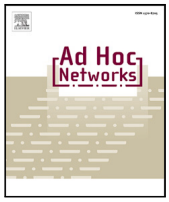
Ad Hoc Networks 156 (2024) 103414

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc



Flexible zero trust architecture for the cybersecurity of industrial IoT infrastructures☆

Claudio Zanasi *, Silvio Russo, Michele Colajanni

Department of Computer Science and Engineering, University of Bologna, Italy

ARTICLE INFO

Keywords:
Zero trust
Micro-segmentation
SDN
Industrial IoT

ABSTRACT

The growing digitalization of industrial systems and the increasing adoption of cloud technologies pose significant challenges to the secure management of modern industrial infrastructures integrating different Industrial Internet of Things (IIoT). Existing cybersecurity solutions can manage uniform and centralized software systems but are not designed to accommodate the requirements of heterogeneous IIoT devices, such as hard real-time operations, high reliability, and decentralization for distributed decision-making.

We present a novel security architecture that is specifically designed to address the stringent requirements of IIoT systems. It is based on a network micro-segmentation that can be seamlessly integrated into existing environments, and two main components: a software-defined network (SDN) ensuring a unified abstraction layer for policy enforcement across diverse environments; and a centralized security management layer that simplifies the policy execution of any architectural design. We demonstrate the feasibility and effects of this original combination through a prototype. It experimentally demonstrates that our peer-to-peer SDN coupled with an asynchronous policy distribution process guarantees resiliency to individual failures, and enables fully decentralized operations while still ensuring a central flexible management of network topology and security policies.

1. Introduction

The rapid proliferation of cyber–physical systems along with the integration of cloud computing with industrial plants, work-from-home environments, and Bring-Your-Own-Device solutions have transformed the industrial landscape [1,2]. The conventional reliance on robust and secure solutions like Virtual Private Networks (VPNs) to interconnect different regions at scale is facing significant challenges [3]. The adoption of cloud computing and the rise of the Internet of Things (IoT) with several types of devices that can be connected remotely have made irrelevant any network perimeter principle. The implicit trust in internally connected devices is becoming obsolete as well. The rise of sophisticated cyber attacks imposes on organizations the adoption of a proactive and dynamic approach to cybersecurity. In this context, the Zero Trust security model is emerging [4] as a paradigm shift for cybersecurity that is adopted by the most mature actors from companies to nations [5–7]. In this model, different elements of the architecture do not implicitly trust each others, but there is a continuous identification and authorization of all the interacting components [8]. This new security model can be applied to any environment and is becoming the preferred approach for protecting critical infrastructure and industrial control systems (ICS) [9]. When it comes to critical infrastructure, such as power grids, transportation systems, and water treatment plants, the cyber risks are severe. Hence, embracing Zero Trust seems imperative for any highly threatened organization, but there are multiple strategies for implementing this security model [5]. We think that micro-segmentation is the most promising technique for industrial environments and systems characterized by heterogeneous IoT devices. This approach allows granular control and isolation of network segments, thus creating additional layers of defense against threats. It enhances network visibility by imposing stringent communication rules among workloads. Furthermore, it facilitates breach containment by enabling efficient traffic monitoring and alignment with security policies. Moreover, it adds a layer of complexity for attackers as they must navigate through multiple micro-segments to reach their target [10]. Despite the multiple benefits, micro-segmentation remains a theoretically valid model that requires a lot of effort to be implemented and maintained.

In large and heterogeneous networks, such as those found in modern industries, the sheer volume of required policies can be overwhelming.

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

* Corresponding author.
E-mail addresses: claudio.zanasi4@unibo.it (C. Zanasi), silvio.russo3@unibo.it (S. Russo), michele.colajanni@unibo.it (M. Colajanni).

https://doi.org/10.1016/j.adhoc.2024.103414
Received 30 May 2023; Received in revised form 4 January 2024; Accepted 18 January 2024
Available online 8 February 2024
1570-8705/© 2024 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Moreover, these policies must be continuously updated to align with the evolving network and business requirements. Managing multiple dynamic policies can be time-consuming and error-prone, potentially introducing novel vulnerabilities. To address these issues, we propose an integration of micro-segmentation with Software Defined Networking (SDN). This integration enables us to operate at the network layer and dynamically modify the network layout based on various factors, including policies, system conditions, and user behavior. Through SDN solutions, we can efficiently propagate and communicate changes in the topology to the entire network.

SDN introduces other challenges that need to be overcome for a truly successful adoption. For example, *node reachability* is crucial when establishing an overlay network, and it must consider the characteristics of the underlying layer that is typically an IP network. In real-world scenarios, achieving end-to-end reachability is not always guaranteed. Network Address Translation (NAT) and firewalls can delay global access to clients located behind them. In such situations, deploying an overlay network requires careful consideration and mitigation strategies to ensure reliable connectivity. *Management and Administration* of an overlay network, especially its control plane, can be complex. Timely detection and resolution of failures, undesired behavior, and performance issues pose significant challenges. Provisioning additional configurations or client certificates in a large, distributed environment can be cumbersome. Our proposal leveraging an SDN approach enables remote and dynamic management and monitoring, mitigating some of these difficulties. *Overhead* can represent another issue because the efficiency of an overlay network in packet processing and forwarding does not match that of the underlying network routers. Moreover, the network topology in an overlay environment may not always be clear and fixed, making route optimization an ongoing concern. We recognize the need to address these overhead issues and strive to optimize the performance of the overlay network while maintaining its flexibility and adaptability.

Let us outline the main contributions of this paper.

1. We propose a novel design for a Zero Trust Architecture that is optimized for heterogeneous industrial environments and that can be seamlessly extended to integrate cloud technologies and IoT solutions.
2. We define a network management strategy that can address the main challenges associated with the micro-segmentation security approach, especially in terms of efficient policy management.
3. We validate the proposed solution through a prototype that emulates operations in a realistic environment.

The rest of the paper is organized as follows. Section 2 discusses related works. Section 3 describes the proposed architecture. Section 4 presents the main details of the prototype. Section 5 analyzes the experimental results. Section 6 discusses the trade-offs of the presented architecture. Section 7 summarizes the main conclusions and future work.

2. Related work

We consider micro-segmentation as the most appropriate implementation strategy for a modern Zero Trust approach [10,11]. By dividing the network into smaller logical segments, micro-segmentation emerges as a pivotal solution in the dynamic domains of the heterogeneous IoTs [12,13] and industrial environments [9]. Previous studies propose a theoretical analysis of the efficacy of micro-segmentation without addressing the real challenges that are related to its design, implementation, and operation. For example, the paper by Li et al. [14] outlines the key challenges of the industrial sector in terms of scalability and security. Network management of 5G-IoT deployments is also identified as a major issue that must be addressed. Configuring and maintaining security policies for a vast and rapidly changing IoT system encompassing thousands of computers and devices can be a daunting task. Our proposal offers an original solution to this problem by integrating micro-services with a flexible software-defined network and a central management system operating at a high-level understandable abstraction.

The authors in [15] propose an interesting approach for implementing zero-trust security in micro-service architectures. However, this proposal relies on assumptions that may not hold true in reality. For instance, it assumes the trustworthiness of the infrastructure provider and that the provider's infrastructure is free of vulnerabilities. These assumptions undermine the foundations of the Zero Trust model. Moreover, the adopted fine-grained access control policies are unsuitable for large-scale deployments especially when the micro-service environment is subject to frequent changes. Our proposal addresses all these limitations by developing a robust and scalable solution that ensures security and scalability in large-scale deployments and facilitates management of the demanding requirements of IIoT environments.

Many modern organizations are challenged by the necessity of configuring and maintaining micro-segmentation across multiple cloud platforms, and adapting consistent security policies with the dynamic nature of cloud infrastructures. If we connect industrial environments with the cloud, then the implementation of a Zero Trust solution is even more challenging [16,17], because the trust concept becomes complex (e.g., [18,19]), and effective management of micro-segmentation is even more crucial. The software-defined network (SDN) adopted by our architectural model can abstract the underlying cloud infrastructure and make viable policy management even for a multi-cloud environment.

The paper [20] explores the role of emerging technology such as SDN in supporting a Zero Trust security strategy in cloud computing environments. The authors discuss the challenges associated with implementing a Zero Trust strategy in cloud computing, including strong authentication and access control policies, securing network traffic between cloud services, and managing multiple cloud providers and platforms. It highlights the effectiveness of the Zero Trust approach in securing data and applications stored and accessed remotely. Micro-segmentation is emphasized as a key component for implementing Zero Trust effectively. However, the authors offer only a theoretical analysis of the proposed security architecture. The high level solution to the identified challenges does not address the practical issues and difficulties that arise during real-world deployment.

This paper is specifically oriented to address the main issues that are raised by the authors in [10]. They describe the benefits of implementing micro-segmentation through SDN but also the challenges associated with this approach, such as substantial network modifications and single-point-of-failure risks of a central controller. Our solution mitigates the need for extensive network changes while maintaining the benefits of micro-segmentation and reducing the risk of single-point-of-failure. The discussions about SDN have go around the crucial decision between centralized and distributed controllers [21,22]. Some papers [23,24] highlight the ability of SDN to provide dynamic network management and control through the separation of the control plane from the data plane, and delve into the integration of SDN with Zero Trust security principles. Although the proposed solutions conjecture the effectiveness of SDN also in IoT contexts, they rely on an SDN approach based on a centralized architecture. This scheme favors network management but introduces significant limits to the scalability and resilience of the infrastructure. Indeed, network disruptions and/or an overwhelming number of requests could jeopardize the single central element responsible for managing the entire network. Furthermore, these central controllers represent a single point of vulnerability for the entire network. To meet the demands of today's computing landscape, where scalability and resilience are of paramount importance, it is imperative that the approach can swiftly adapt to changes in the surrounding environment.

As the complexity of modern infrastructures continues to increase, SDN seems a promising solution for their management. A prevalent approach involves the deployment of solutions based on the OpenFlow protocol, which enables precise control over information flow within the network. However, the centralization of responsibilities in the SDN controller introduces significant security and reliable concerns because it creates a single point of failure and renders the controllers a highly attractive target for potential attackers. A compromised SDN controller can cause a denial of service attack or even expose the entire network to possible vulnerabilities [25,26].

The implementation of network-wide encryption in an OpenFlow-based SDN infrastructure continues to rely on optional features and manual configurations of various components. This process introduces a vulnerability factor stemming from the possibility of human errors. In heterogeneous network environments, where diverse devices and technologies coexist, the complexity of this task is notably increased. Ensuring a consistent and error-free deployment of network-wide encryption remains an open challenge [27].

Unlike existing proposals, our solution adopts a centrally managed overlay SDN with a peer-to-peer communication model and leverages WireGuard [28] to provide encryption and mutual authentication by default. While this peer-to-peer architecture sacrifices precise control over packet flow within the network, it enhances overall network resilience by eliminating the single point of failure represented by the SDN controllers. Through certificate-based mutual authentication, security policies can be directly enforced by individual resources in a decentralized way. Decentralization also mitigates the risk of a complete network takeover in the event of a compromise of a single node.

Bringhenti et al. [29] have proposed a framework for IoT networks that is designed to enforce precise security policies within an SDN. Their approach begins with high-level policy definitions and employs a Satisfiability Modulo Theories (SMT) system to automatically derive an optimal configuration for the SDN controllers. This configuration aims to implement all security policies while simultaneously maximizing network performance. This innovative approach establishes a direct link between network performance and its security configuration because any policy change can trigger a network reconfiguration with unpredictable effects on performance and resiliency. While our proposal aligns with the concept of generating low-level security configurations for network resources from a central policy repository to reduce management complexity, our approach diverges in terms of policy enforcement. Instead of centralizing security policy enforcement at the SDN controller level, our solution delegates this responsibility to individual network resources. This approach results in a complete decoupling of regular network operations from the security infrastructure. Consequently, changes in policy do not impact the performance and resilience profile of the overall network. Moreover, it becomes possible to implement network optimization actions without the risk of compromising its security. Other papers that have explored the idea of distributed controllers (e.g., [30–32]) enhance resilience and scalability, but they complicate overall management and configuration policies, and require careful attention to coordination and configuration synchronization. We propose a hybrid approach to micro-segmentation that combines the advantages of scalability and robustness from distributed controllers, while still maintaining the simplicity benefits of a centralized network configuration.

3. Proposed architecture

We propose an innovative solution for implementing a Zero Trust Architecture (ZTA) that can address the challenges of Industrial Internet of Things (IIoT) systems. In these contexts, the majority of interactions occur between individual computing resources without human supervision. Unlike existing Zero Trust security solutions that primarily focus on regulating user access to protected resources based on user identities, our approach recognizes the central role of computational resources in IIoT systems. Securing the interactions between these resources is crucial for maintaining a robust and trustworthy environment.

By adopting a micro-segmentation approach, we break down the network into small logically decoupled segments, enabling precise control and isolation of interactions between computing resources. Each segment becomes an isolated trust zone, and access privileges are defined based on the specific requirements and trust levels associated with the resources within that segment. This approach ensures that only authorized resources can communicate with each other, effectively reducing the attack surface and minimizing the risk of lateral movement within the network in the event of a security breach.

To design a security architecture suitable for IIoT environments, we have identified the following guiding principles.

- **Resource centric**: the advent of wireless networks, such as 5G, has revolutionized connectivity and rendered the traditional concept of a security perimeter obsolete. In this new paradigm, where resources are interconnected and communication occurs beyond physical boundaries, a shift towards a resource-centric approach becomes essential for ensuring robust security. In a resource-centric network, where the focus is on securing individual resources rather than the network perimeter, authentication mechanisms play a pivotal role. By implementing an authentication mechanism between resources, the network can verify the identities and authorization of each resource involved in the communication process.
- **Decentralization**: automated industrial systems manage all sorts of critical infrastructure where any failure can have severe consequences, including safety implications for both people and equipment. In these environments, it is of utmost importance to avoid single points of failure that can lead to cascading failures, even when the majority of the resources are not directly affected. The primary objective is to ensure the uninterrupted operation of the system and maintain its continuity.
- **Reduced overhead**: in industrial systems, where hard real-time operational constraints are prevalent, maintaining optimal performance is paramount when implementing a security infrastructure. This requirement becomes especially critical in the context of a micro-segmentation approach, where the number of security rules to enforce increases with the granularity of the network segments.
- **Backward compatibility**: in industrial systems, the rate of change is typically lower compared to pure software systems due to the longer expected lifetime of the equipment and the presence of legacy technologies. As a result, the security system implemented in these environments must be designed to minimize disruptive or breaking changes in order to facilitate its integration.

Given these design constraints, our approach is based on a modified logical model of the ZTA defined by the National Institute of Standards and Technology (NIST) [5]. A comparison between our approach and the NIST ZTA model is illustrated in Fig. 1.

The key distinction lies in the architecture governing the Policy Decision Point (PDP) and the Policy Enforcement Point (PEP). Our approach implements a distributed architecture by decentralizing the responsibilities for decision-making and policy enforcement across individual resources within the network. A central management system assumes the role of maintaining the global security configuration of the network within a dedicated policy repository. We implement this procedure through a specific configuration service that calculates the minimal local configuration required for each PDP to enforce all policies pertinent to that specific resource. Once these local configurations are computed, they are securely distributed to all PDPs within the network through authenticated communication channels.

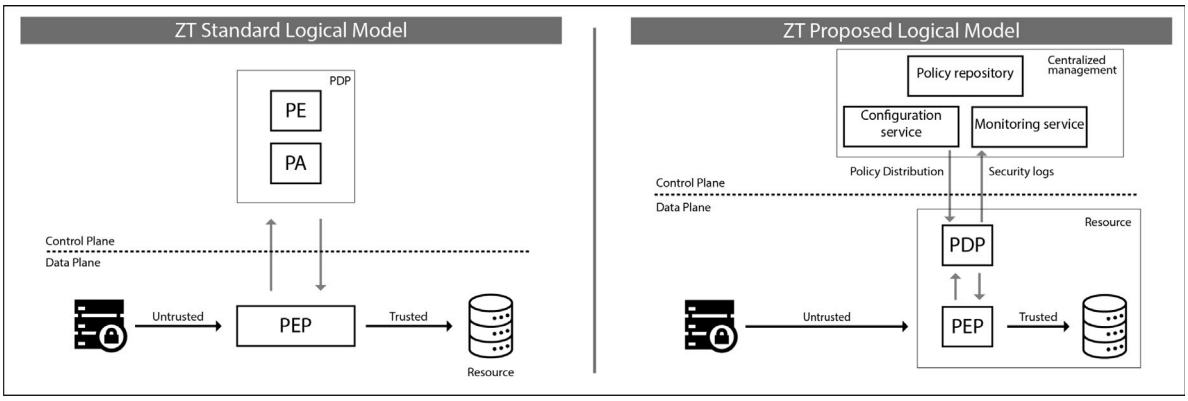


Fig. 1. Zero Trust logical models.

At the same time, a monitoring service continuously collects data from the PDPs to perform a real-time assessment of the network's status. If any inconsistencies or discrepancies are detected between the global configuration and the live status of the network, then the central management system takes some corrective actions by modifying the appropriate security policies. Subsequently, the configuration system recomputes the updated local configurations and distributes them to the PDPs by restoring the network to a secure and compliant state. This distributed architecture ensures the enforcement of security policies even if the central management service becomes temporarily unavailable, thus increasing the scalability and resilience of the network.

One feasible approach to move the PEP to the computing resources is by leveraging smart network interfaces that implement an overlay software-defined network (SDN) on top of the existing infrastructure. The use of an IP-based SDN can provide a consistent abstraction layer for the deployment of security policies in a uniform and standardized manner, regardless of the underlying network. This approach guarantees flexibility and compatibility across different types of environments including cloud-based systems, and makes it easier to integrate the security infrastructure into existing infrastructure. The impact on existing services and applications is minimized since it only requires the update of IP pointers and DNS entries.

To force resource authentication, we use digital certificates at the smart interface level. This approach enables a mutual authentication process to occur before establishing a communication channel. Digital certificates not only verify the identity of the resources but also offer the capability to include additional security metadata, representing the access privileges associated with each resource. During the connection handshake phase, the security information embedded in the digital certificate presented by the initiating resource can be utilized to autonomously determine whether to accept or reject the connection based on a local security policy database, without relying on a central authoritative component, as illustrated in Fig. 2. By adopting this architecture, every connection in the IIoT environment becomes end-to-end encrypted. This added layer of encryption provides enhanced security guarantees, especially for industrial protocols without security features that are vulnerable to cyber-attacks. The SDN layer manages the encryption process that ensures the security and protection of data transmitted between devices and systems from unauthorized access and/or tampering.

The implementation of micro-segmentation solutions can be challenging because it increases the granularity of security policies and the dynamic of the network. Traditional security appliances are often ill equipped to handle similar evolving requirements that augment management complexity and possible performance degradation. The integration of cloud infrastructure further exacerbates these challenges, as there are multiple locations with proprietary interfaces where security policies need to be defined and synchronized. This distributed nature of policy management across multiple environments adds complexity and increases the risk of misconfiguration and inconsistency.

We mitigate the management complexity of micro-segmentation by using a software-defined network (SDN) as the foundational building block of the architecture. By leveraging an SDN, the proposed solution provides a centralized control system that offers a unified interface for defining and managing security policies across the network, including cloud infrastructure. The SDN abstracts the underlying network infrastructure, thus enabling consistent policy enforcement and simplifying the management process. Changes to security policies can be made centrally and propagated to all relevant network components ensuring full synchronization. Additionally, the programmable basis of an SDN allows an automatic orchestration of policy management. Policies can be dynamically defined, modified, and deployed, thus adapting to the changing needs of the network and enabling efficient management in dynamic environments.

The proposed architecture consists of two primary components: the SDN and the centralized management system. The SDN is responsible for the ordinary network operation and provides all the required capabilities to manage and control the flow of data between computing resources such as peer discovery, DNS services, and smart routing. To enhance resiliency and scalability, the SDN is designed to be decentralized. By distributing network functions across multiple devices, the SDN can handle failures and dynamically adapt to changing network conditions.

The centralized management system plays a critical role in monitoring and organizing the underlying infrastructure. It continuously monitors the network, and collects information about the topology and the state of computing resources. This information is used to define and update the network configuration and its security policies. The centralized management system also handles the issuance and distribution of digital certificates, which are essential for resource authentication. Additionally, it manages the security metadata that encodes the access privileges of resources, enabling the enforcement of security policies in a distributed manner.

By separating the operational responsibilities between the SDN and the centralized control system, the proposed system achieves a balance between resilience and centralized management. This division of roles allows for effective security policy enforcement, resource authentication, and dynamic network configuration, ultimately enhancing the security and performance of the overall system.

The transition to a Zero Trust Architecture is typically a complex and long process that is tailored to each specific use case. A key advantage of the proposed solution is its smooth integration path into existing infrastructures. The IP-based SDN acts as a thin abstraction layer that allows for incremental addition of security features in a transparent way, without disrupting existing services and applications. The micro-segmentation strategy further simplifies the transition process. Each individual segment can be migrated independently, providing a

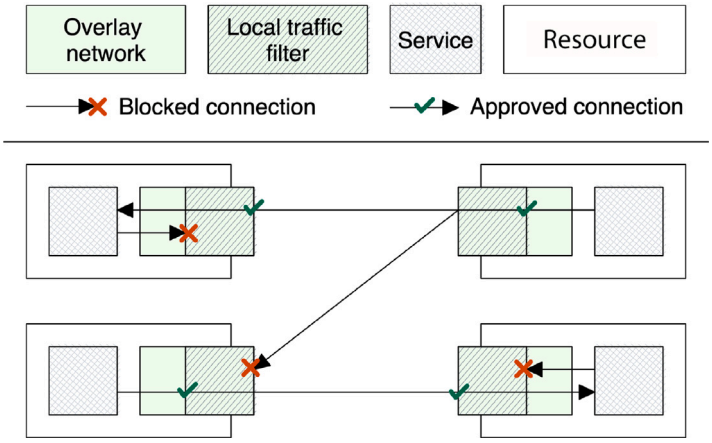


Fig. 2. Schema of how communications are blocked when evaluating local rules by the overlay network (no central controller is used).

clear path towards the full adoption of modern security practices. This incremental approach minimizes disruptions and allows organizations to gradually implement security measures at their own pace, based on their specific needs and requirements.

4. Implementation details

The implementation of a prototype that demonstrates the feasibility of the proposed architecture is based on the open source Nebula software-defined overlay network solution. Nebula offers a range of features that are relevant to the proposed security strategy. It implements by default encrypted point-to-point communication channels and it has the ability to define high-level security groups within the digital certificates. These security groups can be used to define high-level firewall rules that can be enforced directly on the resource allowing for precise control over the network.

We have developed a configuration management system[1] and a certificate distribution service[2] for our infrastructure to streamline the deployment and operation of a micro-segmentation security architecture. The configuration management system enables a centralized definition of the entire network architecture and security policies, while keeping the individual Nebula configuration files in sync. The certificate distribution service automates the process of creating and distributing certificates and configuration files, reducing the complexity and manual effort required to configure large-scale deployments.

To ensure the secure operation of the Nebula network, the certificate distribution service must support a range of security features, including automatic certificate renewal, mutual authentication between clients and the management system, and encryption of communication. The service also needs to provide capabilities for signing client-generated public Nebula keys, generating Nebula key pairs for low computational power clients, creating and signing Nebula certificates containing the client's Nebula public keys, and distributing the signed certificates and configuration files. To manage the signing and distribution process we adapted the existing Enrollment over Secure Transport (EST) [33] protocol to our specific use case. The resulting high-level system architecture consists of three primary components as shown in Fig. 3: the Nebula Enrolling over Secure Transport (NEST) service, the Configuration Management service, and the Certificate Authority (CA) service.

The NEST service is at the core of the management system. It acts as a reverse proxy forwarding client's requests to the internal CA and configuration management services. This is the only component of the architecture publicly exposed to the Internet since its accessibility is essential for performing the initial enrollment process for new resources.

The Certificate Authority (CA) service is responsible for managing digital certificates for the Nebula overlay network. The CA service securely stores the root certificate and its master key, which are required to issue new client certificates and renew expired ones. To ensure maximum security, the CA service is deployed in a separate network only accessible through its REST API by the NEST service.

The configuration service manages the centralized repository containing the entire network configuration along with all the security policies. It provides an API to let client request their initial configuration during the enrollment phase and receive every subsequent configuration update.

4.1. Testbed infrastructure

The implemented infrastructure and network configuration are represented in Fig. 4. This diagram describes the deployed network, the arrangement, and connections of the various components of the testbed.

The test network was deployed using a combination of Microsoft's Azure cloud platform and personal devices. Microsoft Azure was utilized for deploying all NEST services and Linux clients. Additionally, specific personal devices were used for deploying certain components: a Raspberry Pi4b was used for the Lighthouse component, which is a fundamental piece of a Nebula SDN and provides the virtual name and virtual IP address resolution service to the whole network A real Windows PC was used for the Windows client; a smartphone was used for the Android client.

All resources deployed on the cloud platform were provisioned as "B1ls" Azure virtual machines, which are the least performing available devices with only 512MB of RAM and one virtual CPU. An exception is represented by the Configuration Management service, which was provisioned as a "B2s" virtual machine with 2 virtual CPUs and 4 GB of RAM. The choice of using low-resource instances in the deployment of the security infrastructure is done to highlight the minimal overhead introduced by the system.

All virtual networks in the deployment were configured to deny all incoming traffic by default. An exception was made for UDP traffic on port 4242, which is the default port used by Nebula for communication. This allows clients to establish secure connections through the Nebula overlay network. Furthermore, the NEST service exposes port 8080 to the Internet, allowing clients to access it for enrollment and re-enrollment operations. The Configuration Management service and Certificate Authority are deployed in an isolated network that is only accessible by the NEST service.

[1] https://github.com/securityresearchlab/dhall-nebula
[2] https://github.com/securityresearchlab/nebula-est

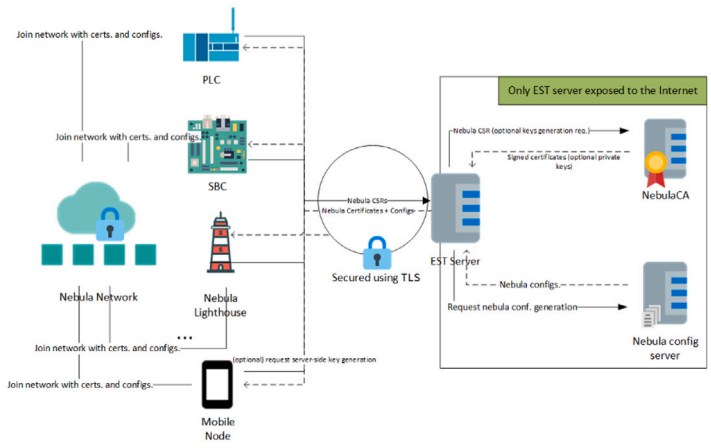


Fig. 3. High level architecture.

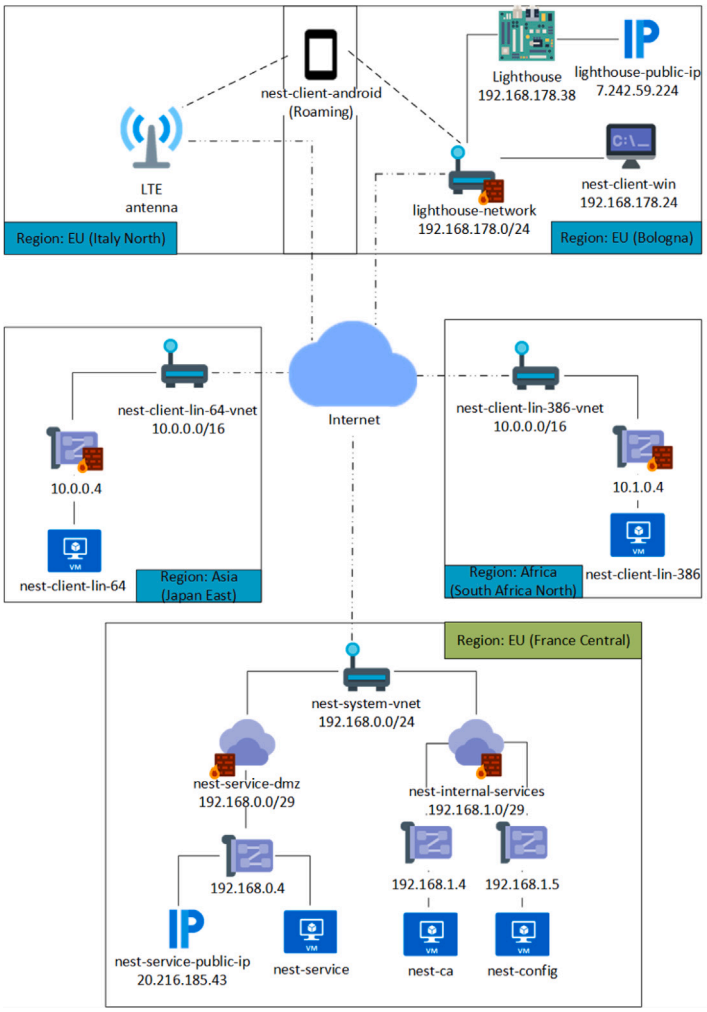


Fig. 4. Testbed architecture.

4.2. Clients

We develop the *NEST client* to enable resources to interact with the proposed security infrastructure. A client is a daemon that needs to be deployed along with the Nebula software, and is responsible for all communications with the central NEST service. Moreover, it manages the Nebula network interface including configuration updates and digital certificate renewals. The NEST client is written in GO programming language which is the same language used by Nebula. Therefore, if a component can run Nebula, it is able to run the NEST client. Furthermore, the choice to use the GO language allows us to develop code that is portable to different platforms, such as MIPS or ARM architectures, which are prominent in the IoT and embedded devices world. By leveraging the GO language cross-compilation capabilities,

HOSTNAME	OS/ARCH.	REGION	NEBULA IP	PUBLIC IP
nest-service	Linux/x64	EU (France Central)	192.168.80.3/24	Yes
nest-ca	Linux/x64	EU (France Central)	192.168.80.1/24	No
nest-config	Linux/x64	EU (France Central)	192.168.80.2/24	No
lighthouse	Linux/arm64	Bologna	192.168.90.1/24	Yes
nest-client-lin-64	Linux/x64	Asia (Japan East)	192.168.90.2/24	No
nest-client-win	Windows/x64	Bologna	192.168.90.3/24	No
nest-client-lin-386	Linux/i386	Africa (South Africa North)	192.168.90.4/24	No
nest-client-android	android/arm64	Roaming (Italy North - Bologna)	192.168.90.5/24	No

Fig. 5. Resources description.

this should be easy enough with the given attention to the libraries to be used, trying to always choose the ones that are most compatible with every architecture we may need to support.

We have successfully deployed the NEST client on various CPU architectures and operating systems within a hybrid cloud setup, demonstrating the adaptability of our system across different environments. Detailed information regarding each resource can be found in Fig. 5.

The test network was deployed through a combination of Microsoft's Azure cloud platform and personal devices. Microsoft Azure was utilized for deploying all the services and Linux clients. Additionally, specific personal devices were used for deploying certain components: a Raspberry Pi4b was used for the lighthouse client, a real Windows PC was used for the Windows client, and a smartphone was used for the Android client.

All communications between the NEST client and the NEST service must be authenticated. The implemented authentication scheme relies on a secret token that must be provided to the client during its initialization. To generate the secret token, HMAC is applied to the client hostname using a 256-bit symmetric key. This approach ensures that the NEST service can verify the authenticity of the client's request. By using the resource hostname for secret generation, the scope of the token is limited. Even if the secret token is compromised, it cannot be used to gain access to other resources. We assume that the secret token can be securely provided to the resource. Since we designed this architecture to be used in various environments, from cloud computing to IoT devices, we have not codified in the NEST protocol a specific process for secret management. This facilitates the adoption into existing environments since it is possible to choose an appropriate solution for each use case, taking into account the specific security considerations and requirements of the infrastructure.

4.3. Configuration management

Configuration is a critical aspect of every complex system. Misconfigurations can result in improper functionality, performance degradation, noncompliance, and security vulnerabilities. While security policies are acknowledged as vital, the importance of configuration management in system security is often undervalued. In recent years, there has been a growing recognition of the challenges involved in configuring secure systems, resulting in the increasing popularity of Policy-as-Code and Configuration-as-Code. Our micro-segmentation approach places a significant emphasis on configuration, specifically in terms of its definition, management, and application.

To ensure efficient management of the network configuration, we have developed a system capable of generating individual node configurations from a high-level specification of the desired security policies, enabling their implementation. The entire network's configuration can be centrally defined and subsequently distributed to network agents for distributed operation.

We have designed and implemented a library, using the Dhall language, to specify the configurations for a micro-segmentation network topology of arbitrary complexity. This library includes type definitions for every element: such as resources, certificate authorities, security groups, and firewall rules. Additionally, it provides multiple helper functions that facilitate the modular specification of the network and a set of validation functions to check the structural integrity of the final configuration. The validation can be executed during the type-checking phase, allowing for the verification of configuration correctness before its deployment, which greatly minimizes the likelihood of accidental errors.

The configuration includes a top-level component named "Network". Fig. 6 shows a subset of the testbed configuration related to security policies. The essential parts of the network definition are:

- the *hosts* field which declares all the authorized hosts of the network.
- the *groups* field which contains the definition of all security groups.
- the *connections* field in which are specified the security policies governing the network.

In this example, the first policy enables SSH connections between resources belonging to the *andr* security group. The second policy allows the establishment of TCP connections between resources in the *lin* security group, unlike the first policy, this rule does not specify the port of the connection. This configuration highlights the modularity of the approach, where various components of the configuration can be defined in separate files and then imported into the main configuration file. By separating these components into distinct files, it becomes easier to manage and organize the configuration. Changes and updates can be made to specific files without affecting the entire configuration.

In addition, we have developed an automation tool that can take the high-level network definition as input and generate the required configuration files for deploying the network nodes. This tool is also responsible for the validation of incoming certificate signing requests (CSR). In fact, before signing the final certificate, the tool performs a security check by comparing the information included in the CSR with the security properties specified in the network configuration to prevent attempts of privilege escalation.

```
let ssh_port = nebula.Port.Port 22

-- Firewall rule
let andr_connection
    : nebula.Connection
    = nebula.mkIntraGroupConnection
        -- Which port
        ssh_port
        -- Which protocol
        nebula.Proto.TCP
        -- Which host/group it affects
        andr_group
        --Additional information
        (None Text)
        (None Text)

let lin_connection
    : nebula.Connection
    = nebula.mkIntraGroupConnection
        nebula.Port.AnyPort
        nebula.Proto.TCP
        lin_group
        (None Text)
        (None Text)

let network
    : nebula.Network
    = { hosts = hosts_list
      , groups = [ all_group, lin_group, andr_group ]
      , connections = [ lin_connection, andr_connection, outbound_connection,
icmp_connection ]
      , blocklist = [] : List Text
      , cipher = nebula.Cipher.Chachapoly
      , ip_mask = 24
      }
```

Fig. 6. Testbed configuration example.

4.4. Certificate authority

The Nebula overlay network uses custom certificates that are not X.509 compliant, requiring the development of a custom PKI solution. The solution must accommodate Nebula Certificate Signing Requests (NCSR) and key pair generation while offering flexibility in managing the additional metadata necessary for the security infrastructure. We have developed a service that functions as the Certificate Authority (CA) for the Nebula network. Recognizing the crucial role of the digital certificates within Nebula, we prioritize minimizing the attack surface of this service. Therefore, we have implemented an isolated Nebula network specifically for deploying this component. This network is accessible solely from the NEST service, ensuring that all interactions are authorized and limiting any potential unauthorized access.

The Nebula overlay network does not support traditional Certificate Revocation Lists (CRLs). Instead, it offers the capability to define certificate blacklists within the node configuration. Our configuration management solution (Section 4.3) makes this process viable at scale by providing a centralized interface. Additionally, the CA maintains a database of all authorized clients within the network, enabling a straightforward approach to phasing out compromised certificates. By removing the certificate's public key from the CA's database, the renewal process is halted. The combination of a short validity window for digital certificates and the blacklisting mechanism provides a robust method for managing the infrastructure's certificates. Moreover, each node in the Nebula overlay network has the capability to trust multiple Certificate Authorities simultaneously. This feature is particularly advantageous for establishing CA federations, where one CA remains online while the others serve as backups. Such an arrangement ensures continuity and enables a seamless transition when a CA certificate reaches its expiration or it is compromised.

4.5. NEST service

The NEST service serves as the central component of the proposed architecture. It implements the EST (Enrollment over Secure Transport) protocol adaptation for the Nebula network, exposing the necessary REST endpoints for clients to perform certificate provisioning and renewal. To maintain an up-to-date database of all network resources and configurations, the NEST service periodically queries the CA and Configuration services. This ensures that the NEST service has the latest information regarding certificates and network configurations, allowing it to effectively manage and coordinate the provisioning process.

The two main services provided are the Enroll function for the first-time certificate provision and the Re-enroll function for certificate renewal.

Enrollment. Fig. 7 outlines the process flow of the Enrollment request. As noted, this process can only start once the NEST service has updated its internal policy database to validate incoming requests. The enrollment process begins with an authenticated request from the client, as described in Section 4.2. The NEST service performs a validation check on the data received and notifies the resource that the enrollment process has started. The resource then requests the digital certificate template from the NEST service, this template is being generated by the NEST service according to the current security policies. If the *ServerKeygen* function is utilized, both the public and private keys will be generated server-side and sent to the resource through an encrypted channel, otherwise, the resource will generate the key pair locally and include only the public key in the certificate signing request. The NEST service delegates the tasks of certificate signing and configuration generation to the back-end services. Once these processes are completed, the NEST service provides the final result to the resource. With the signed certificate and the YAML configuration, the resource is then able to initialize the Nebula interface and join the network. In addition, the resource sets up a timer to trigger the Re-enroll function before the certificate expires. This ensures that the resource can automatically renew its certificate in a timely manner, preventing any disruptions in its connectivity within the Nebula network.

Fig. 8 shows the request log of the NEST service that correctly reports all the steps required for the enrollment of the lighthouse component.

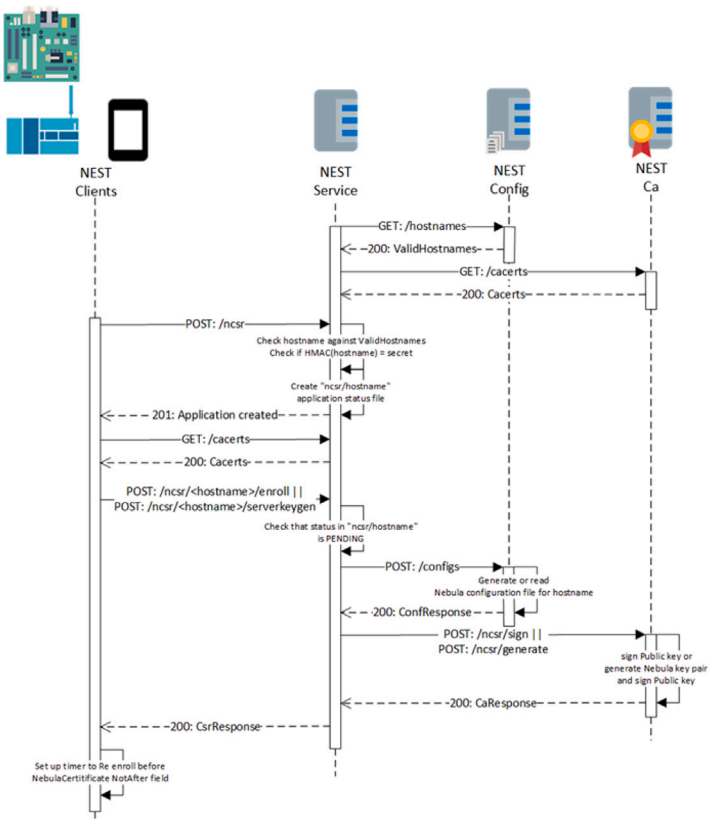


Fig. 7. Enrollment process flow.

```
                                  # NEST SERVICE
     DateTime       |  HTTP Code  | Request time   |  Requestor IP    |      HTTP method and url
[GIN] 2023/02/15 - 12:28:02 |  200  |     149.199µs |    7.242.59.224 |  GET    "/cacerts"
[GIN] 2023/02/15 - 12:28:03 |  201  |     458.199µs |    7.242.59.224 |  POST   "/ncsr"
[GIN] 2023/02/15 - 12:28:03 |  200  |  270.688441ms |    7.242.59.224 |  POST   "/ncsr/lighthouse/enroll"
                                  # NEST CA
[GIN] 2023/02/15 - 12:28:03 |  200  |  245.880957ms |    192.168.80.3 |  POST   "/ncsr/sign"
                                  # NEST CONFIG
[GIN] 2023/02/15 - 12:28:03 |  200  |  12.115584ms  |    192.168.80.3 |  GET    "/configs/lighthouse"
```

Fig. 8. Logs of the first enrollment procedure.

Re-enrollment. The Re-enrollment procedure is utilized to extend the validity of a certificate, its process flow is reported in Fig. 9. During the initial request, the current certificate is presented as an authorization mechanism. The NEST service performs several checks, including verifying the validity of the certificate, ensuring there is no ongoing renewal process, and verifying that the certificate has not been blacklisted as per the security policies. If all the checks yield positive results, the NEST service proceeds to send the signing request to the CA. The signing request can involve the signing of the public key or the regeneration of a new key pair, depending on the requirements. Additionally, the NEST service requests the Configuration service to provide the updated version of the Nebula configuration. This ensures that the resource always uses the most recent configuration information. Finally, the renewed certificate and the updated Nebula configuration are returned to the resource that can continue to participate in the Nebula network.

Once all resources have been successfully enrolled in the network, we proceed to test the re-enrollment process. To simulate a high frequency of enrollment requests, we configure the certificate validity to two minutes. The logs reported in Fig. 10 delineates the details of the re-enrollment process.

The top section displays the logs of the NEST service, indicating the re-enrollment requests received from different clients. This demonstrates the continuous renewal of certificates to maintain access within the network. The bottom-left section represents the logs of the Certificate Authority, showcasing the API calls made by the NEST service to sign new certificates. The bottom-right section reports the logs of the Configuration Management service. It shows the API calls made by the NEST service for configuration updates during the re-enrollment process ensuring that all resources can benefit of the most up-to-date configurations.

The NEST client is specifically designed to minimize interference with the normal networking operations of the enrolled resources. By using TCP-over-UDP it is possible to restart a Nebula network interface without dropping existing open connections. The NEST client leverages this feature to provide a disruption-free re-enrollment process. The NEST client operates as a standalone process, coexisting with the Nebula interface. It continuously monitors the resource status within the SDN by establishing periodic communications with the NEST service. During the re-enrollment process, the NEST client acquires updated configurations, including certificates and security policies, from the central server while the Nebula interface remains operational. Once all necessary data is obtained, the NEST client performs an atomic operation by overwriting the current working configurations and restarting the Nebula interface. This operation introduces a localized spike in connection latency, typically on the order of 100 ms, as determined by our testing. Importantly, this spike does not disrupt ongoing operations. It is worth noting that on Windows clients, this restart process may take longer, approximately 3–4 s. This discrepancy is primarily due to the optimization of the Nebula interface for Windows. We anticipate that this issue will be resolved in future Nebula releases, aligning Windows performance with that of other platforms.

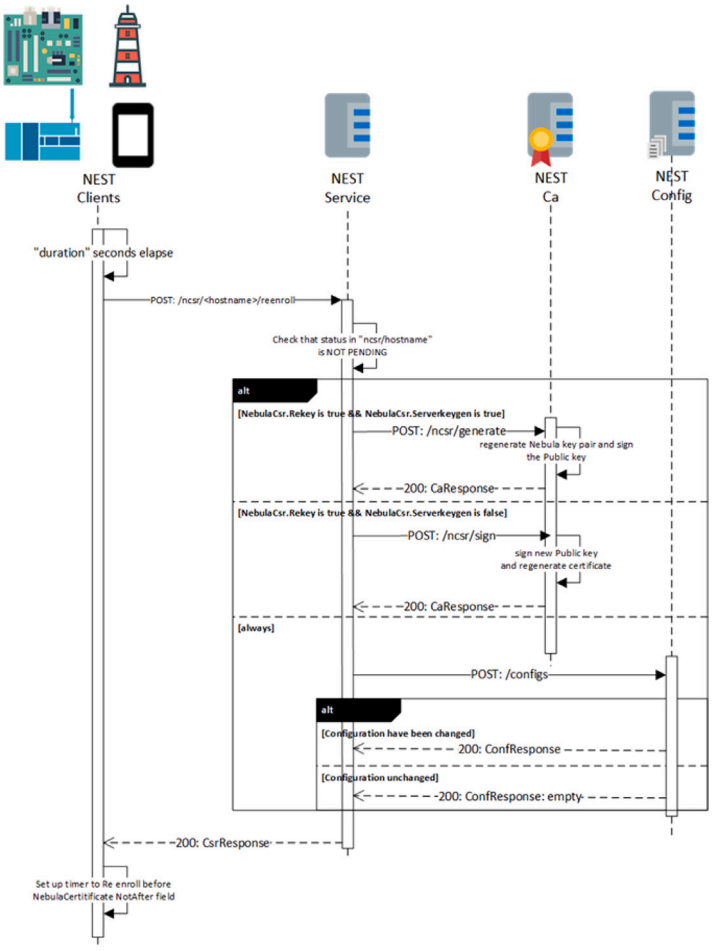


Fig. 9. Re-enrollment process flow.

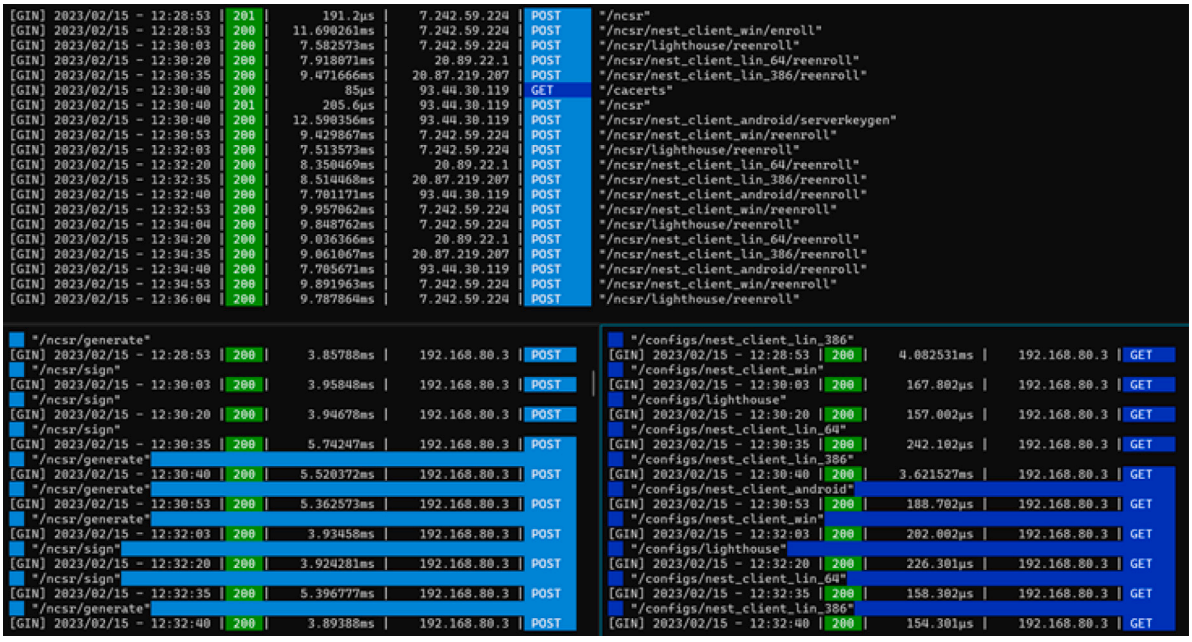


Fig. 10. Re-enrollment process logs.

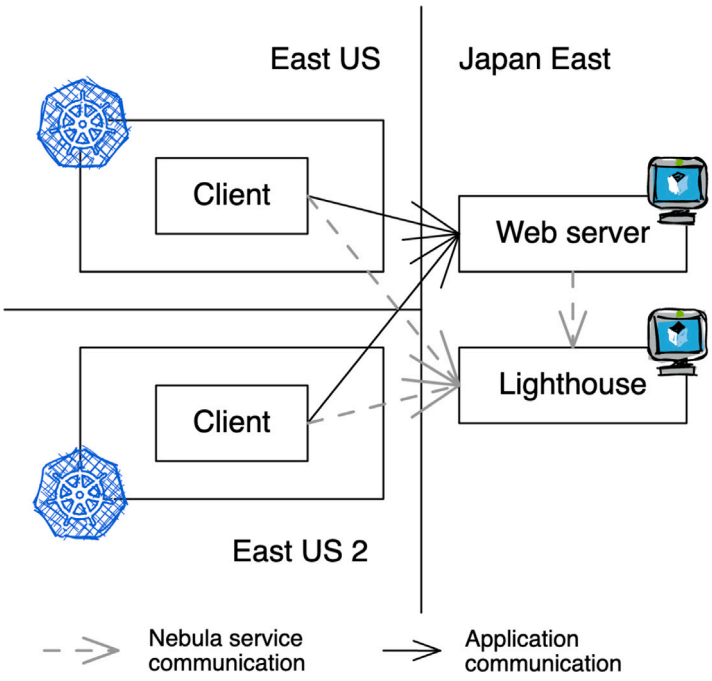


Fig. 11. Performance test setup.

Table 1
Nebula overheads.

RAM	CPU	Connections	CPU usage	RAM usage
1 GB	1vCPU	50	18.4%	1.5%
4 GB	2vCPU	200	12.8%	<1%
8 GB	2vCPU	>300	15.1%	<1%

5. Performance results

The goal of the experiments is to demonstrate the feasibility of the prototype and to evaluate the overhead with a focus on resource-constrained devices. The testbed consists of two virtual machines and two Kubernetes clusters deployed on a cloud infrastructure spread across multiple availability zones as shown in Fig. 11. One virtual machine hosts the lighthouse that is necessary for node discovery. The other machine hosts a Web server that manages HTTP GET requests. HTTP clients are executed on the Kubernetes clusters where each cluster node is equipped with 6 a vCPU and 16 GB of RAM.

Each client performs an HTTP request to the Web server every 250 ms, for 15 min. In total, we execute twelve tests for increasing numbers of clients going from 50 to 300 with steps of 50 clients. The clients are equally split into the two Kubernetes clusters in different data centers. The results of resource utilization for different configurations are presented in Table 1. Notably, even a Web server equipped with just 1 GB of RAM and 1 vCPU can effortlessly manage over 50 concurrent connections. Furthermore, even when hardware resources are scaled up, the overhead caused by Nebula remains remarkably small. These results demonstrate the efficiency and versatility of the proposed solution in accommodating diverse hardware configurations.

Figs. 12, 13 and 14 report the response times of HTTP requests obtained from the test scenarios. Each figure presents a side-by-side comparison, with the left chart representing the data collected while using the Nebula overlay network, and the right chart showing the results referring to the baseline architecture not using our proposal. We can observe the noticeably better results of the proposal with respect to the baseline in terms of lower response time and failure rate. The performance difference can be attributed to different effects including the distributed architecture and the TCP-over-UDP operation mode used by Nebula SDN. In our test scenario, where the Web server is under considerable stress due to increasing connections, the network congestion control algorithms of TCP tend to misclassify the increased latency and reduce the transmission rate of the clients.

Fig. 14 shows multiple horizontal bands corresponding to the exponential back-off triggered by the congestion control algorithm. The alternative use of the TCP-over-UDP platform can continue to transmit packets tolerating the increased response time of the Web server. This feature is greatly advantageous for industrial systems and IoT applications that are characterized by stringent real-time requirements.

All the results show the practical applicability of the proposed SDN-based solution in real-world environments, as it exhibits no performance degradation even when deployed on resource-constrained devices. The peer-to-peer architecture of the SDN underlying the security infrastructure enables the network to scale efficiently with the number of resources by distributing the workload of policy enforcement and maximizing network performance and utilization. This is quite effective in industrial networks where a substantial portion of communications occurs within the local network without accessing remote resources. We can conclude that the overall scalability of this solution is constrained just by the management layer.

Given that each individual enrollment/re-enrollment flow is logically independent, the NEST Service, which is responsible for handling communications with the resources, can be easily replicated. Hence, all these operations can be parallelized. A similar independence holds true for the certification authority, where each request is stateless because all the relevant state information is encoded inside the certificate signing requests. Upon receiving a CSR, this authority can validate it without depending on other external components to check the provided digital signatures. Hence, both the NEST Service and the certificate authority can be independently scaled horizontally based on the volume of incoming requests, which is a function of the number of managed resources and the certificate duration length.

The Configuration Service, which is responsible for maintaining the global state of the network, represents the possible bottleneck of the architecture as it must be queried in each enrollment/re-enrollment flow to provide the most updated and consistent information. To achieve horizontally scalability of this component it is crucial to properly manage the data consistency between the replicas. It is worth noting

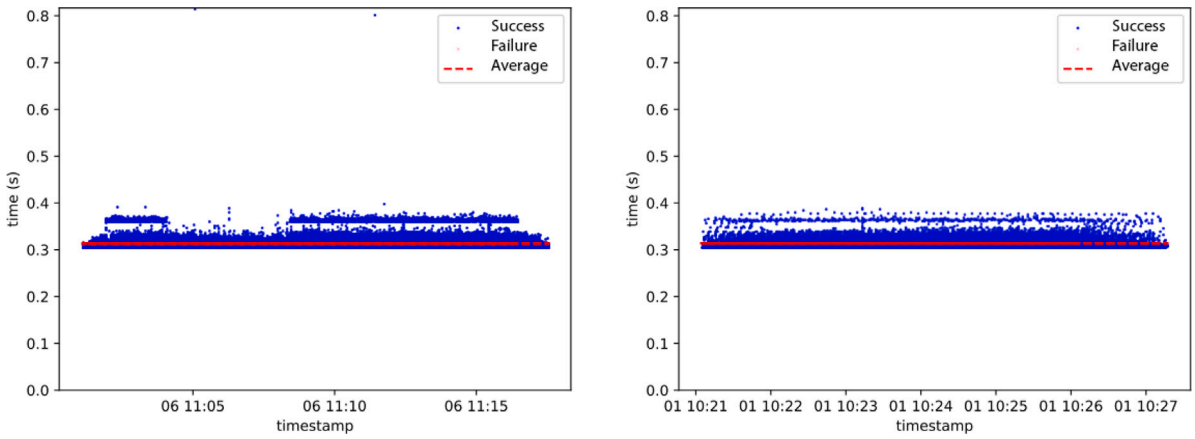


Fig. 12. Response times (50 clients).

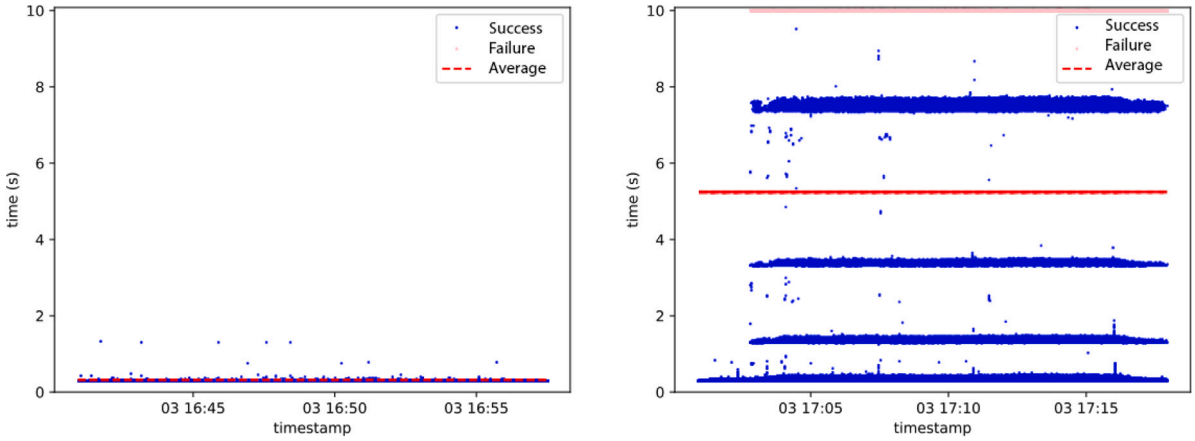


Fig. 13. Response times (150 clients).

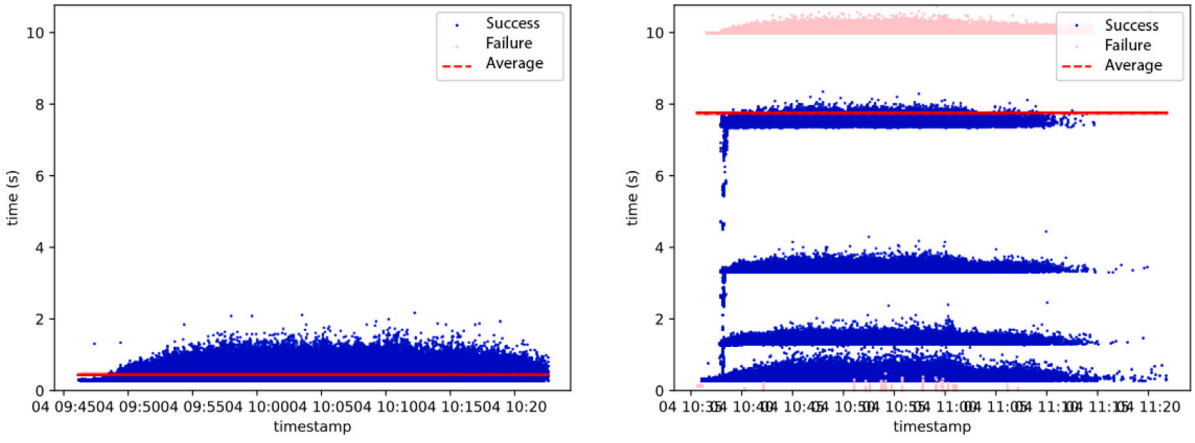


Fig. 14. Response times (300 clients).

that the rate of change for the security configurations managed by this service should be fairly limited, considering they are critical data requiring high-privilege access. In addition, by utilizing security tags to define the policies, precise rules can be established in abstract terms. This approach not only keeps the overall configuration manageable but it also minimizes the number of policy changes required to modify the topology of the network, even for substantial refactors. Consequently, the access pattern to this service is read-heavy and cache-friendly, thus enabling a wide range of performance optimizations to scale it as the network grows.

Other centralized SDN solutions often rely on address-based security policies to implement a micro-segmentation network topology. In order to keep the security policies synchronized with the underlying network, this solution necessitates frequent and rapid updates especially

in highly dynamic networks. For this reason, similar approaches do not scale.

6. Discussion

The flexibility and low overhead of the presented architecture can facilitate the convergence of Information Technology (IT) and Operational Technology (OT) systems within hybrid networks. Many state-of-the-art security solutions are primarily designed for IT systems and operate at high layers of the network stack. Integrating these solutions into industrial infrastructures often requires substantial modifications to existing hardware or the creation of complex interfaces. These adaptations can introduce new vulnerabilities by compromising the original security assumptions. The absence of a unified security abstraction that effectively bridges the realms of both IT and OT systems presents a further vulnerability in hybrid networks. Similar issues extend beyond technical concerns because they influences the overall cyber risk management perspective in large industrial networks and critical systems, where the OT part typically has the largest economic value. Despite their central role and increased vulnerabilities, OT systems have often received fewer security attention than that characterizing their IT counterparts.

The proposed solution aims to introduce a unified and versatile security layer that is strategically designed to address the evolving needs of contemporary hybrid infrastructures. They are characterized by a diverse range of technology stacks, hardware architectures, and the imperative for seamless integration with legacy systems. We leverage Nebula as the low-level interface to build a security infrastructure because it is widely supported on many different classes of hardware. By working at the network layer, our design provides the same protection to both IT and OT components. Furthermore, the clear separation between the network layer and the security layer allows independent scaling of these components along the rest of the infrastructure.

Our proposal can be easily augmented with other security solutions that operate at higher levels of the software stack since they do not interfere with each other. This flexibility allows organizations to leverage the existing security infrastructure and combine them to create a comprehensive and layered security strategy.

Devices lacking native support for Nebula can be integrated by introducing an additional device working in *Nebula unsafe mode*. In this operational mode, the device assumes the role of a Network Address Translation (NAT) gateway between the unsupported hardware and the broader Nebula network. The term *unsafe mode* derives from the fact that when one device provides this NAT service to multiple resources, the connections between them are initially not encrypted. This possible drawback can be solved by integrating a low-power NAT device into each unsupported resource. Hence, there is minimal effort to integrate the proposed solution into an existing infrastructure.

We evidence the pivotal role in establishing and maintaining a secure infrastructure played by the monitoring infrastructure. As no single node of the network possesses a global view of the entire topology, the central configuration service and the information of the monitoring service assume a crucial role in coordinating and orchestrating the diverse components ensuring the secure operation of the overall network. The distribution of some functions in separate services, that are traditionally associated with the Policy Decision Point (PDP), increments the value of our solution. A simplified version of the PDP is integrated directly with the Policy Enforcement Point (PEP) within the final network resource. The PDP, which is deployed alongside a resource, maintains a local repository comprising all pertinent policies required to autonomously make security decisions regarding the underlying resource with only a partial knowledge of the network. This capability is achieved because security policies are defined through high-level expressions that rely on the security metadata contained within the digital certificates of network resources, rather than being based on low-level firewall rules. As a result, even in highly dynamic network environments that are characterized by resources with short lifespans and frequent topology changes, these defined policies can continue to operate effectively without significant modifications during normal operations. Thanks to this approach, a security policy requires an update only when there is a fundamental change in the underlying security semantic through the network or as a response to a security breach requiring remediation activities.

This design presents a set of deliberate trade-offs that are optimized for the specific requirements of a decentralized hybrid IT-OT network, a configuration commonly found in scenarios where a central IT system coexists with various OT systems across multiple production plants that are geographically distributed. These trade-offs are instrumental in achieving the desired performance profile and network resilience. Traditional centralized PDPs introduce a mandatory round-trip for every authorization request, even for connections within the same industrial plant. This is not compatible with the real-time requirements of large industrial systems especially when there is considerable physical distance between the industrial plant and the central system. In contrast, our system leverages point-to-point connections, minimizing latency and maximizing transmission speed between individual resources. Additionally, centralized PDPs typically process all network traffic in real time to enforce security policies. While effective, this centralized processing creates a single point of failure that has the potential to disrupt the entire network during an outage or failure event. In contrast, a distributed design, as employed in our solution, enhances network resiliency and scalability by distributing the enforcement of security policies.

On the other hand, it is important to recognize that a dynamic alteration of the network topology is not an instantaneous process. When changes occur, the configuration service must push the updated configurations to all nodes within the network, and this process takes time to propagate across the network. Moreover, it necessitates continuous monitoring to ensure that the change propagation is complete and that all nodes are operating with the updated configurations. This process is critical, especially in scenarios where a resource has been compromised. In such cases, the time required to execute remediation activities may be longer compared to a centralized system. However, it is important to highlight that the micro-segmentation topology minimizes the potential blast radius of damage resulting from a security breach. Furthermore, the decentralized nature of the network inherently raises the bar for privilege escalation and lateral movement attempts. As a consequence, similar actions become more challenging and time-consuming for attackers.

This architecture also makes global exploration more difficult for an intruder. Given that all connections are peer-to-peer and end-to-end encrypted, the traditional approach of capturing the raw traffic from a centralized security appliance that processes all the traffic is no longer possible. In this context, the PDP deployed on the network resource plays a dual role. It governs policy enforcement and works as a reporting node, forwarding comprehensive security information related to instances of policy violations to the central monitoring service. The aggregation of these sets of data is essential for reconstructing a global view of the security status of a network. The presence of mutual authentication through digital certificates can significantly elevate the ability to detect anomalous activities and lateral movements. Mutual authentication drastically increases the complexity of carrying out spoofing attacks, making them exponentially more difficult to execute. The strong attestation provided by mutual authentication, coupled with a micro-segmentation architecture using high-level and very precise security policies, makes policy violations a high signal-to-noise indicator of potentially malicious or unauthorized activities within the network.

7. Conclusions

We propose a novel security architecture that utilizes software-defined networks to implement a micro-segmentation Zero Trust Architecture specifically designed for industrial systems and heterogeneous

environments. Our solution addresses the issues of existing security approaches by offering a flexible configuration management system that can be deployed in various industrial environments, including multi-cloud and hybrid-cloud setups.

The implementation leverages the Nebula software-defined network solution as the underlying infrastructure. We also propose a configuration management system that automates the generation and distribution of individual resource configurations and digital certificates based on the defined security policies. This system ensures that each resource is equipped with the necessary security credentials to participate in secure communication and interaction within the network. Additionally, we create a client that manages the smart network interface of the resources. This client supports multiple operating systems and hardware configurations, enabling seamless integration of a diverse range of devices into our architecture.

By integrating all these components, we demonstrate that our security approach identifies a possible comprehensive solution for securing modern infrastructures including IIoT systems, different networks, and even cloud platforms. It provides the necessary flexibility and adaptability to meet the unique requirements of complex industrial environments that can be seamlessly integrated into existing infrastructure without modifications to the final applications. The only necessary changes are represented by modifications of IP addresses and DNS pointers that are centrally managed. The proposed architecture paves the way for a novel approach of extending even to IIoT environments what we consider the necessary Zero Trust cybersecurity principles.

The current monitoring service primarily focuses on detecting disparities between the network-defined topology in the global configuration and the actual network status. In future work, we plan to enhance the system by proposing an analytical engine capable of identifying malicious activities based on the security information generated by the network's resources. This represents another step in augmenting the security posture. By leveraging artificial intelligence, we aim to proactively detect and mitigate potential threats, thereby increasing the resilience against malicious actors.

CRediT authorship contribution statement

Claudio Zanasi: Methodology, Software, Writing – original draft. **Silvio Russo:** Validation, Writing – original draft, Writing – review & editing. **Michele Colajanni:** Funding acquisition, Methodology, Project administration, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Y. Wang, J. Wei, K. Vangury, Bring your own device security issues and challenges, in: 2014 IEEE 11th Consumer Communications and Networking Conference, CCNC, 2014, pp. 80–85, http://dx.doi.org/10.1109/CCNC.2014.6866552.

[2] T.A. Wani, A. Mendoza, K. Gray, BYOD in hospitals-security issues and mitigation strategies, in: Proceedings of the Australasian Computer Science Week Multiconference, ACSW '19, Association for Computing Machinery, New York, NY, USA, 2019, http://dx.doi.org/10.1145/3290688.3290729.

[3] Zscaler, VPN risk report, 2021, https://info.zscaler.com/rs/306-ZEJ-256/images/2021-VPN-Risk-Report-Zscaler.pdf.

[4] E. Bertino, Zero trust architecture: does it help? IEEE Secur. Privacy 19 (05) (2021) 95–96.

[5] S. Rose, O. Borchert, S. Mitchell, S. Connelly, Zero Trust Architecture, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, 2020, http://dx.doi.org/10.6028/NIST.SP.800-207.

[6] R. Ward, B. Beyer, BeyondCorp: A new approach to enterprise security, 39, (6) 2014, pp. 6–11, ;login:.

[7] C. Cunningham, The Zero Trust Extended (ZTX) Ecosystem, Cambridge, 2018.

[8] J. Kindervag, et al., Build security into your network's dna: The zero trust network architecture, Forrester Res. Inc 27 (2010).

[9] C. Zanasi, F. Magnanini, S. Russo, M. Colajanni, A zero trust approach for the cybersecurity of industrial control systems, in: 2022 IEEE 21st International Symposium on Network Computing and Applications, NCA, vol. 21, IEEE, 2022, pp. 1–7.

[10] N.F. Syed, S.W. Shah, A. Shaghaghi, A. Anwar, Z. Baig, R. Doss, Zero trust architecture (ZTA): A comprehensive survey, IEEE Access 10 (2022) 57143–57179, http://dx.doi.org/10.1109/ACCESS.2022.3174679.

[11] N. Basta, M. Ikram, M.A. Kaafar, A. Walker, Towards a zero-trust micro-segmentation network security strategy: An evaluation framework, in: NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, 2022, pp. 1–7, http://dx.doi.org/10.1109/NOMS54207.2022.9789888.

[12] J. Voas, R. Kuhn, P. Laplante, S. Applebaum, Internet of things (IoT) trust concerns, NIST Tech. Rep. 1 (2018) 1–50.

[13] A. Wasicek, The future of 5G smart home network security is micro-segmentation, Netw. Secur. 2020 (11) (2020) 11–13.

[14] S. Li, M. Iqbal, N. Saxena, Future industry internet of things with zero-trust security, Inf. Syst. Front. (2022/03/10) http://dx.doi.org/10.1007/s10796-021-10199-5.

[15] Z. Zaheer, H. Chang, S. Mukherjee, J. Van der Merwe, Eztrust: Network-independent zero-trust perimeterization for microservices, in: Proceedings of the 2019 ACM Symposium on SDN Research, SOSR '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 49–61, http://dx.doi.org/10.1145/3314148.3314349.

[16] L. Ferretti, F. Magnanini, M. Andreolini, M. Colajanni, Survivable zero trust for cloud computing environments, Comput. Secur. 110 (2021) 102419.

[17] V.N.S.S. Chimakurthi, The challenge of achieving zero trust remote access in multi-cloud environment, ABC J. Adv. Res. 9 (2) (2020) 89–102.

[18] M. Firdhous, O. Ghazali, S. Hassan, Trust management in cloud computing: A critical review, 2012, arXiv preprint arXiv:1211.3979.

[19] I.M. Abbadi, M. Alawneh, A framework for establishing trust in the cloud, Comput. Electr. Eng. 38 (5) (2012) 1073–1087, http://dx.doi.org/10.1016/j.compeleceng.2012.06.006, URL https://www.sciencedirect.com/science/article/pii/S0045790612001152, Special issue on Recent Advances in Security and Privacy in Distributed Communications and Image processing.

[20] S. Mehraj, M.T. Banday, Establishing a zero trust strategy in cloud computing environment, in: 2020 International Conference on Computer Communication and Informatics, ICCCI, 2020, pp. 1–6, http://dx.doi.org/10.1109/ICCCI48352.2020.9104214.

[21] F. Bannour, S. Souihi, A. Mellouk, Distributed SDN control: Survey, taxonomy, and challenges, IEEE Commun. Surv. Tutor. 20 (1) (2017) 333–354.

[22] Y.E. Oktian, S. Lee, H. Lee, J. Lam, Distributed SDN controller system: A survey on design choice, Comput. Netw. 121 (2017) 100–111.

[23] T.H. Szymanski, The "cyber security via determinism" paradigm for a quantum safe zero trust deterministic internet of things (IoT), IEEE Access 10 (2022) 45893–45930, http://dx.doi.org/10.1109/ACCESS.2022.3169137.

[24] C. DeCusatis, P. Liengtiraphan, A. Sager, M. Pinelli, Implementing zero trust cloud networks with transport access control and first packet authentication, in: 2016 IEEE International Conference on Smart Cloud, (SmartCloud), 2016, pp. 5–10, http://dx.doi.org/10.1109/SmartCloud.2016.22.

[25] Y. Maleh, Y. Qasmaoui, K. El Gholami, Y. Sadqi, S. Mounir, A comprehensive survey on SDN security: threats, mitigations, and future directions, J. Reliab. Intell. Environ. 9 (2) (2023) 201–239.

[26] J.C.C. Chica, J.C. Imbachi, J.F.B. Vega, Security in SDN: A comprehensive survey, J. Netw. Comput. Appl. 159 (2020) 102595.

[27] S. Khorsandroo, A.G. Sánchez, A.S. Tosun, J.M. Arco, R. Doriguzzi-Corin, Hybrid SDN evolution: A comprehensive survey of the state-of-the-art, Comput. Netw. 192 (2021) 107981.

[28] J.A. Donenfeld, Wireguard: next generation kernel network tunnel, in: NDSS, 2017, pp. 1–12.

[29] D. Bringhenti, J. Yusupov, A.M. Zarca, F. Valenza, R. Sisto, J.B. Bernabe, A. Skarmeta, Automatic, verifiable and optimized policy-based security enforcement for SDN-aware IoT networks, Comput. Netw. 213 (2022) 109123.

[30] A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, R. Kompella, Towards an elastic distributed SDN controller, ACM SIGCOMM Comput. Commun. Rev. 43 (4) (2013) 7–12.

[31] A.A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, R. Kompella, Elasticon: An elastic distributed sdn controller, in: Proceedings of the Tenth ACM/IEEE Symposium on Architectures for Networking and Communications Systems, 2014, pp. 17–28.

[32] J. Cui, Q. Lu, H. Zhong, M. Tian, L. Liu, A load-balancing mechanism for distributed SDN control plane using response time, IEEE Trans. Netw. Serv. Manag. 15 (4) (2018) 1197–1206.

[33] M. Pritikin, P. Yee, D. Harkins, Enrollment Over Secure Transport, Tech. rep., 2013.

Claudio Zanasi received the master's degree in computer engineering summa cum laude with a thesis on blockchain based systems from the University of Modena and Reggio Emilia, Italy, in 2016. He later worked as cybersecurity analyst for a major italian bank and as a technology consultant for an italian startup in the finance/insurance space. Currently he is pursuing the Ph.D. degree in the University of Bologna with a research on analysis of cybersecurity practices within green energy production and distribution systems.

Silvio Russo is a Ph.D. student in Computer Science and Engineering at the Department of Computer Science and Engineering, Alma Mater Studiorum University of Bologna, Italy. From the same institution, he received the Master's Degree in Computer Engineering summa cum laude in 2022 with a thesis on Zero Trust applied to the Industrial sector. His research interests are on cyber–physical and industrial cybersecurity, machine and deep learning applications for cybersecurity.

Michele Colajanni is a full professor of computer engineering since 2000, he has been at the University of Bologna since 2021. His research activities include technologies and management of cybersecurity, the design and testing of scalable and resilient systems, even through big data analytics and machine learning methods. Founder of the Cyber Academy for ethical hacker training, and of the Interdepartmental Research Center on Security and Risk Prevention (CRIS) at the University of Modena and Reggio Emilia, he has been collaborating for years with the Bologna Business School where he directs the postgraduate course in Cybersecurity Management. He coordinates various training courses and dissemination initiatives even oriented to overcome the (gender) digital divide in IT. He is involved in multiple research and technology transfer projects at national and international level.