future internet

Article

A Methodological Approach to Securing Cyber-Physical Systems for Critical Infrastructures

Antonello Calabrò [1], Enrico Cambiaso [2], Manuel Cheminod [2], Ivan Cibrario Bertolotti [2], Luca Durante [2], Agostino Forestiero [3], Flavio Lombardi [4,*], Giuseppe Manco [3], Eda Marchetti [1], Albina Orlando [4] and Giuseppe Papuzzo [3]

[1] Istituto di Scienza e Tecnologie dell'Informazione (ISTI), Consiglio Nazionale Delle Ricerche (CNR), 56124 Pisa, Italy
[2] Istituto di Elettronica e di Ingegneria dell Informazione e delle Telecomunicazioni (IEIIT), Consiglio Nazionale Delle Ricerche (CNR), 10129 Torino, Italy; luca.durante@cnr.it (L.D.)
[3] Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR), Consiglio Nazionale Delle Ricerche (CNR), 87036 Rende, Italy
[4] Istituto per le Applicazioni del Calcolo (IAC), Consiglio Nazionale Delle Ricerche (CNR), 00185 Roma, Italy
* Correspondence: flavio.lombardi@cnr.it

Abstract: Modern ICT infrastructures, i.e., cyber-physical systems and critical infrastructures relying on interconnected IT (Information Technology)- and OT (Operational Technology)-based components and (sub-)systems, raise complex challenges in tackling security and safety issues. Nowadays, many security controls and mechanisms have been made available and exploitable to solve specific security needs, but, when dealing with very complex and multifaceted heterogeneous systems, a methodology is needed on top of the selection of each security control that will allow the designer/maintainer to drive her/his choices to build and keep the system secure as a whole, leaving the choice of the security controls to the last step of the system design/development. This paper aims at providing a comprehensive methodological approach to design and preliminarily implement an Open Platform Architecture (OPA) to secure the cyber-physical systems of critical infrastructures. Here, the Open Platform Architecture (OPA) depicts how an already existing or under-design target system (TS) can be equipped with technologies that are modern or currently under development, to monitor and timely detect possibly dangerous situations and to react in an automatic way by putting in place suitable countermeasures. A multifaceted use case (UC) that is able to show the OPA, starting from the security and safety requirements to the fully designed system, will be developed step by step to show the feasibility and the effectiveness of the proposed methodology.

Keywords: cybersecurity; monitoring; firewalling; rule distribution; slow DoS attack; denial of service; industrial security; critical infrastructure protection; security investments.

check for updates

Citation: Calabrò, A.; Cambiaso, E.; Cheminod, M.; Bertolotti, I.C.; Durante, L.; Forestiero, A.; Lombardi, F.; Manco, G.; Marchetti, E.; Orlando, A.; et al. A Methodological Approach to Securing Cyber-Physical Systems for Critical Infrastructures. *Future Internet* **2024**, *16*, 418. https://doi.org/10.3390/fi16110418

Academic Editors: Massimo Cafaro, Italo Epicoco, Marco Pulimeno and Paolo Bellavista

Received: 2 September 2024
Revised: 29 October 2024
Accepted: 5 November 2024
Published: 12 November 2024

1. Introduction with Grand Challenge

The emergence of cyber-physical systems has ushered in unprecedented levels of automation and interconnectivity across critical infrastructures. Historically, Operational Technology (OT) systems strongly emphasized operational safety and robustness but often lacked the advanced security protocols in Information Technology (IT) systems. In today's highly interconnected world, critical infrastructures such as power grids, transportation systems, water treatment plants, and healthcare facilities increasingly rely on modern Information and Communication Technology (ICT) infrastructures. These infrastructures, therefore, combine both Information Technology (IT) and Operational Technology (OT), forming complex cyber-physical systems (CPSs).

Integrating these systems improves efficiency, automation, and scalability, but also brings about a new set of security and safety challenges. Indeed, critical infrastructures are exposed to new forms of cyberattacks that exploit vulnerabilities in both the physical and digital domains. For example, incidents like STUXNET [1,2] highlighted the devastating potential of cyberattacks targeting industrial control systems, resulting in widespread damage to physical systems.

Additionally, cyber-physical systems that manage critical infrastructure have become attractive targets for cyberattacks, posing significant national and economic security risks. Their reliance on interconnected systems, data sharing, and instantaneous operational responses makes them susceptible to various threats, including network breaches, malware, distributed denial of service (DDoS) attacks, and complex cyber espionage campaigns. Consequently, as security concerns within these infrastructures become increasingly prominent, integrating IT, OT, and CPS introduces new layers of complexity. For instance, OT, traditionally focused on safety and reliability, must now adopt IT security measures, including data protection, encryption, and threat detection. However, this amalgamation often leads to decreased performance, operational delays, and reduced system efficiency, prompting the following important questions:

RQ1: How can these infrastructures be secured without compromising their performance?

RQ2: How can systems be developed to identify real-time emerging threats, mitigate their impacts, and recover promptly?

These urgent issues present a significant challenge for researchers, engineers, and cybersecurity professionals dedicated to protecting critical infrastructure.

To tackle these questions, this paper presents a comprehensive methodological useful approach to design and preliminarily implement an Open Platform Architecture (OPA) to secure critical infrastructure cyber-physical systems. The proposed OPA endeavors to establish a flexible and scalable architecture that integrates existing and newly developed systems seamlessly. It offers advanced security monitoring, threat detection, and automated response mechanisms. The architecture focuses on real-time threat management, enabling autonomous anomaly detection, response, and implementation of self-healing measures to mitigate risks. Emphasizing an open and modular component (hence, the name Open Platform), the proposed architecture allows for the integration of heterogeneous security technologies, ensuring long-term adaptability and resilience against emerging cyberthreats.

As shown in Figure 1, the proposed OPA is designed to serve as a foundational blueprint for developing and deploying secure ICT systems in critical infrastructures. It provides a structured approach for integrating advanced security measures, ensuring that system security is treated as a dynamic and evolving feature rather than a static one. The architecture enables critical infrastructures to perform the following:

- **Monitor and Collect Data:** The OPA system has comprehensive monitoring tools designed to continuously gather data from various system components, including network traffic, user activity, and resource consumption. These data play a critical role in facilitating real-time analysis and the identification of potential threats.
- **Detect Security Threats:** The OPA utilizes predefined thresholds and advanced machine learning algorithms to identify anomalies and discrepancies between expected and actual system behavior. These detection mechanisms can recognize emerging threats like unauthorized access attempts, data manipulation, and abnormal usage patterns, ensuring the early detection of potential attacks.
- **React and Adapt in Real-Time**: The OPA facilitates the automatic reconfiguration of security measures in response to identified threats. This capability enables the system to independently adjust to changing cyber threats by implementing suitable countermeasures, including firewalls, encryption, and user access limitations. Additionally, it includes self-healing functionality, which allows the system to recuperate from security breaches without requiring human intervention.
- **Provide Human Oversight**: While automation is crucial for promptly addressing cyber threats, the OPA also incorporates measures for human oversight. Critical decisions, such as significant system reconfigurations or the implementation of specific countermeasures, as well as decisions on security investments based on the allo-

cated budget, can be reviewed and approved by humans, thus maintaining a balance between automation and control.

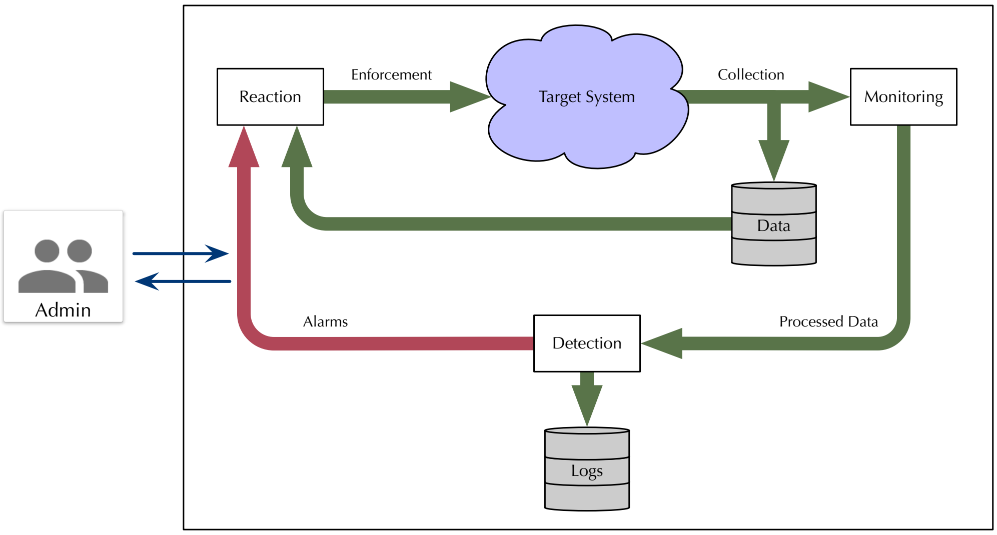

Figure 1. Concise Open Platform Architecture (OPA) (updated with the addition of admin).

In Figure 1, the proposed OPA enhances the target system (TS) with the mentioned features. It also includes data repositories for storing data and logs for offline analysis, like forensic analysis and formal analysis based on a system model. The OPA components can be integrated into the TS nodes (e.g., monitoring agents within the TS nodes) with support from the TS itself. All capabilities, techniques, mechanisms, and repositories are equipped with well-defined, harmonized, and shared interfaces, rendering the architecture a versatile platform amenable to further extensions.

Figure 1 illustrates how the OPA allows the administrator to specify that specific actions, such as selecting between alternative remedies or implementing substantial reconfigurations of the TS, may necessitate human oversight and authorization. Notably, the OPA distinctiveness lies in its adept management of critical, intricate, and dispersed infrastructure, achieved through integrating automated and autonomous detection and response agents with human supervision.

The OPA's instrumentation components (monitoring, detection, and reaction) are designed to be protected from cybersecurity threats. They are distributed and potential targets of attacks. OPA components will be designed to operate on the original TS and the combined system, including the OPA components. This will create a self-protecting and self-healing system.

This paper presents a multifaceted use case to show that the proposed OPA is feasible and effective. In the following sections, readers will be guided through the process of securing a complex critical infrastructure system, step by step. The use case covers the following key aspects:

- Identifying security and safety requirements.
- Designing and implementing the OPA to meet these requirements.
- Validating the system's ability to detect and respond to cyber threats in real time.

The use case draws on real-world scenarios involving a Mobile Control Panel (MCP), an IoT framework, and a Robotics Operating System (ROS) testbed, demonstrating how the OPA can be adapted to different types of cyber-physical systems and how it can be considered a foundational guide for cybersecurity professionals, researchers, and system designers tasked with protecting the critical infrastructures that are vital to modern society.

The above goals are achieved by precisely describing the needed steps, i.e., the methodology itself, leading to building the OPA: We start with an abstract view of the OPA and move on to describe the state of the art (SoA) in Section 2, which is structured by taking into account the building blocks of Figure 1. Then, Section 3 extends beyond the SoA by listing the scientific and technical challenges that need to be addressed in order to reach the above-listed goals. These challenges require some help from the TS, that is, Section 4 depicts the expected support from the TS that is needed to fulfill all the OPA requirements. Finally, Section 5 illustrates the practical application of the proposed architecture, which is the last step of the proposed methodology, whereby a TS is able to be transformed into a secure target system. We would not be able to describe the overall implementation of the proposed architecture without emphasizing its peculiarities; moreover, the main goal of the paper is to show the overall methodology. For these reasons, this section presents the implementation of the proposed architecture in three realistic (sub-)target systems: a Mobile Control Panel (MCP), an IoT framework, and an ROS testbed. Section 6 draws some conclusions.

2. State of The Art

This section provides an overview of the recent literature contributions on activities categorized according to the proposed architecture (Figure 1).

Regarding the monitoring and detection pre-setup, we first refer to slow DoS characterization and detection. Considering denial of service (DoS) attacks, if we exclude exploit-based threats, which are often able to target specific software versions, we can distinguish between flooding-based DoS threats [3], which typically require the consideration of attack resources, and slow DoS attacks [4], which are characterized by the use of minimum amounts of attack resources. In the slow DoS context, although [4] exhaustively contributes to the description of the threat and the possible approaches that a malicious user could exploit, specific attacks of such kind have been introduced in the literature. To this aim, the authors of [5] designed attacks that are able to target specific communication protocols. By focusing on the attacking nodes, [6] considers slow DoS attacks executed from mobile devices, showing that it is actually feasible to perpetrate DoS attacks against a network service by using as little as a single mobile device.

As far as behavioral monitoring is concerned, runtime monitoring systems are receiving increasing attention in various applications because they are considered an effective means for addressing or predicting behavioral patterns and properties using algorithms and analytics. Usually, runtime monitoring systems rely on an event-based data sent by the target environment's component, sensors, or subsystem when specific conditions occur. These kinds of systems can be used to improve the quality of service (QoS), prevent or reduce violations of properties, and provide recovery mechanisms in case of detected problems [7,8] especially in dynamic environments where unpredictable events may occur. Usually, the runtime monitor includes a rule-based complex event processor [9] that collects and analyzes data during the execution according to the concept of observability [10]. Researchers have explored runtime monitoring in various domains, such as industrial automation, healthcare monitoring, and smart cities. Available proposals include monitoring based on logical or axiomatic approaches [11], monitoring focused on instrumentation algorithms for asynchronous components [12], monitoring relying on model-driven specification [13,14] or based on an event-driven approach [9,10].

In this context, machine learning (ML) plays an important role as it allows for identifying complex patterns faster than humans, especially with large datasets, making it valuable for detecting anomalies in behavioral monitoring [15]. Its application in environments requiring real-time, continuous, and reliable processing, such as the Internet of Things (IoT), has produced important and positive results [16]. Indeed, ML algorithms, through supervised and unsupervised learning, enable data analysis and predictions without explicit programming [17]. Despite its effectiveness in threat detection, enhancing the explainability of ML decisions, especially in IoT security, remains a critical area of ongoing research to make outcomes understandable at a human level [18]. Explainable AI (XAI), a specialized domain within machine learning (ML), focuses on refining the processes of training, learning, and representation to yield explanations that can be easily understood by humans [19]. Enhancing the transparency of a model's decision-making processes facilitates the validation of key attributes such as privacy, fairness, and trust [20].

Another relevant issue is the requirement to increase computing and data security and safety with acceptable performance loss. Most approaches involve making use of advanced abstract machines [21] in order to balance safety, security, and performance requirements. In particular, hardware virtualization allows multiple tenants to share resources with a higher degree of security and isolation than containerization [22]. In fact, virtualization enables the protected transparent introspection of the activity and content of virtual machines, allowing for more robust monitoring and control. In fact, most approaches allow for the use of introspection techniques for monitoring intrusions and anomalies by analyzing the system memory and checking its consistency [23]. Hyperupcalls [24] allows a hypervisor to execute in a safe way verified code that is provided by a guest virtual machine to protect data transfers. Hyperupcalls have complete access to guest data structures, potentially allowing for more detailed debugging and monitoring. KVMIveggur [25] offers isolation leveraging containers and virtual machines to enable access control for VMI in cloud environments. It supports continuous monitoring, allows using both active and passive VMI, and can have a very low overhead.

With regard to monitoring and detection execution, we refer to ontology-based monitoring. It is based on utilizing ontologies, which are structured representations of knowledge, to enhance the monitoring of various systems, including, but not limited to, IoT (Internet of Things) devices, networks, and applications. In recent years, substantial research has been conducted on ontology in the representation of different realities and contexts such as DOLCE [26] or BFO, or possibly domain-specific representation in order to describe specific domains or individual systems [27].

All in all, in the last decade, the individual topics mentioned so far have been seen as facets of a comprehensive, unified, and automated cyber-defence architecture [28–30], thus progressing in the same direction as the OPA discussed in this paper. In particular, the OPA being proposed is aligned with the SANS Security Controls 4, 6, 10, 11, 13, 16, and 19 discussed in [28] and is coherent with the high-level NATO Cyber Defence framework also presented therein. The incorporation of machine learning elements also makes the OPA ready to support next-generation cybersecurity paradigms leveraging artificial general intelligence, like the one proposed in [31].

It is worth highlighting that effective cybersecurity is more than a technical matter, as it involves the whole business and the economic aspect of it is very important. Indeed, huge financial losses can arise from a cyber incident, including damages to tangible and intangible assets and liabilities to third parties such as customers, suppliers, employees, and shareholders. A key point is to optimally invest money in order to manage cyber risk, given a certain budget. The peculiarity of cyber risk makes this goal challenging Dacorogna and Marie [32]. A first step to mitigate cyber risk is to raise the level of security by investing in self-protection. Many contributions concern the use of risk metrics borrowed from economic and financial contexts to evaluate these investments; among which are [33–35]. Another tool to mitigate cyber risk is cyber insurance, a risk transfer option which allows one to shift residual cyber risks to an insurance company. Comprehensive existing reviews concerning state of the art, trends, and future directions of this kind of insurance include [36,37].

Finally, as far as reaction means are concerned, firewalls (FWs) are widely deployed popular software and hardware components that (a) block unwanted traffic forwarding, (b) prevent malicious messages from reaching their targets, and (c) can deploy attack countermeasures [38].

Nevertheless, FWs may well become performance bottlenecks and/or targets, e.g., denial-of-service (DoS) attacks. Mitigating approaches to this problem exist, either by operating within a single FW, i.e., *intra-firewall* techniques, or by relying on network-level approaches leveraging multiple FWs, i.e., *inter-firewall* approaches. Some intra-firewall approach include (optimal) rule ordering [39,40], rule analysis [41,42], and firewall compression [43]. Oftentimes, special architectures such as parallel firewalls [44] support inter-firewall approaches. Alternatively, the transfer of filtering rules among firewalls located in the same network can be adopted [45,46]. In particular, the last proposal is very interesting as it is conceptually simple and computationally very efficient, but there is a technical gap to be filled: the firewalls considered in those papers are not real devices, but generic abstract models of firewalls, such as the widely used one introduced by [47], whereas the deployment of the above techniques deals with real devices, such as IPTABLES [48].

The framework provided by the description of the state of the art confirms that scientific research on these topics is very rich. Nevertheless, many open challenges must be addressed to achieve increasingly satisfactory results.

3. Open (Sub-)Challenges

In this section of the paper, we report a list of the open challenges we have identified referring to the considered scenario and how it is possible to enhance its security. In detail, we first consider transversal security challenges affecting IT and OT systems, and then discuss challenges related to monitoring, detection, and reaction aspects.

3.1. Transversal Challenges

Privacy and Security. Ensuring security and privacy is crucial within these extensively automated IoT networks. However, the constrained resources available on IoT devices, coupled with stringent operational requirements, often make traditional security solutions inadequate. Security strategies for IoT must consider the limited computational power, radio bandwidth, and energy resources of these devices while ensuring the efficient processing of large volumes of data and quick response times. Furthermore, security measures that may initially be effective can become obsolete as attackers continuously evolve their tactics to bypass detection mechanisms. Consequently, the unique requirements of IoT systems demand the development of innovative, sophisticated, and customized security approaches to address the wide range of security challenges they face.

Economics of Cybersecurity. As stated in Section 2, effective cybersecurity must take economic aspects into account. Indeed, the special feature of cyber risk poses many open challenges. A first issue in quantifying cyber risk from an economic and financial perspective is to estimate the frequency and severity of cyber incidents. Indeed, the availability of data on financial losses due to cyber incidents is a challenging issue for several reasons: cyber events and data loss-related data are scarce and usually not available in the desired granularity and amount [49,50]. Further, cyber threats evolve in a highly non-stationary cyber risk landscape. Aggregated cyber risks are due to common IT architectures or complex interconnections that are difficult to capture [32]. Even the term 'cyber' risk can refer to many heterogeneous types of risk with different types of impact and, ultimately, root causes [51].

Regarding security investments, the key quantity is the ratio of cost to benefit. In terms of a production function, it would be the amount of output per unit of input. An open challenge is the quantification of the benefits deriving from a specific investment, the security level of the information system, and the suitable metrics to compare different investments strategies [52]. It follows that we need to quantify the monetary cost of security mapped with respect to the security level, and then we need to assess the benefits stemming from it. Finally, to compare different investment strategies, we need to choose suitable security metrics. An open challenge here is the selection of a suitable risk reduction strategy which can be economically affordable.

Another point highlighted in Section 2 is that, aside from risk mitigation by investing in countermeasures, the financial risk of security incidents can be transferred to third parties, notably cyber-insurers. Cyber insurance can be seen as complementary to investing in protection measures. The open challenge here is to find reliable criteria to determine whether it is worthwhile to sign an insurance policy. In principle, if the premium is lower than the difference between the benefit and cost of security, this can be considered a viable investment option [53].

3.2. Monitoring Challenges

Ontology-based Monitoring. As mentioned in Section 2, current research focuses on behavioral or ontology-based monitoring. However, despite the growing interest, some practical challenges still need to be addressed to make the monitor an effective tool for predicting and addressing behavioral patterns and properties. These issues can be summarized as follows:

- *CH1: White-box and black-box assessment*: Distinguishing between white-box and black-box assessment processes is paramount. The objective is to gather internal execution data (white-box) from critical infrastructure components without delving into their source code structure (black-box). This methodology promotes transparency and facilitates the evaluation of both functional and non-functional properties, all while maintaining alignment with the principles of loose coupling and implementation neutrality.
- *CH2: Data collection vs. data management*: It is crucial to differentiate monitoring data collection features from those necessary for knowledge management and validation. This distinction is vital for smart collection classification and prediction, assisting in development and assessment activities. By clearly delineating these aspects, the monitor can enhance its ability to categorize and predict data effectively, contributing to a more innovative and informed approach to development and assessment tasks.
- *CH3: Managing countermeasures*: The knowledge management process enables the customization of manual or automatic countermeasures tailored to risk analysis results. This flexibility empowers the mitigation of vulnerability detection risks during monitoring activities. By incorporating this capability, the system gains the advantage of adapting countermeasures in response to the identified risks, enhancing overall security measures.

Advanced Secure Monitor. A significant obstacle to virtual machine introspection is that it requires highly privileged access to the host system. Current hypervisors see the virtual machine as a black-box. As such, the information which the hypervisor can access is quite limited (this problem is known as a semantic gap). Monitoring through introspection a large application or a full VM software stack can prove overkill for performance, and its usage is very limited. [22] In fact, introspecting containers and Unikernels can also be hard, due to the compressed nature of their memory footprint. Nevertheless, the limited amount of occupied memory can also provide some benefits as it limits monitoring scope [54]. Further investigating these approaches is vital to allow for more protected and secure computations leveraging actual CPU capabilities.

3.3. Detection Challenges

Slow DoS Attack Detection. A relevant aspect to consider when protecting from slow DoS attacks is related to the behavior of the attacks themselves. In particular, from one side, such threats are characterized by low-bandwidth rate [4]. Therefore, it is not trivial to detect an ongoing attack by monitoring the overall network flow [55,56]. In addition, as such threats typically alternate periods of activity to period of sleep, such behavior may resemble the behavior of legitimate clients actively communicating with the server (e.g., to ask for a specific resource), hence temporarily interrupting the communication (e.g., to read the received content). Moreover, if we consider the payload exchanged during the attack between the client(s) and the server, such payload is compliant to the exploited protocol [57], as no malformed packets or byte sequences are sent by the attacker. Finally, by considering the connection peaks which characterize the execution of a slow DoS attack—although at first sight, such peaks may identify a malicious activity—it is important to consider that very similar peaks characterize legitimate flash crowds [58]. In virtue of the considered aspects, to the best of our knowledge, two aspects are today considered open research challenges in the slow DoS attack scenario: on one hand, it is the design of effective intrusion detection algorithms that are able to identify ongoing threats [59]; and on the other hand, it is the design of novel attacks affecting specific protocols or contexts [60], such as Industry 4.0 environments.

Data Analysis and Interpretation. Behavioral data from smart devices often include high-dimensional datasets that are dynamic and evolve over time. Analyzing these data to extract useful information and produce *interpretable* results requires sophisticated machine learning models and real-time processing capabilities [31]. Developing algorithms that can accurately identify and interpret user behaviors without significant errors is challenging. This involves reducing false positives (incorrectly identifying a behavior that did not occur) and false negatives (failing to identify an actual behavior), which are critical for applications like fraud detection or personalized recommendations.

3.4. Mitigation Challenges

Automatic Firewall Reconfiguration. Firewall filtering operates by evaluating each incoming packet against a predefined set of rules. As a result, the packet is either permitted to pass through or blocked. The effectiveness of this filtering is significantly influenced by the average number of rules that each packet must navigate. When there is a significant surge of incoming traffic, such as during a denial of service (DoS) attack or a temporary spike in traffic, the packet processing delay may increase substantially. This situation can lead to packet loss as well.

One potential remedy might involve redistributing rules across multiple cascaded firewalls, which could help maintain a high rate of packet processing, even when the firewall is under attack. A promising strategy for this has been detailed in reference [46]. However, it utilizes the same abstract model of stateless firewalls found in several foundational studies [47,61], indicating that it cannot be directly applied as is; there remains a need to bridge the gap between the abstract, stateless models and actual stateful firewalls used in practice.

In particular, IPTABLES [48], which is one of the most recognized and widely used firewall technologies, is also open source. This makes it a primary candidate to address the aforementioned gap.

The key differences between the *abstract firewall model* in references [46,47,61] and IPTABLES that we must carefully consider to achieve our objective are as follows:

1. The abstract firewall model consists of a singular list of rules, which are sequentially examined against certain fields of the packet header until a match is found with a rule that specifies an action of *ACCEPT* or *DROP*. In contrast, IPTABLES evaluates packets against a more complex framework of rule lists organized into *chains* and *tables* [48], following a specific pathway deeply outlined in [62], where the *Packet Flow in Netfilter and Generale Networking's* well-known diagram depicts all paths foreseen for a packet entering an IPTABLES firewall. In general, the chosen path depends on the source and destination IP addresses: whether the packet originates from outside or from the firewall itself, and whether the recipient is external or the firewall itself. On the other hand, the mitigation challenge proposed here shall only deal with packets originating from outside the firewall, and whose recipient is external. For this reason, Figure 2 only shows the packets' path of interest for us through the *packet processing diagram* (PPD), a subset of that which is depicted in [62]. In particular, each element of the flow followed by a packet entering an IPTABLES firewall is a sequence of rules belonging to the *chain* whose name is prefixed by *C:* and contained in the table listed in the name that is prefixed by *T:*. The two elements outlined by the red line highlight the sequence of rules which will be addressed by the mitigation challenge proposed here.
2. The *abstract firewall model* operates on the assumption that only the packet header fields are compared to each rule, meaning that it acts as a *stateless firewall*. On the other hand, IPTABLES supports more advanced checks, such as determining whether the packet is part of an established connection, and can enact actions that alter the *netfilter* status that has already been assessed.

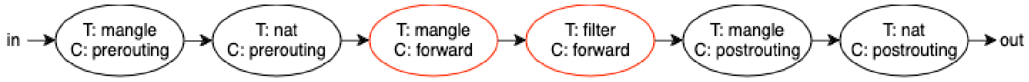

Figure 2. Packet processing diagram (PPD).

The challenge lies in adapting the algorithm from reference [46] for IPTABLES while considering the unique characteristics and features outlined above.

4. Reference Architecture

The architecture presented provides a large picture of the main components of the proposed system that can be instrumented as needed to allow (amongst other features) for monitoring, detection, and reaction to unforeseen and/or unwanted issues as much as possible in an automatic fashion.

As mentioned above, the OPA/Reference Architecture sketched in Figure 1 represents the main components that have to be combined with the TS in order to analyze and protect it.

In fact, the OPA/Reference Architecture must be able to interact with the TS to gather runtime data, such as network traffic, user and agent activities, resource consumption, and so on. The OPA must also be able to react to the potential mismatches and anomalies. In fact, as introduced above, some components of the OPA can also be embedded in the TS nodes (e.g., monitoring agents running in nodes of the TS).

These details impose some requirements on the TS that will be detailed below to better narrow the range of actual devices and systems that can actually host/integrate and benefit from our proposed approach(es).

Target System Requirements

The techniques, methodologies, and approaches introduced in Sections 2 and 3 apply to a broad class of TSs that are nevertheless required to implement some widely available technology. The main requirements that the TS shall satisfy for this purpose are listed in the following.

In regard to the Advanced Secure Monitor (ASM) component, which allows for the simple but effective management of specific services and, as a consequence, potentially entire servers, some support is required in the underlying hardware. In fact, this would include one or more of the following:

- (Open Source) Virtualization Technology aimed at supporting effective system and service isolation;
- (Well-Defined) Semantic introspection capabilities (i.e., the possibility of being able to understand the inner structure and workings of the components);
- (Restricted/Protected Access) Computing environments such as kernel-based (e.g., eBPF) or CPU-assisted ones (e.g., Enclaves/ringlevels).

ASM instantiation can be potentially performed on most virtualization-capable TSs. The TS can be a network-exposed service where both the actual provided service and the monitoring system integrity could be protected by leveraging advanced Secure Transparent Virtualization capabilities. The ASM requires the TS to be able to perform multiple simultaneous activities (i.e., multithreading or multiprocessing architecture) in order to allow the ASM to iteratively evaluate service integrity over time. In order to allow the ASM to perform the intended purpose, the TS architecture has to provide full introspection to its internal components (e.g., by being fully open source or by providing specific APIs to do so). As such, the TS, provided that technological support is present, can also include automation scenarios with advanced hardware/software smart robot controllers.

With regard to the behavioral monitoring of smart devices, the TS is expected to make use of TCP-oriented protocols to communicate and exchange information. In fact, the behavioral monitoring component can also be used to detect slow DoS attacks that typically exploit connection-oriented characteristics (e.g., timeouts).

The slow DoS detection approach imposes some requirements on the TS environment; in particular, it must be possible to perform the following:

- Obtain raw network data captured either at the network or host level (i.e., in the form of PCAP packet capture files for offline analysis, or to access live raw PCAP data);
- Provide the possibility to generate legitimate traffic, as well as to perpetrate attacks on the network, for implementation and testing activities.

As the methodology for identifying cyberattacks that we aim to design involves user behavior analytics, using AI methodologies to detect user behavior suggests the following:

- It is possible to represent user actions as a series of sequential events. For example, these sequences can be mapped using characteristics extracted from deep/machine learning (D/ML) models by fitting them to the user's typical behavior since event sequence patterns present usual, accurate, and readable trends of the user behavior.
- It is possible to make use of suitable classification approaches to distinguish between normal and unusual behavior to identify possible attacks through DL or XAI methodologies.

The economics of cybersecurity effort/component aim to provide as much relevant and useful information as possible about risk assessment from an economic point of view that can help to make appropriate decisions and countermeasures. As such, there are no specific requirements imposed to the TS. Nevertheless, some ingredients are needed to perform such an evaluation. In particular, this includes the availability of suitable data concerning the frequency and the severity of attacks to assess the expected loss. Regarding the severity of attacks, the ideal condition should be to have data about financial losses. Nevertheless, the severity could also be measured from a technical point of view (e.g., the number of data breached in each attack) and then "converted" into financial losses based on some results and models from the existing literature. Other key information concerns the available budget to make investments and/or buy an insurance contract, and the cost of countermeasures needed to face cyber attacks.

Mitigation and reaction features such as Automatic Firewall Reconfiguration requires the knowledge of the following:

1. Runtime performance of each firewall;
2. Configuration of each firewall;
3. System topology, in order to know all cascaded firewalls to be loaded with (a part of) the traffic causing unacceptable overloading of a certain firewall.

The above list can be, respectively, obtained through the following means:

1. Some kind of monitoring, relying on the time needed by packets to cross firewalls. In this case, some additional hardware equipment or features are needed, e.g., packet time stamping. Another way can be through taking into account the (average) length of the firewall internal queue of messages waiting for processing in a suitable time window. In both cases, a detection alert shall be raised when some threshold is reached.
2. Firewall configurations are both downloaded and uploaded.
3. System topology is stored in a suitable repository, but this cannot be enough for systems whose configurations can change quickly. In this case, automatic techniques that are able to keep the system aligned and its description (model) should be available.

The three above activities shall be coordinated and managed by a software tool that, when triggered by alerts raised by (1), identifies the firewalls to be taken into account by using (3), and, starting from the current firewalls' configurations, computes the new ones and suitably updates the firewalls (2).

5. Proposed Approaches and Solutions

This section illustrates the practical application of the proposed architecture. Describing the overall implementation of the proposed architecture would not have allowed us to emphasize its peculiarities. For this reason, this section presents the implementation of the proposed architecture in three realistic (sub-)target systems: a Mobile Control Panel, an IoT framework, and an ROS testbed. They can be considered different pieces of the same puzzle, since they can live in the same TS.

In particular, taking as reference the architecture of Figure 1, variations in the considered target systems allow us to carry out the following:

- Explain the implementation of a specific architecture component and discuss in detail its features, challenges, and peculiarities.
- Showcase the adaptability and flexibility of the proposed architecture to the different application domains, HW/SW components, and systems.
- Showcase the integration of additional components and tools to leverage the overall features of the proposed architecture.

Each section introduces the specific target system and describes the implementation of one or more architecture components.

5.1. Instantiating the Monitoring Component

The following sections describe the monitoring component implementation considering an industrial manufacturing context where a Mobile Control Panel (MCP) regulating the movement of machinery must be monitored and managed to avoid cybersecurity vulnerabilities.

5.1.1. Mobile Control Panel

Usually, an MCP is used to manage and synchronize various machines' movements in the factory. For ensuring efficient coordination between different equipment, such as robotic arms, and assembly line components, they usually include some specific components such as the following (see the left–bottom square in Figure 3):

- *Control Interface:* It allows us to monitor and control the motion of machines. This interface includes hardware components such as touchscreens, buttons, and software for user-friendly usage and visualization.
- *PLC (Programmable Logic Controller):* It is the component that executes programmed instructions based on input signals and predefined logic.
- *Motion Controllers:* They translate commands from the PLC into precise movements for the connected machines; therefore, they are responsible for regulating speed, acceleration, and deceleration to ensure smooth operation.
- *Sensors:* Different sensors (such as proximity sensors, encoders, and vision systems) are used to monitor the system or enable (reaction) activity, such as to adjust machine movements. Sometimes, a *sensor gateway* can aggregate or transmit data from multiple sensors, acting as a bridge between sensors and the broader network infrastructure.
- *Communication Protocols:* Specific communication protocols can be used to exchange data with other machines or systems. Sometimes, a *Message Broker* can simplify communication or ensure reliable and scalable message delivery across heterogeneous systems.

Even if MCPs usually include mechanisms for anomalies or error detection and corrective action execution, they are susceptible to cybersecurity vulnerabilities, such as the following:

- *Lack of Authentication and Authorization:* Weaknesses in access control systems could provide unwanted users access to critical features. Attackers could be able to take over the machinery in the absence of adequate identification and authorization procedures, which might cause production interruptions or safety risks.
- *Insecure Communication Protocols*: Without proper encryption or authentication, the communication protocols or Message Broker can open the path to illegal command injection, data manipulation, and eavesdropping by network-acquired attackers.
- *Insufficient Firmware Security:* Firmware vulnerabilities or backdoors could be exploited to gain unauthorized access to the panel's functionalities and compromise the entire safety of the system.

- *Lack of Security Updates:* Regular security updates or patches are the most effective way to avoid system vulnerability and malware attacks.
- *Physical Access:* If not controlled, physical access can bypass security controls or directly manipulate the control panel to disrupt production or cause damage.
- *Inadequate Network Segmentation:* A breach in one part of the networks could potentially compromise the control panels and vice versa.
- *Supply Chain Risks:* Supply chain attacks can introduce malware, backdoors, or other security vulnerabilities into the manufacturing environment, compromising integrity and security.
- *Human Factors:* Insider threats or unintentional actions by employees, such as negligent handling of credentials, failure to follow security protocols, or falling victim to social engineering attacks, can also be exposed to cybersecurity risks.
- *Legacy Systems:* Outdated or unsupported software, making them more susceptible to exploitation due to the lack of security updates and patches.
- *Lack of Security Awareness:* Insufficient training and awareness regarding the best practices of cybersecurity can lead to inadvertent actions compromising the security of motion control panels.

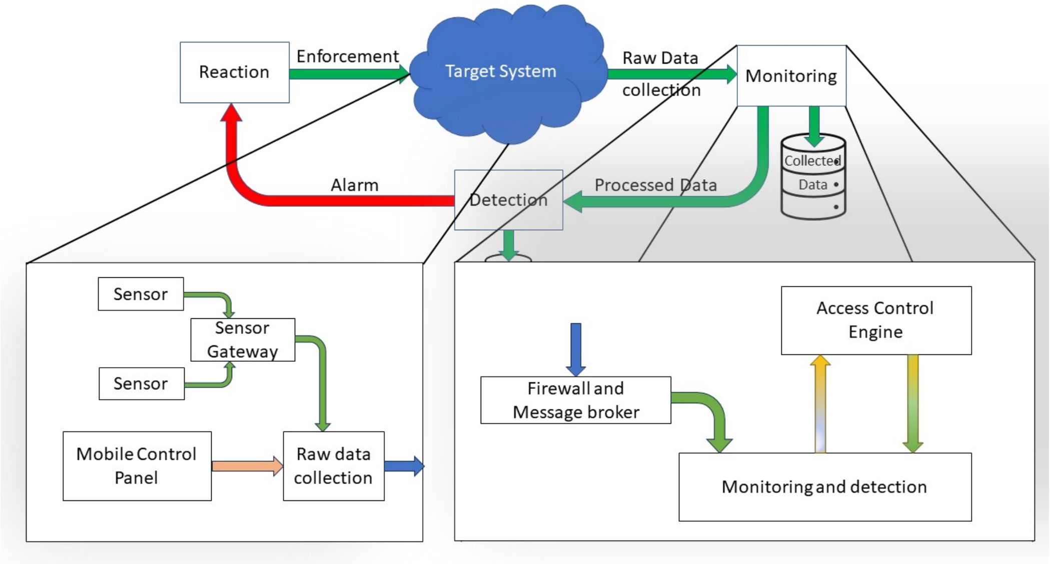

Figure 3. MCP abstract architectural view.

5.1.2. Monitoring Component

Figure 3 provides an instantiation of the monitoring component of Figure 1 considering the Mobile Control Panel as a (sub-)target system. In particular, due to the extensibility and adaptability of the proposed architecture, to address the previously mentioned vulnerability, three specific components are included and showcased in Figure 3:

- **Access Control Engine** manages the Mobile Control Panel access and enforces specific security policies defined by the Business Managers and Administrators. This component interacts with the monitoring component to send the proper authorization decision.
- **Monitoring** provides a dynamic and flexible solution to transparent access control network execution or a specific sensor. It works in synergy with the Reaction component for sending notifications to specific security or performance constraints in case of violations. Such constraints are not mandatorily specified at system startup. Still, they can be automatically set via the rule engines at runtime by injecting new rules on the event processors. The monitoring component is responsible for the motion control panel, the complex event processing, and for the Access Control Engine, actuators, and sensors. It is also able to manage the generation of access control request.

- **A Firewall and Message Broker:** This component collaborates with the monitoring by collecting performance and traffic figures to detect anomalies that can be mitigated by reconfiguring in real-time (parts of) the TS, e.g., active network devices in particular, like firewalls, as thoroughly addressed in Section Firewall Reconfiguration.

The instantiation also has the peculiarities of decoupling sensor management activities from resource usage and alarm administration. This eases the defining of new primitive events generated by (new/updated) sensors. Further, it renders the inference of events completely independent of the access and usage control rules. In addition, it allows for a quick update of the general resource access and usage regulations. Finally, by leveraging the specific sensor network, it leads to plane-specific corrective actions when resource violations occur and alarms are activated.

Delving into technical details, in the considered instantiation, the Message Broker runs on top of an Enterprise Service Bus and it handles all communications among the monitoring components. Different solutions, like SSL certificates (self-signed certificates), can be considered to improve communication security. The sensor nodes can monitor power consumption, detect indoor movements like Passive Infrared Sensors (PIR-based motion sensors), detect environmental noise, and measure temperature and relative humidity. To satisfy the access control security issues in the current instantiation, each Mobile Control Panel user has a Bluetooth Low Energy (BLE) periodically sending a message to identify and locate the user.

Considering the Access Control Engine represented in Figure 3, in the considered implementation (a subset), different control rules can be instantaneously activated or inhibited, but also (re)defined following runtime organization needs. This leaves the factory organization manager the freedom to select/define, at any specific time, the more appropriate control rules. Specific features are included for directly specifying the appropriate control rules or selecting the most suitable ones from a set of frequently adopted rules.

Considering the monitoring component presented in Figure 3, a dedicated complex event processor instance is responsible for managing the frequent updates of the rule set, with the need to check the correctness and compliance of the new rules. This includes the possibility of overriding the active rules set with a new one, with no impact on the overall factory management. In response to the challenges of CH1, CH2, and CH3 listed in Section 3, adopting automated infrastructure for defining and assessing control rules and managing monitoring activity represents an important improvement in the quality and performance of the industrial environment in case it allows for improving the control of security issues or simple violations.

For instance, considering CH3, the manager would be able to change the access control rules of a specific automatic environment. The manager would be able to perform this many times during any time period, affecting either the role of the access-requiring subject, the access time, or the available sensor values.

Considering the use of probes inside the components of the Mobile Control Panel in CH1 presented in Section 5.1.1, the monitoring component can collect internal execution data (white-box data) without knowing their source code structure (black-box data). Indeed, the implemented methodology makes the architecture and its components m "transparent" for functional and non-functional property assessments and prediction without revealing their internals. Data are collected through probes, preserving the principles of loose coupling and implementation neutrality.

The proposed instantiation allows the management of the following scenarios:

- Scenario 1: Each employee can use the Mobile Control Panel only during business hours (from 8 a.m. to 8 p.m.);
- Scenario 2: Only an authorized employee can update the Mobile Control Panel firmware during business hours;
- Scenario 3: The supervisor can access the Mobile Control Panel anytime.

5.2. Instantiating the Detection Component

5.2.1. Machine Learning-Based Detection

Machine Learning (ML) enables the efficient and rapid detection of complex trends and patterns, surpassing human capabilities, even when training models with vast datasets to identify security risks might not be suitable for unreliable, continuous, and real-time architectures. Our approach proposes to classify unusual patterns in the behavior of smart devices as potential cybersecurity threats within the IoT framework [63]. An explainable AI (XAI) strategy is designed to address these behavioral anomalies. Predictive association modeling is developed to enhance its effectiveness in identifying behavioral patterns and providing transparent explanations for them and their classifications. This approach stems from combining association-rule mining with classification techniques. The proposed methodology utilizes the representation of network users' (devices') activities through sequences of events that outline their behavior patterns.

5.2.2. ROS Testbed

An important ongoing research activity aims to better comprehend and analyze novel threats possibly affecting industrial environments. In a way, this activity can be seen as complementary to automatic network reconfiguration, i.e., reaction, because the goal here is to prevent what the reaction tries to mitigate. More specifically, the focus will be on the design of slow DoS threats affecting industrial protocols similar to those found in Internet of Things (IoT) environments and discussed in [4]. To this aim, a deep study of the context will be performed to assess the potential weaknesses of the communication protocols under test, leading to a potential attack exploitation.

The testbed depicted in Figure 4 was designed for this purpose. It represents a simplified (but still realistic) implementation of a target system, according to the definition given in Section 1. As shown, it consists of several fixed Robot Operating System (ROS) nodes (called *A*, *B*, and *C* in the figure) along with an ROS-based Autonomous Mobile Robot (AMR) (shown in the upper-right corner). Considering, in particular, slow DoS attacks, the work will be focused on implementing the considered scenario to analyze how a potential adversary may perpetrate malicious activities on the network.

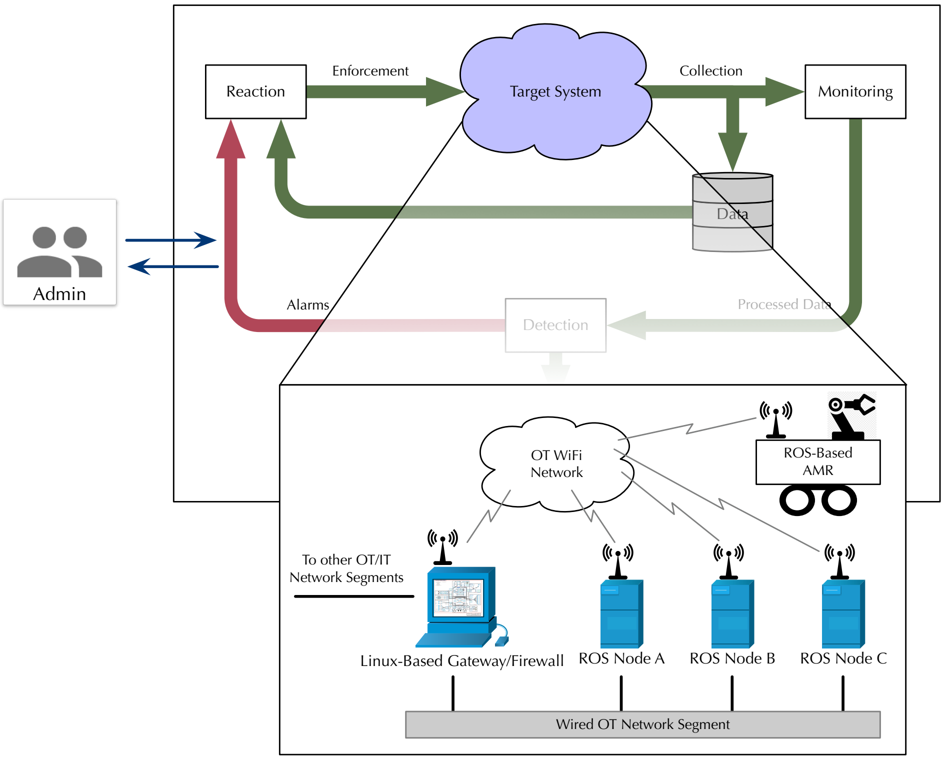

Figure 4. ROS testbed for slow DoS attack evaluation (updated with the addition of admin).

5.3. Instantiating the Reaction Component

Firewall Reconfiguration

As mentioned in Section 3, a variety of anomalous conditions may lead to an overload of some active network devices, firewalls in particular. An important, ongoing goal is to enable the system to automatically react to such a condition by transferring part of the load from the overloaded devices towards others without changing the overall security policy while doing so.

In the foreseen network protection scenario, the occurrence of an alarm for anomalous device load or link bandwidth consumption triggers a *reaction* phase, in which an automatic computation of a new security configuration of the affected devices take place. The updated configuration is then deployed in the subsequent *enforcement* phase.

All network functions are thus kept alive while the causes of the anomaly are investigated, leading to the selection and application of more specific countermeasures if needed. Once the offending traffic pattern ceases, either naturally or as a consequence of the countermeasures just mentioned, the system may return to its original configuration.

In contrast to the current state of the art outlined in Section 2, as well as the approach thoroughly discussed in [46], the practical relevance of this method heavily relies on the ability to deal with real-world device configuration. Being able to handle, for instance, IPTABLES firewall configuration as specified by the authors of [48] rather than abstract, simplified ones is indeed a challenging and important step.

Figure 5 is a specialization of the Open Platform Architecture of Figure 1, tailored to dynamic network reconfiguration. It shows how the Reaction Module calculates and enforces a new security configuration, taking alarms from the target system as input. As shown in the figure, the target system model, built from a suitable data model, helps to identify the optimal set of devices to be affected by load redistribution. The current security configuration, to be transformed and enforced by the Reaction Module, is also stored in a data model.

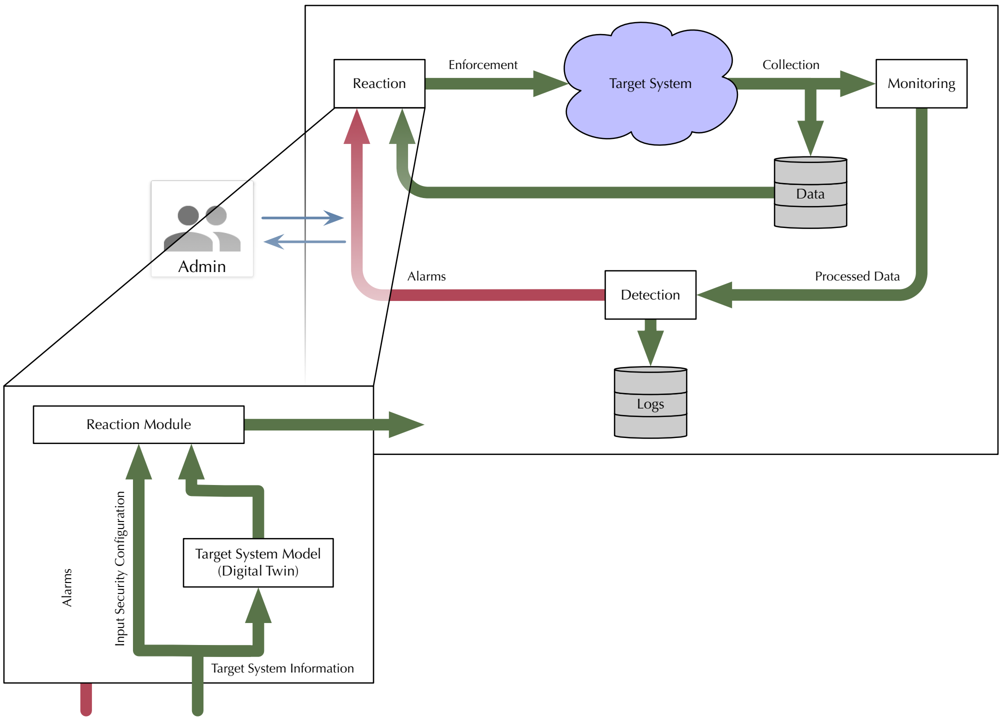

Figure 5. Automatic reconfiguration of network devices (updated with the addition of admin).

5.4. Financial Perspective

As stated in Section 1, the OPA enables the administrator to define that certain actions, such as choosing between alternative remedies or carrying out significant reconfigurations of the TS, may require human oversight and authorization. This must include the financial perspective, that is the economic impact assessment of the chosen countermeasures. Keeping in mind the challenges highlighted in Section 3, a key point is to quantify both the monetary cost of investing in countermeasures mapped to security level and the benefits stemming from these investments. Moreover, in order to compare different choices, suitable security metrics are needed.

In this context, a dynamic approach is required which involves continuously assessing risks, planning targeted investments, and adapting security measures based on real-time feedback. The process includes evaluating vulnerabilities, making cost-effective security investments, and using continuous monitoring to assess effectiveness, also by implementing the security metrics proposed in the reference literature. Feedback is then used to update the system's risk profile and guide future investments, ensuring an iterative optimization of resources and resilience. The described procedure cannot, of course, disregard the budget allocated for security investments. Finally, the opportunity to transfer part of the risk by purchasing an insurance policy to cover cyber risk can be evaluated.

6. Conclusions

The advanced skills demonstrated by CyberCrime today present significant challenges to the technical and scientific communities that often have to navigate conflicting requirements such as resources, performance, and costs. This document outlines the Open Platform Architecture (OPA), illustrating how cutting-edge security measures and countermeasures can be integrated into various target systems. This is achieved by meticulously designing both the target system and the security mechanisms to address a broad spectrum of potential attacks. As a result, the secured target system can swiftly and automatically detect and respond to hazardous, undesirable, or unexpected situations.

The secure design and/or upgrade of cyber-physical systems and critical infrastructures can be a very large task that cannot be carried out by simply adding secure mechanisms and controls due to the requirements stemming from interdependencies, resources constrained to some extent, performance, and costs, whose ultimate goal is to build a secure target system that is able to swiftly and automatically detect and respond to hazardous, undesirable, or unexpected situations. Then, such a task shall be carefully driven by a suitable methodology, allowing us to take into account all constraints, and the design/upgrade of each system component at the same time. Such a methodology has been presented here through the OPA, where security challenges are addressed together with a careful design of interfaces among a set of structured subsystems and components. Our paper presented here a methodology through the OPA, where security challenges are addressed together with suitable interfaces among a set of structured subsystems and components. The proposed methodology allows for taking into account heterogeneous constraints by considering the design of each system component. A multifaceted use case has been outlined here to show the potential target systems considered by the project, focusing on monitoring, detection, and reaction activities. Further work will be devoted to implementing and evaluating a full deployment of a use case integrating and extending what has been presented here. Further steps in this work will be performed towards a full deployment of a use case integrating the ones presented here.

Author Contributions: A.C. contributed to the Mobile Control Panel and ontology-based monitoring topics mentioned in this paper. E.C. and I.C.B. collected and combined contributions on the Approaches and Solutions section of the manuscript and contributed to the slow denial service topic mentioned in the document. M.C. contributed to the work by revising it. L.D. coordinated the work, collected and combined contributions, prepared the manuscript's Abstract, Introduction, and Conclusion sections, and contributed to the Firewall Reconfiguration sections of the document. A.F., G.M. and G.P. contributed to the privacy, security, data analysis, and interpretation aspects mentioned in the paper. F.L. coordinated the work, collected and combined contributions on the Reference Architecture section of the manuscript, contributed on Advanced Secure Monitor aspects, and helped to revise the paper. E.M. collected and combined contributions on the Approaches and

Solutions section of the manuscript and contributed on the Mobile Control Panel and ontology-based monitoring topics mentioned in the paper. A.O. collected and combined contributions on the state-of-the-art section and contributed on the economic and financial aspects of cybersecurity mentioned in the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union—NextGenerationEU.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Langner, R. Stuxnet: Dissecting a Cyberwarfare Weapon. *IEEE Secur. Priv.* **2011**, *9*, 49–51. [CrossRef]
2. Chen, T.M. Stuxnet, the real start of cyber warfare? [Editor's Note]. *IEEE Netw.* **2010**, *24*, 2–3. [CrossRef]
3. Pratama, I.P.A.E. Tcp syn flood (dos) attack prevention using spi method on csf: A poc. *Bull. Comput. Sci. Electr. Eng.* **2020**, *1*, 63–72. [CrossRef]
4. Cambiaso, E.; Papaleo, G.; Chiola, G.; Aiello, M. Slow DoS attacks: Definition and categorisation. *Int. J. Trust. Manag. Comput. Commun.* **2013**, *1*, 300–319. [CrossRef]
5. Vaccari, I.; Aiello, M.; Cambiaso, E. Slowtt: A slow denial of service against iot networks. *Information* **2020**, *11*, 452. [CrossRef]
6. Papaleo, G.; Cambiaso, E.; Farina, P.; Aiello, M. Perpetrate network attacks from mobile devices. In Proceedings of the 2015 Seventh International Conference on Ubiquitous and Future Networks, Sapporo, Japan, 7–10 July 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 597–602.
7. Henzinger, T.A.; Karimi, M.; Kueffner, K.; Mallik, K. Runtime Monitoring of Dynamic Fairness Properties. In Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, FAccT 2023, Chicago, IL, USA, 12–15 June 2023; ACM: New York, NY, USA, 2023; pp. 604–614. [CrossRef]
8. Vierhauser, M.; Egyed, A. Runtime Monitoring for Systems of System. In *Digital Transformation: Core Technologies and Emerging Topics from a Computer Science Perspective*; Springer: Berlin/Heidelberg, Germany, 2023; p. 203.
9. Barsocchi, P.; Calabrò, A.; Ferro, E.; Gennaro, C.; Marchetti, E.; Vairo, C. Boosting a Low-Cost Smart Home Environment with Usage and Access Control Rules. *Sensors* **2018**, *18*, 1886. [CrossRef]
10. de Freitas Bezerra, D.; de Medeiros, V.W.C.; Gonçalves, G.E. Towards a control-as-a-service architecture for smart environments. *Simul. Model. Pract. Theory* **2021**, *107*, 102194. [CrossRef]
11. Aceto, L.; Achilleos, A.; Attard, D.P.; Exibard, L.; Francalanza, A.; Ingólfsdóttir, A. A Monitoring Tool for Linear-Time μ HML. In Proceedings of the Coordination Models and Languages: 24th IFIP WG 6.1 International Conference, COORDINATION 2022, Held as Part of the 17th International Federated Conference on Distributed Computing Techniques, DisCoTec 2022, Lucca, Italy, 13–17 June 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 200–219.
12. Attard, D.P.; Aceto, L.; Achilleos, A.; Francalanza, A.; Ingólfsdóttir, A.; Lehtinen, K. Better late than never or: Verifying asynchronous components at runtime. In Proceedings of the Formal Techniques for Distributed Objects, Components, and Systems: 41st IFIP WG 6.1 International Conference, FORTE 2021, Held as Part of the 16th International Federated Conference on Distributed Computing Techniques, DisCoTec 2021, Valletta, Malta, 14–18 June 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 207–225.
13. Ackermann, C.; Lindvall, M.; Cleaveland, R. Towards Behavioral Reflexion Models. In Proceedings of the ISSRE 2009, 20th International Symposium on Software Reliability Engineering, Mysuru, Karnataka, India, 16–19 November 2009; IEEE Computer Society: Piscataway, NJ, USA, 2009; pp. 175–184. [CrossRef]
14. Wendehals, L.; Orso, A. Recognizing Behavioral Patterns Atruntime Using Finite Automata. In Proceedings of the 2006 International Workshop on Dynamic Systems Analysis, WODA '06, Shanghai, China, 23 May 2006; pp. 33–40. [CrossRef]
15. Leenen, L.; Meyer, T.A. Artificial Intelligence and Big Data Analytics in Support of Cyber Defense. In *Developments in Information Security and Cybernetic Wars*; IGI Global: Hershey, PA, USA, 2019.
16. Mothukuri, V.; Khare, P.; Parizi, R.M.; Pouriyeh, S.; Dehghantanha, A.; Srivastava, G. Federated learning-based anomaly detection for IoT security attacks. *IEEE Internet Things J.* **2021**, *9*, 2545–2554. [CrossRef]
17. Hussain, F.; Hussain, R.; Hassan, S.A.; Hossain, E. Machine Learning in IoT Security: Current Solutions and Future Challenges. *IEEE Commun. Surveys Tuts.* **2020**, *22*, 1686–1721. [CrossRef]
18. Samek, W.; Montavon, G.; Lapuschkin, S.; Anders, C.J.; Müller, K.R. Explaining deep neural networks and beyond: A review of methods and applications. *Proc. IEEE* **2021**, *109*, 247–278. [CrossRef]
19. Carvalho, D.V.; Pereira, E.M.; Cardoso, J.S. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics* **2019**, *8*, 832. [CrossRef]
20. Doshi-Velez, F.; Kim, B. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv* **2017**. [CrossRef]
21. Gabbrielli, M.; Martini, S. Abstract Machines. In *Programming Languages: Principles and Paradigms*; Undergraduate Topics in Computer Science ((UTICS)); Springer International Publishing: Cham, Switzerland, 2023; Chapter 1, pp. 1–24. [CrossRef]

22. Di Pietro, R.; Lombardi, F. *Security for Cloud Computing*; Artec House: Boston, MA, USA, 2015; ISBN 978-1-60807-989-6.
23. Baiardi, F.; Sgandurra, D. Building Trustworthy Intrusion Detection through VM Introspection. In Proceedings of the Third International Symposium on Information Assurance and Security, Manchester, UK, 29–31 August 2007; pp. 209–214. [CrossRef]
24. Amit, N.; Wei, M. The Design and Implementation of Hyperupcalls. In Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC 18), Boston, MA, USA, 11–13 July 2018; pp. 97–112.
25. Sentanoe, S.; Dangl, T.; Reiser, H.P. KVMIveggur: Flexible, secure, and efficient support for self-service virtual machine introspection. Proceedings of the Twenty-Second Annual DFRWS USA. *Forensic Sci. Int. Digit. Investig.* **2022**, *42*, 301397. [CrossRef]
26. Partridge, C.; Mitchell, A.; Cook, A.; Sullivan, J.; West, M. *A Survey of Top-Level Ontologies—To Inform the Ontological Choices for a Foundation Data Model*; CDBB: Cambridge, UK, 2020. [CrossRef]
27. Lynch, K.; Ramsey, R.; Ball, G.; Schmit, M.; Collins, K. Conceptual design acceleration for cyber-physical systems. In Proceedings of the 2017 Annual IEEE International Systems Conference (SysCon), Montreal, QC, Canada, 24–27 April 2017; pp. 1–6. [CrossRef]
28. McCallam, D.H. *An Analysis of Cyber Reference Architectures*; Technical Report STO-EN-IST-170, NATO Science and Technology Organization. 2019. Available online: https://www.sto.nato.int/publications/STO%20Educational%20Notes/STO-EN-IST-170/EN-IST-170-09.pdf (accessed on 8 November 2024).
29. DoD. Department of Defense Cybersecurity Reference Architecture. 2023. Available online: https://dodcio.defense.gov/Portals/0/Documents/Library/CS-Ref-Architecture.pdf (accessed on 8 November 2024).
30. Mpekoa, N. An Analysis of Cybersecurity Architectures. *Int. Conf. Cyber Warf. Secur.* **2024**, *19*, 200–207. [CrossRef]
31. Pleshakova, E.; Osipov, A.; Gataullin, S.; Gataullin, T.; Vasilakos, A. Next gen cybersecurity paradigm towards artificial general intelligence: Russian market challenges and future global technological trends. *J. Comput. Virol. Hacking Tech.* **2024**, *20*, 429–440. [CrossRef]
32. Dacorogna, M.; Marie, K. Managing cyber risk, a science in the making. *Scand. Actuar. J.* **2023**, *10*, 1000–1021. [CrossRef]
33. Böhme, R.; Thomas, N. *Dependability Metrics: Advanced Lectures*; Chapter Economic Security Metrics; Springer: Berlin/Heidelberg, Germany, 2008; pp. 176–187.
34. Orlando, A. Cyber Risk Quantification: Investigating the Role of Cyber Value at Risk. *Risks* **2021**, *9*, 184. [CrossRef]
35. Thareem, Y.; Azka, A.; Haider, A.; Muhammed, F.A.; Narmeen, S. Framework for calculating return on security investment (ROSI) for security-oriented organizations. *Future Gener. Comput. Syst.* **2019**, *95*, 754–763. [CrossRef]
36. Marotta, A.; Martinelli, F.; Nanni, S.; Orlando, A.; Yautsiukhin, A. Cyber-insurance survey. *Comput. Sci. Rev.* **2017**, *24*, 35–61. [CrossRef]
37. Tsohou, A.; Diamantopoilou, V.; Gritzalis, S.; Lambrinoudakis, C. Cyber insurance: State of the art, trends and future directions. *Int. J. Inf. Secur.* **2023**, *22*, 737–748. [CrossRef]
38. Scarfone, K.; Hofman, P. *Guidelines on Firewalls and Firewall Policy*; NIST SP 800-41 Rev. 1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2009. [CrossRef]
39. Mohan, R.; Yazidi, A.; Feng, B.; Oommen, B.J. On optimizing firewall performance in dynamic networks by invoking a novel swapping window—Based paradigm. *Int. J. Commun. Syst.* **2018**, *31*, e3773. [CrossRef]
40. Harada, T.; Tanaka, K.; Mikawa, K. A Heuristic Algorithm for Relaxed Optimal Rule Ordering Problem. In Proceedings of the 2nd Cyber Security in Networking Conference (CSNet), Paris, France, 24–26 October 2018; pp. 1–8. [CrossRef]
41. Jabal, A.A.; Davari, M.; Bertino, E.; Makaya, C.; Calo, S.; Verma, D.; Russo, A.; Williams, C. Methods and Tools for Policy Analysis. *ACM Comput. Surv.* **2019**, *51*, 1–35. [CrossRef]
42. Bodei, C.; Ceragioli, L.; Degano, P.; Focardi, R.; Galletta, L.; Luccio, F.; Tempesta, M.; Veronese, L. FWS: Analyzing, maintaining and transcompiling firewalls. *J. Comp. Sec.* **2021**, *29*, 77–134. [CrossRef]
43. Daly, J.; Liu, A.X.; Torng, E. A Difference Resolution Approach to Compressing Access Control Lists. *IEEE/ACM Trans. Netw.* **2016**, *24*, 610–623. [CrossRef]
44. Hadjadj, T.E.; Tebourbi, R.; Bouhoula, A.; Ksantini, R. Optimization of Parallel Firewalls Filtering Rules. In Proceedings of the International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 19–21 September 2019; pp. 1–6. [CrossRef]
45. Bagheri, S.; Shameli-Sendi, A. Dynamic Firewall Decomposition and Composition in the Cloud. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3526–3539. [CrossRef]
46. Durante, L.; Seno, L.; Valenzano, A. A Formal Model and Technique to Redistribute the Packet Filtering Load in Multiple Firewall Networks. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 2637–2651. [CrossRef]
47. Al-Shaer, E.; Hamed, H.; Boutaba, R.; Hasan, M. Conflict Classification and Analysis of Distributed Firewall Policies. *IEEE J. Sel. Areas Commun.* **2005**, *23*, 2069–2084. [CrossRef]
48. The Netfilter Core Team. The netfilter.org "iptables" Project. 1999–2021. Available online: https://www.netfilter.org/projects/iptables/index.html (accessed on 8 November 2024).
49. Biener, C.; Eling, M.; Wirfs, J.H. Insurability of cyber risk: An empirical analysis. *Geneva Pap. Risk Insur.—Issues Pract.* **2015**, *40*, 131–158. [CrossRef]
50. OECD. *Enhancing the Availability of Data for Cyber Insurance Underwriting, The Role of Public Policy and Regulation*; OECD: Paris, France, 2020. Available online: https://web-archive.oecd.org/2020-08-18/546625-Enhancing-the-Availability-of-Data-for-Cyber-Insurance-Underwriting.pdf (accessed on 8 November 2024).

51. OECD. *Types of Cyber Incidents and Losses*; OECD: Paris, France, 2017. [CrossRef]
52. Böhme, R. Security Metrics and Security Investment Models. In Proceedings of the Advances in Information and Computer Security. IWSEC 2010. Lecture Notes in Computer Science, Kobe, Japan, 22–24 November 2010; Springer: Berlin, Germany, 2010.
53. Skeoch, H. Expanding the Gordon-Loeb model to cyber-insurance. *Comput. Secur.* **2022**, *112*, 102533. [CrossRef]
54. Sung, M.; Olivier, P.; Lankes, S.; Ravindran, B. Intra-unikernel isolation with Intel memory protection keys. In Proceedings of the 16th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, VEE '20, Lausanne, Switzerland, 17 March 2020; pp. 143–156. [CrossRef]
55. Reed, A.; Dooley, L.S.; Mostefaoui, S.K. A reliable real-time slow DoS detection framework for resource-constrained IoT networks. In Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 7–11 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
56. Gaggero, M.; Di Paola, D.; Petitti, A.; Caviglione, L. When time matters: Predictive mission planning in cyber-physical scenarios. *IEEE Access* **2019**, *7*, 11246–11257. [CrossRef]
57. Aiello, M.; Papaleo, G.; Cambiaso, E. SlowReq: A weapon for cyberwarfare operations. Characteristics, limits, performance, remediations. In Proceedings of the International Joint Conference SOCO'13-CISIS'13-ICEUTE'13, Salamanca, Spain, 11–13 September 2013; Springer: Berlin/Heidelberg, Germany, 2014; pp. 537–546.
58. Yu, S.; Zhou, W.; Jia, W.; Guo, S.; Xiang, Y.; Tang, F. Discriminating DDoS attacks from flash crowds using flow correlation coefficient. *IEEE Trans. Parallel Distrib. Syst.* **2011**, *23*, 1073–1080. [CrossRef]
59. Sikora, M.; Gerlich, T.; Malina, L. On detection and mitigation of slow rate denial of service attacks. In Proceedings of the 2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Dublin, Ireland, 28–30 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.
60. Cambiaso, E.; Papaleo, G.; Aiello, M. Implementation of SlowDroid: Slow DoS Attack Performed by a Smartphone. *Int. J. Comput. Digit. Syst.* **2015**, *4*, 165–173. [CrossRef] [PubMed]
61. Hamed, H.; Al-Shaer, E. Dynamic Rule-Ordering Optimization for High-Speed Firewall Filtering. In Proceedings of the ACM Symp. on Information, Computer and Communications Security (ASIACCS), Taipei, Taiwan, 21–24 March 2006; pp. 332–342. [CrossRef]
62. Wikipedia. Netfilter—Wikipedia, The Free Encyclopedia. 2024. Available online: http://en.wikipedia.org/w/index.php?title=Netfilter&oldid=1232791514 (accessed on 22 October 2024).
63. Costa, G.; Forestiero, A.; Ortale, R. Rule-Based Detection of Anomalous Patterns in Device Behavior for Explainable IoT Security. *IEEE Trans. Serv. Comput.* **2023**, *16*, 4514–4525. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.