 apps." https://docs.microsoft.com/en-us/windows/win32/memory/data-execution-prevention, 2021. [Online; accessed November 25, 2021].

[11] R. Roemer, E. Buchanan, H. Shacham, and S. Savage, "Return-oriented programming: Systems, languages, and applications," ACM Transactions on Information and System Security (TISSEC), vol. 15, no. 1, p. 2, 2012.

[12] N. R. Weidler, D. Brown, S. A. Mitchel, J. Anderson, J. R. Williams, A. Costley, C. Kunz, C. Wilkinson, R. Wehbe, and R. Gerdes, "Return-Oriented Programming on a Cortex-M Processor," in 2017 IEEE Trustcom/BigDataSE/ICESS, pp. 823–832, Aug 2017.

[13] N. R. Weidler, D. Brown, S. Mitchell, J. A. Anderson, J. R. Williams, A. Costley, C. Kunz, C. Wilkinson, R. Wehbe, and R. Gerdes, "Return-oriented programming on a resource constrained device," Sustainable Computing: Informatics and Systems, vol. 22, pp. 244–256, 2019.

[14] N. Maunero, P. Prinetto, G. Roascio, and A. Varriale, "A FPGA-based Control-Flow Integrity Solution for Securing Bare-Metal Embedded Systems," in 2020 15th Design Technology of Integrated Systems in Nanoscale Era (DTIS), pp. 1–10, 2020.

[15] E. Baccelli et al., "RIOT: An open source operating system for low-end embedded devices in the IoT," IEEE Internet of Things Journal, 2018.

[16] A. Kolehmainen, "Secure Firmware Updates for IoT: A Survey," in 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 112–117, 2018.

[17] F. J. Acosta Padilla, E. Baccelli, T. Eichinger, and S. K., "The Future of IoT Software Must be Updated." https://hal.inria.fr/hal-01369681/document, 2016. [Online; accessed August 22, 2022].

[18] IETF, "Website of : IETF SUIT draft architecture." https://tools.ietf.org/html/draft-ietf-suit-architecture, 2019. [Online; accessed November 25, 2021].

[19] B. Moran, H. Tschofenig, H. Birkholz, and K. Zandberg, "A Concise Binary Object Representation (CBOR)-based Serialization Format for the Software Updates for Internet of Things (SUIT) Manifest," Internet Draft, 2021.

[20] IETF, "Website of : Firmware updates for internet of things devices - an information model for manifests draft-ietf-suit-information-model-03." https://tools.ietf.org/html/draft-ietf-suit-information-model-03, 2019. [Online; accessed November 25, 2021].

[21] K. Zandberg, K. Schleiser, F. Acosta, H. Tschofenig, and E. Baccelli, "Secure Firmware Updates for Constrained IoT Devices Using Open Standards: A Reality Check," IEEE Access, 2019.

[22] G. Banegas, K. Zandberg, A. Herrmann, E. Baccelli, and B. Smith, "Quantum-Resistant Security for Software Updates on Low-power Networked Embedded Devices," arXiv preprint arXiv:2106.05577, 2021.

[23] K. Zandberg and E. Baccelli, "Femto-Containers: DevOps on Microcontrollers with Lightweight Virtualization & Isolation for IoT Software Modules," arXiv preprint arXiv:2106.12553, 2021.

[24] European Union, "Regulation (EU) 2016/679 of the European Parliament (General Data Protection Regulation)." https://eur-lex.europa.eu/eli/reg/2016/679/oj, 2016. [Online; accessed October 27, 2021].

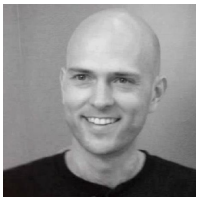
[25] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements," Requirements Engineering, vol. 16, pp. 3–32, Mar 2011.

[26] S. D. C. D. Vimercati, S. Foresti, P. Samarati, and S. Jajodia, "Access control policies and languages," Int. J. Comput. Sci. Eng., vol. 3, pp. 94–102, Nov. 2007.

[27] C. Bösch, P. Hartel, W. Jonker, and A. Peter, "A survey of provably secure searchable encryption," ACM Computing Surveys (CSUR), vol. 47, no. 2, pp. 1–51, 2014.
[28] Q. Gan, C. Zuo, J. Wang, S.-F. Sun, and X. Wang, "Dynamic searchable symmetric encryption with forward and backward privacy: A survey," in Network and System Security (J. K. Liu and X. Huang, eds.), (Cham), pp. 37–52, Springer International Publishing, 2019.
[29] "Clusion library." https://github.com/encryptedsystems/Clusion, 2021. [Online; accessed December 17, 2021].
[30] S. Kamara and T. Moataz, "Boolean searchable symmetric encryption with worst-case sub-linear complexity," in Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 94–124, Springer, 2017.
[31] S. Borah, R. Panigrahi, and A. Chakraborty, "An enhanced intrusion detection system based on clustering," in Advances in Intelligent Systems and Computing, pp. 37–45, Springer Singapore, dec 2017.
[32] R. C. Borges Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari, and S. Pan, "Machine learning for power system disturbance and cyber-attack discrimination," in 2014 7th International Symposium on Resilient Control Systems (ISRCS), pp. 1–8, 2014.
[33] S. Marksteiner, V. J. Exposito Jimenez, H. Valiant, and H. Zeiner, "An overview of wireless IoT protocol security in the smart home domain," in 2017 Internet of Things Business Models, Users, and Networks, pp. 1–8, 2017.
[34] K. Hofer-Schmitz and B. Stojanović, "Towards formal methods of iot application layer protocols," in 2019 12th CMI Conference on Cybersecurity and Privacy (CMI), pp. 1–6, 2019.
[35] K. Hofer-Schmitz and B. Stojanović, "Towards formal verification of iot protocols: A review," Computer Networks, vol. 174, p. 107233, 2020.
[36] B. Blanchet, "An efficient cryptographic protocol verifier based on prolog rules," in Proceedings. 14th IEEE Computer Security Foundations Workshop, 2001., pp. 82–96, 2001.
[37] K. Hofer-Schmitz, "A Formal Analysis of EnOcean's Teach-in and Authentication," in The 16th International Conference on Availability, Reliability and Security, pp. 1–8, 2021.
[38] D. Dolev and A. Yao, "On the security of public key protocols," IEEE Transactions on Information Theory, vol. 29, no. 2, pp. 198–208, 1983.
[39] H. Vallant, B. Stojanović, J. Božić, and K. Hofer-Schmitz, "Threat Modelling and Beyond-Novel Approaches to Cyber Secure the Smart Energy System," Applied Sciences, vol. 11, no. 11, p. 5149, 2021.
[40] M. Kwiatkowska, G. Norman, and D. Parker, "Prism: Probabilistic symbolic model checker," in International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, pp. 200–204, Springer, 2002.
[41] N. Morkevicius, A. Venčkauskas, N. Šatkauskas, and J. Toldinas, "Method for dynamic service orchestration in fog computing," Electronics, vol. 10, no. 15, p. 1796, 2021.
[42] T. L. Saaty and L. G. Vargas, "The seven pillars of the analytic hierarchy process," in Models, methods, concepts & applications of the analytic hierarchy process, pp. 23–40, Springer, 2012.
[43] L. Szekeres, M. Payer, T. Wei, and D. Song, "Sok: Eternal war in memory," in 2013 IEEE Symposium on Security and Privacy, pp. 48–62, May 2013.
[44] S. Proskurin, M. Momeu, S. Ghavamnia, V. P. Kemerlis, and M. Polychronakis, "xmp: Selective memory protection for kernel and user space," in IEEE Symposium on Security and Privacy (Oakland), May 2020.
[45] "Dpo tool - evaluate business process." https://dpotool.cs.ut.ee/, 2020. [Online; accessed December 6, 2021].
[46] R. Matulevičius, J. Tom, K. Kala, and E. Sing, "A method for managing gdpr compliance in business processes," in International Conference on Advanced Information Systems Engineering, pp. 100–112, Springer, 2020.
[47] A. Bolton, M. Enzer, and J. Schooling, "The Gemini principles: Guiding values for the national digital twin and information management framework," Cambridge, UK: Centre for Digital Built Britain.
[48] GE Digital, "Digital twins: The bridge between industrial assets and the digital world.." https://www.ge.com/digital/blog/digital-twins-bridge-between-industrial-assets-and-digital-world, 2017. [Online; accessed November 24, 2021].
[49] H. V. M. Catapult, "Feasibility of an immersive digital twin: The definition of a digital twin and discussions around the benefit of immersion," in UK: HVM Catapult Visualisation and VR Forum, 2018.
[50] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in Transdisciplinary perspectives on complex systems, pp. 85–113, Springer, 2017.
[51] N. K. Kandasamy, S. Venugopalan, T. K. Wong, and L. J. Nicholas, "EPICTWIN: An Electric Power Digital Twin for Cyber Security Testing, Research and Education," arXiv preprint arXiv:2105.04260, 2021.
[52] T. Becejac, C. Eppinger, A. Ashok, U. Agrawal, and J. O'Brien, "PRIME: a real-time cyber-physical systems testbed: from wide-area monitoring, protection, and control prototyping to operator training and beyond," IET Cyber-Physical Systems: Theory & Applications, vol. 5, no. 2, pp. 186–195, 2020.
[53] K. Barnes and B. Johnson, "National SCADA test bed substation automation evaluation report," tech. rep., Idaho National Laboratory (INL), 2009.
[54] T. Morris, A. Srivastava, B. Reaves, W. Gao, K. Pavurapu, and R. Reddi, "A control system testbed to validate critical infrastructure protection concepts," International Journal of Critical Infrastructure Protection, vol. 4, no. 2, pp. 88–103, 2011.
[55] Q. Lu, A. K. Parlikad, P. Woodall, G. Don Ranasinghe, X. Xie, Z. Liang, E. Konstantinou, J. Heaton, and J. Schooling, "Developing a digital twin at building and city levels: Case study of West Cambridge campus," Journal of Management in Engineering, vol. 36, no. 3, p. 05020004, 2020.
[56] D. N. Ford and C. M. Wolf, "Smart cities with digital twin systems for disaster management," Journal of management in engineering, vol. 36, no. 4, p. 04020027, 2020.
[57] J. Dutta and S. Roy, "IoT-fog-cloud based architecture for smart city: Prototype of a smart building," in 2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence, pp. 237–242, 2017.
[58] G. Alsuhli and A. Khattab, "A Fog-based IoT Platform for Smart Buildings," in 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), pp. 174–179, 2019.
[59] Z. Chevallier, B. Finance, and B. C. Boulakia, "A Reference Architecture for Smart Building Digital Twin," in SeDiT@ ESWC, pp. 1–12, 2020.
[60] P. Ciholas, A. Lennie, P. Sadigova, and J. M. Such, "The security of smart buildings: a systematic literature review," arXiv preprint arXiv:1901.05837, 2019.
[61] J. Laufs, H. Borrion, and B. Bradford, "Security and the smart city: A systematic review," Sustainable cities and society, vol. 55, p. 102023, 2020.
[62] L. Malina, P. Dzurenda, S. Ricci, J. Hajny, G. Srivastava, R. Matulevičius, A.-A. O. Affia, M. Laurent, N. H. Sultan, and Q. Tang, "Post-Quantum Era Privacy Protection for Intelligent Infrastructures," IEEE Access, vol. 9, pp. 36038–36077, 2021.

GIANLUCA ROASCIO is a PhD Student at the Department of Control and Computer Engineering at Polytechnic of Turin and member of the Cybersecurity National Laboratory of CINI, the Italian National Consortium for Informatics. He got his Master Degree in Embedded Systems in 2018, when he started his research career. Since 2020, he is lecturer of Computer Security at the Italian Army Academy. His research topic is security of hardware devices and of the IoT. He is mainly focused on memory protection and control-flow integrity.

GABRIELE COSTA is Associate Professor at the SySMA Group of IMT School for Advanced Studies and member of the Cybersecurity National Laboratory of CINI. He received his Laurea degree in Computer Science in 2007 and his Ph.D in Computer Science in 2011. He was member of the cybersecurity group of the Istituto di informatica e Telematica of the CNR. His appointments include a period as visiting researcher at ETH Zurich in 2016-2017. He has been co-founder of the Computer Security Laboratory (CSec) at DIBRIS. He is co-founder and CRO of Talos, a startup focusing on cybersecurity.



EMMANUEL BACCELLI is Professor for "Open and Secure IoT Ecosystem" at Freie Universität Berlin, in partnership with Inria and the Einstein Center Digital Future (ECDF). He received his PhD in 2006 in Paris from École Polytechnique. In 2012, he completed his habilitation at Université Pierre et Marie Curie. Since 2007, Emmanuel Baccelli has joined Inria as scientific researcher, where he currently takes part in the project-team TRiBE (Inria is the French national research institute for digital sciences). Since 2013, Emmanuel Baccelli is also co-founder and coordinator of the open source community developing RIOT Operating System.

LUKAS MALINA received the M.Sc. degree (Hons.) and obtained the Dean prize for master's thesis from the Brno University of Technology (BUT), in 2010, and the Ph.D. degree from BUT in 2014. He is currently a Senior Researcher with the Department of Telecommunications, BUT. He has published more than 70 papers in international journals and conferences and has provided several invited research and teaching lectures abroad, i.e., in Finland (University of Tampere, 2013), Spain (URV Tarragona, 2015), Russia (St. Petersburg ITMO, 2017), Belgium (KU Leuven, 2017). He is currently leading Privacy-by-Design Task in SPARTA H2020 project.

RAIMUNDAS MATULEVIČIUS received his Ph.D. in computer and information science from the Norwegian University of Science and Technology. He is currently Professor of Information Security at the University of Tartu, Estonia. His publication record includes more than 100 articles published in peer-reviewed journals, conferences, and workshops. He has been a program committee member at international conferences, i.e., NordSec, PoEM, REFSQ, and CAiSE. He is an author of the book Fundamentals of Secure System Modelling (Springer, 2017). He is currently involved in the SPARTA H2020 project (task: Privacy-by-Design).

MARIUS MOMEU is a PhD student at Technical University of Munich. He got his Master Degree in 2020. His research deals with software memory corruption vulnerabilities, with particular emphasis on operating system kernel hardening. Further research interests include automated testing through symbolic execution, software mitigations for architectural flaws and secure e-voting. He is currently leading the "Hardening Legacy Components" task in SPARTA H2020 Project.

NERIJUS MORKEVIČIUS is an Associated Professor at the Kaunas University of Technology (KTU). He received the PhD in Computer Science in Kaunas University of Technology department of Applied Mathematics in 2002. He has taken part in several national and EU funded research projects, has a number of publications in high impact international conferences and journals. His main research interests are information management systems, IoT technologies, cybersecurity and information systems security.

ENRICO RUSSO is Assistant Professor in Computer Engineering at the University of Genoa. He received his M.Sc. in Computer Science in 2001 and his Ph.D. in Computer Science in 2021. His work is focused on Cyber Range systems with particular emphasis on investigating techniques for simplifying the generation of the training environments and for automatizing the tasks of personnel involved in exercises. He also co-founded ZenHack, the Capture the Flag team of the University of Genova, and is personally involved in different live-fire cyber exercises as a member of the Green/Blue team.

BRANKA STOJANOVIĆ is a key researcher and the deputy head of the Cyber Security and Defence Research Group at Joanneum Research Digital – Institute for Information and Communications Technologies, Austria. She graduated in Electrical Engineering and obtained Ph.D. degree in Electrical Engineering and Computer Science, from the School of Electrical Engineering (ETF), University of Belgrade, Serbia. She is CISSP certified. Her field of research activities covers cybersecurity, artificial intelligence, pervasive computing, biometrics and computer vision. Recently, her work focuses on formal modeling of protocols for the IoT. With this topic, she is included in the "Secure Orchestration" task of the SPARTA H2020 Project.

AIMILIA TASIDOU works as a post-doctoral researcher at Télécom SudParis, Polytechnic Institute of Paris, France. She holds a PhD in Data Privacy from the Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece. Prior to her PhD, she received a BSc in Informatics from the University of Piraeus, Greece, and an MSc in Artificial Intelligence from the University of Edinburgh, UK. She has worked on several research projects, in Greece, UK and France. Research experience includes work on anonymization techniques for real-world datasets, privacy-preserving utilization of personal data within personalized services, as well as context-aware systems.