The Role of Security Orchestrator in Network Slicing for Future Networks

Shalitha Wijethilaka, *Student Member, IEEE,* Madhusanka Liyanage, *Senior Member, IEEE*

***Abstract*—The traditional paradigm of connecting mobile devices over the telecommunication networks for voice communication has evolved to a myriad of novel applications with heterogeneous network requirements. The conventional telecommunication networks require a radical change to support these applications. Network Slicing (NS) is one of the utilitarian technologies in future telecommunication networks to address this challenge by dividing the physical network into multiple logical networks with different network characteristics. The evolution in the applications and telecommunication networks intensifies the attention towards the security aspects. Since NS architecture is at its preliminary level, there is no security-specific element in the slicing architecture to perform security-related operations. Hence, we introduce the novel concept of security orchestrator for the NS architecture. This paper extensively discusses the expected advantages and design aspects of such a separate security orchestrator for an NS ecosystem. Moreover, the viability and the benefits of employing the proposed security orchestrator are demonstrated via a testbed implementation and relevant simulations. Finally, a set of potential future research directions related to the security orchestrator are introduced to further improve the proposed architecture's performance.**

***Index Terms*—Network Slicing, Security, Orchestration, Network Architecture**

I. Introduction

Connecting anything from anywhere at any time is an ubiquitous requirement in the journey towards a smart life. The traditional scenario of connecting people and mobile over the network extends to novel services such as autonomous vehicles, remote surgeries, and smart cities. These diverse services require heterogeneous network requirements. Hence, the "One size fits all" concept is not reconcilable in future networks. Deploying a dedicated physical network for each application is not viable to address this challenge. Thus, a radical change in the telecommunication networks is required to accomplish these requirements [1]. The novel Fifth Generation (5G) telecommunication architecture is specifically designed to address this challenge with the support of a myriad of technologies. Network Slicing (NS) is one of the predominant technologies in the 5G architecture.

NS enables a revolutionary transformation in telecommunication networks to facilitate diverse communication requirements of different applications. Dividing the physical network into multiple logical networks with specific network characteristics can be defined as the concept of NS [2]–[4]. NS eliminates the static nature of the network. Improving scalability, security, privacy, and efficiency in resource management are some other benefits provided by NS [5]. Software-Defined Networking (SDN), Network Function Virtualization (NFV), and cloud computing are the dominant technologies that support the NS realization [6]. A particular network slice spans from Radio Access Network (RAN) to core network across the transport network.

Standardization organizations such as International Telecommunication Union (ITU) and Third Generation Partnership Project (3GPP), define the 5G service areas into three main categories: enhanced Mobile BroadBand (eMBB), Ultra-Reliable Low latency Communications (URLCC), and massive Machine Type Communications (mMTC) [7]. NS performs a vital role in the realization of these service areas in 5G and beyond networks. In addition to these primitive service areas, separate network slices can be allocated to specific applications such as industrial automation, military situations, and smart grids. A fully functional network slice can successfully route a packet throughout the network without having disturbances from other slices. In addition to advantages for third parties such as industries and enterprises, NS provides several benefits to Mobile Network Operators (MNOs). NS generates new business concepts such as Network Slice as a Service (NSaaS) which allows operators to provide network slices for vertical industries more agilely [8], to increase the profit of the MNOs. Dynamic resource allocation between slices helps to increase the efficiency of network resource utilization and the scalability of the network [9].

Apart from the advantages that can be achieved, NS brings a novel set of security and privacy issues that need to be addressed. As a result of becoming smart, everything around us exchanges a massive amount of sensitive and critical information across the networks. Hence, security and privacy are major concerns in modern telecommunication networks. Telecommunication networks are vulnerable to several security attacks such as Distributed Denial of Services (DDoS), Man-in-the-Middle (MITM) and zero-day [10]. In addition to these conventional attacks, NS can generate a new set of security threats resulting from new interfaces, components, and capabilities [11]. Therefore, the attack surface of the NS-based system could be more severe than the traditional telecommunication networks. Hence, the management of these complex security requirements in the NS ecosystem is critical.

However, NS deployments are still at their preliminary stage. Hence, the reference architecture has been mainly designed without considering security aspects [12]. Security

Shalitha Wijethilaka is with the School of Computer Science, University College Dublin, Ireland (email: mahadurage.wijethilaka@ucdconnect.ie).

Madhusanka Liyanage is with the School of Computer Science, University College Dublin, Ireland and Centre for Wireless Communications, University of Oulu, Finland (email: madhusanka@ucd.ie).

breaches can occur anywhere in the network. Deploying security Virtual Network Functions (VNFs) or performing security-specific configurations is required to eliminate those challenges. Since security VNFs consume network resources, the engine that performs these operations should be centralized to manage network resources efficiently. Moreover, managing the life-cycle of the security VNFs of different slices is a complicated task. A central element can be helpful in performing security operations in the NS ecosystem independently without being a burden to the existing traditional Network Slice Manager (NSM) which manoeuvre the operations in the network slice life-cycle management process.

Thus, a central element is required to be introduced to the NS architecture to manage security-related operations. This paper presents the novel concept of security orchestrator to the NS architecture. The proposed security orchestrator can offer various security services such as attack detection, proactive and reactive security deployment and security service life-cycle management. The paper also illustrates the high-level abstraction and the modular-level design of the proposed security orchestrator. The feasibility of the proposed architecture is demonstrated by a testbed implementation developed using standard tools. In addition, a set of extensive simulations are executed to show the impact of the security orchestrator.

A comprehensive analysis of security orchestration is presented in section II. Section III discusses the potential security challenges in the NS ecosystem. The architecture of the proposed security orchestrator and key components are proposed in Section IV. Section V presents a detailed discussion on the modules of the proposed security orchestrator. Security services that can be provided by the orchestrator have been discussed in section VI. Section VII provides the feasibility evaluation of the proposed concept. Implementation setup, the evaluation of the test results, and a set of simulations are presented here. Potential future research directions are discussed in section VIII while discussing implementation challenges. Finally, Section IX concludes the paper.

II. Related works - Security Orchestration

Security solutions developed by multiple vendors using heterogeneous technologies and paradigms to prevent cyber-attacks need to work in an integrated fashion to support security operation centres while increasing the efficiency, and effectivity [13]. Security orchestration targets to integrate multivendor security tools to inter-operate. In [13], Islam et al. classified the main functionalities of security orchestration into three principal areas - unification, orchestration, and automation, and also, they identified core components of a security orchestration platform. The requirement of security orchestration in telecommunication networks is highlighted in [14], and they identified four objectives of a security orchestration framework: constantly measure, control and limit access, detecting threats earlier, and rapid response. These objectives were advantageous when proposing our framework to an NS ecosystem.

Security orchestration is a novel paradigm in the NS ecosystem. Therefore, scientific investigations related to security orchestration in an NS ecosystem are rare. However, scientific investigations related to security orchestration of technologies which supports the realization of NS such as SDN and NFV can be found.

In [15], Jaeger et al. extended the ETSI NFV reference architecture by introducing a security orchestrator with required interfaces to inter-work with the reference architecture while defining the security orchestration tasks. A conceptual design framework for NFV based security management and service orchestration is presented in [16] to dynamically and adaptively deploy security functions. They developed an NFV based access control system to illustrate the feasibility of the proposed security orchestration framework. In [17], Pattaranantaku et al. proposed a security extension module to the NFV MANO architecture based on TOSCA data model. An access control use case is used to illustrate the usage of the proposed security extension, and the results exhibited that their security extension can work together with the NFV orchestrator.

A semantic aware, zero-touch and policy-driven security orchestration framework in SDN/NFV aware IoT scenarios for automatic and conflict-less security orchestration is proposed in [18]. The proposed framework ensures the optimal allocation and the Service Function Chaining (SFC)s [19] of the Virtual Security Functions (VSFs). In [20], Hermosilla et al. proposed a novel NFV/SDN based zero-touch security orchestration framework for configuring and deploying VSFs in Multi-Access Edge Computing (MEC)- Unmanned Aerial Vehicles (UAVs). They have implemented, deployed, and evaluated the proposed solution in a real-testbed to verify its feasibility and performance.

Almost all these existing researches consider security orchestration in SDN/NFV environments. However, the concepts, modules, and architectures can be utilised as references when designing a security orchestration framework in an NS ecosystem.

III. Background

The NS ecosystem is fully-fledged with security challenges. Those can be divided into two main categories: challenges originated from the device end and propagated through the network and challenges within the slicing ecosystem itself. The security orchestration framework should be able to handle all these security challenges.

A. Security challenges originated from the device

Several security attacks, such as DDoS, MITM, and botnet, can be originated from the device end. The impact of these attacks may extend to the network level. For instance, in a DDoS attack situation, the whole network may collapse. In [21] Sattar et al. proposed a mathematical model to mitigate DDoS attacks in 5G NS using network isolation. In addition to conventional security threats, the rapid expansion of the Internet of Things (IoT) systems that consist of simple connected devices intensify the vulnerability of telecommunication networks to security attacks. The majority of the IoT devices are resource constraints [22]. Hence, implementing complex

security mechanisms on the device side is complicated. Also, due to the competition in the IoT industry, companies disburse less attention to the security mechanisms in their IoT products. Lightweight security protocols and standards are still developing for the IoT industry. These reasons attract attackers towards IoT. Thus, the integration of IoT devices also increases the attack space of telecommunication networks. Network-level security solutions are required to mitigate or reduce the impacts of IoT security attacks to protect IoT devices as well as telecommunication networks.

Moreover, the security requirements are diverse in different IoT applications. For example, while autonomous vehicles or military applications require high-security requirements, environmental monitoring applications demand primitive security requirements. However, facilitating these diverse security requirements over a single physical network is complex. Hence, allocating separate slices of the network with diverse security configurations and security functions for these individual applications is a viable solution.

B. Security challenges within the slicing ecosystem

In [23], Olimid et al. divide NS related security perspectives into three main categories:

1) **Life-cycle security** which considers security aspects related to different phases of the life-cycle of NS
2) **Intra-slice security** that involves in security aspects within the slice itself
3) **Inter-slice security** that considers security aspects among slices.

Vigorous slice isolation is a primary security requirement in the slicing ecosystem [24]. Since a particular User Equipment (UE) can access more than one slice at once, UE can launch side-channel security attacks on other slices if robust slice isolation mechanisms are not implemented. Though slice isolation is critical, absolute slice isolation can not be achieved due to several reasons. For instance, inter-slice communication is required in some scenarios, such as receiving services from resources starving VNFs in other slices. Hence, slice isolation should be managed appropriately. In [25], Cunha et al. present leading security challenges in the slicing ecosystem, specifically in the packet core. A dynamic, machine-readable, continuous, future-proof, and automatic approach to add cyber-security requirements to future networking paradigms is presented in [26].

Moreover, network resources are required to be shared between multiple slices due to resource scarcity. Therefore, the traffic on a particular slice can be accessible to other slices which have shared network resources or equipment. This originates novel security and privacy challenges in the slicing ecosystem [12]. Application Programming Interfaces (APIs) allow third parties to perform several operations in the NS ecosystem. These APIs will be an entry point for intruders to perform security breaches. In addition, adversaries can impersonate the host platform with the support of these APIs. Impersonation of the NSM allows gaining access to all the slices, resulting in a breach of system confidentiality and integrity [27].

Allocated resources for security functions in network slices should also be managed optimally to increase the resource utilization efficiency. Moreover, third-party tenants (slice owners) such as industries and governments significantly involve in network configuration (specifically slice creation) for their applications [28]. They present their security requirements via Security Service Level Agreements (SSLAs). These security requirements can be significantly dynamic due to the dynamic nature of future applications. Thus, dynamic management of security functions and frequent re-configurations of slice-level security services will be essential requirements beyond the 5G ecosystem. Hence, this is a considerable security management challenge in the NS ecosystem.

C. Motivation scenario

In this paper, we consider a motivation scenario where a centralized security orchestrator provides heterogeneous security requirements of a network slicing ecosystem in a smart hospital environment. Smart hospital can be replaced by any novel application such as smart factory, smart farm with appropriate slice allocations. Figure 1 shows the role of the security orchestrator in the hospital environment.

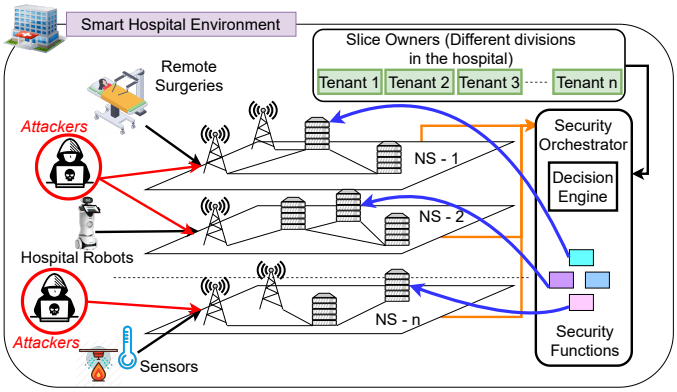

Fig. 1: Role of security orchestrator in an NS environment

Network slices can be allocated to specific applications that have diverse security requirements. Security orchestrator collects monitoring information from network slices and security requirements from slice owners and decides security operations that need to be performed in network slices to accomplish those requirements. Attackers can perform attacks on a specific slice or multiple slices at once. The centralized nature of the security orchestrator supports identifying all these attacks in the NS ecosystem.

IV. Proposed Security Orchestrator Architecture

In this section, we describe the architecture of our security orchestrator. Moreover, the system constraints that need to be satisfied by the proposed system are presented here.

A. Architecture design of the security orchestrator

The initial step in designing the security orchestrator is to identify the optimal location for the security orchestrator in

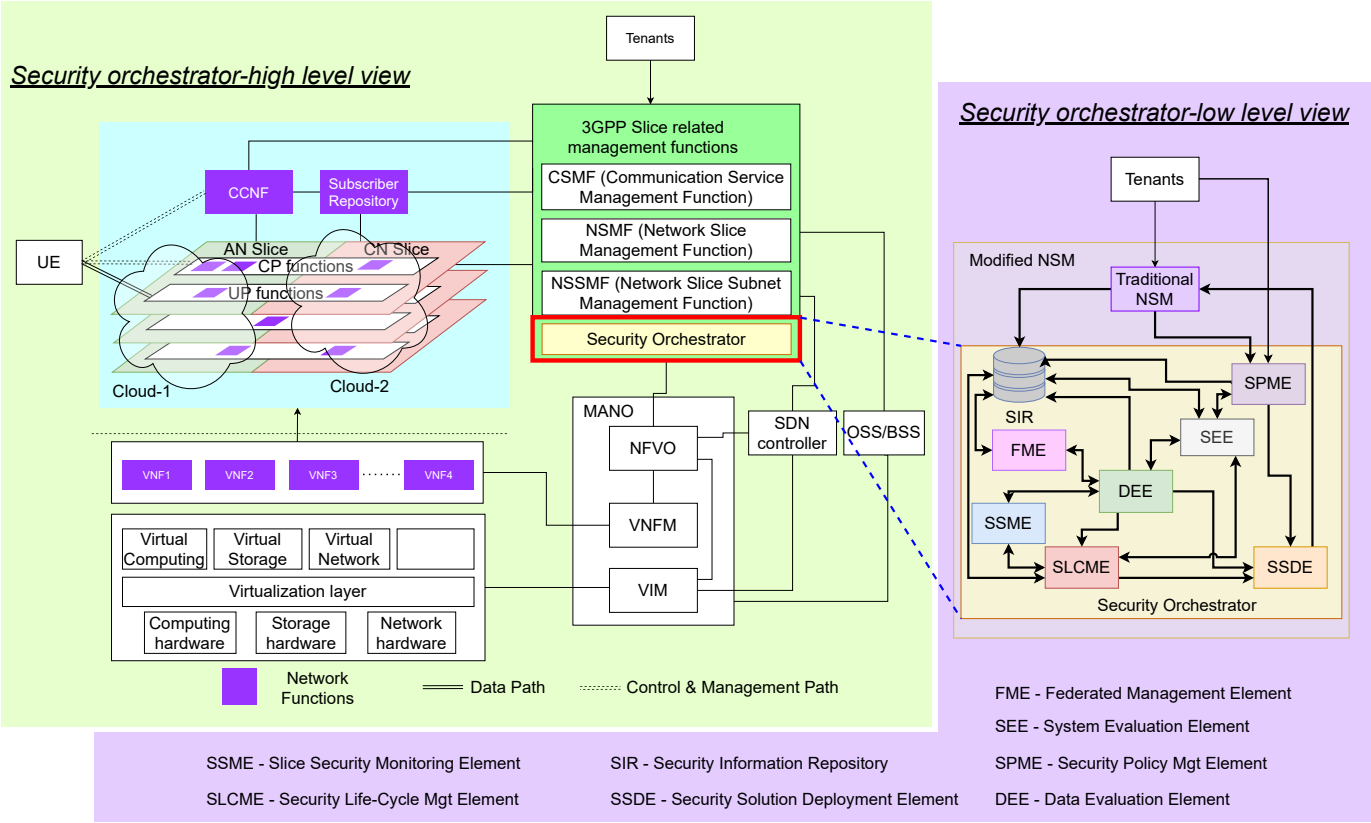

Fig. 2: Proposed Security Orchestrator Architecture

the traditional NS architecture. We considered several factors to determine the location of the security orchestrator. First, it should be able to monitor the whole NS ecosystem to receive security information from the slices. Second, the security orchestrator should be able to perform relevant security service configurations of the slices. Accessing the Network Function Virtualization Management and Orchestration (NFV-MANO) via traditional NSM is required to accomplish this requirement. Moreover, the location should be accessible to third party tenants to present their security requirements. Considering these factors, we propose implementing the security orchestrator as an addition to the traditional NSM. Hence, the novel NSM should consist of Communication Service Management Function (CSMF), Network Slice Management Function (NSMF), Network Slice Subnet Management Function, and the security orchestrator. The left side of the figure 2 depicts the proposed location of the security orchestrator in the conventional NS architecture.

The architecture of the security orchestrator is designed to be modularized to improve manageability. It simplifies the updating process of each module independently without affecting other modules and increases the scalability. It also enables the possibility of adding extra modules to satisfy future specific-security service requirements. The modular level design of the security orchestrator is shown on the right side of figure 2. These modules have to collaborate to realize the security services offered by the proposed security architecture. The proposed security orchestrator consists of eight main modules. Those modules and their primitive functionalities are as follows.

- **Slice Security Monitoring Element (SSME)**: Collect security-related information from network slices and pass them to relevant entities
- **Security Information Repository (SIR)**: Store all the security-related information in the NS ecosystem to support the functionality of the security orchestrator
- **Data Evaluation Element (DEE)**: Perform initial analysis of the collected monitoring information, and decide on initial action to mitigate the identified attacks
- **Security Life-Cycle Management Element (SLCME)**: Manage the life-cycle of performed security operations in the network slices
- **Security Policy Management Element (SPME)**: Accept security policies from tenants and translate them to an understandable manner to the security orchestrator
- **Security Solution Deployment Element (SSDE)**: Interact with the NSM to perform the selected security operations in the NS ecosystem
- **Security Evaluation Element (SEE)**: Evaluate the status of a particular network slice to perform the selected security operations
- **Federated Management Element (FME)**: Maintain the security-related Machine Learning (ML) model repository to support the federated learning process in the NS ecosystem

B. System constraints of the proposed model

Here, we present the considered system model with utilized notations to define the constraints to follow in the security orchestrator design. We considered a slicing ecosystem that consists of K network slices that span over multiple-cloud environments. N denotes a network slice, C denotes a cloud, L denotes the latency, and V denotes a network function. Therefore, the k^{th} VNF in j^{th} cloud in i^{th} network slice can be denoted by the tuple (N_i, C_j, V_k). R denotes resource amount in a particular entity, and it could equal to $\{RAM, CPU, STORAGE\}$. Resource amount in i^{th} network slice can be defined as follows.

$$R_{N_i} = \sum_j \sum_k R_{(N_i, C_j, V_k)} \tag{1}$$

When considering the E2E latency of a particular network slice, it depends on two parameters: service delay and propagation delay [18]. Service delay considers the processing time of the data sent by a source VNF at the destination VNF. It is a function of network resources. We can calculate the service delay of $v \in V$ using $F(v, \delta_v)$. δ_v is the resources used by v. Propagation delay is the time required to transfer one single bit from one VNF to another. We assumed that the propagation delay in two VNFs in the same cloud is negligible. T_{C_{i-1}, C_i} denotes the propagation delay from C_{i-1} to C_i. Then, the E2E latency of a particular slice can be defined as follows.

$$L_{N_i} = \sum_j T_{C_{j-1}, C_j} + \sum_k F(v_k, \delta_{v_k}) \tag{2}$$

Lets consider the i^{th} slice that span over M clouds $\{C_1, C2, ..., C_M\} \in N_i$. When deploying selected security functions in the i^{th} network slice, following constraints should be considered.

The allocated resources for security functions should not exceed the maximum allocated resources for the slice.

$$R_s \leq R_{N_{i,max}} - \sum_j \sum_k R_{(N_i, C_j, V_k)} \tag{3}$$

The added latency due to selected security functions should not exceed the maximum allowable latency in the slice.

$$L_s \leq L_{N_{i,max}} - \sum_i T_{C_{i-1}, C_i} - \sum_k F(v_k, \delta_{v_k}) \tag{4}$$

V. Extended discussion on the modules of the proposed orchestrator

A detailed discussion on the architecture, functional requirements, and operations of each module of the proposed security orchestrator is presented in this section.

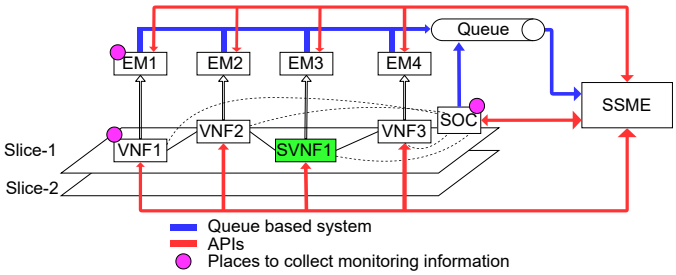

Fig. 3: Different data collection mechanisms of SSME

A. Slice Security Monitoring Element (SSME)

SSME is responsible for performing all the security-related monitoring operations in the slicing ecosystem. Figure 3 shows the data-flows and the operation schemes of the SSME in a NS ecosystem.

As shown in the figure, three potential places can be identified to collect security information.

- SOC (Security Orchestrator Client): A slice specific central element that contact with all the VNFs in the slice to collect monitoring information and send them to SSME
- EMs (Element Managers): Specific to each VNF that take care of the configuration and management of the network functions. Security orchestrator and EMs can intercourse to exchange data
- VNFs: Directly interact with security orchestrator to communicate data and VNF templates need to be modified to communicate with security orchestrator

Using those three places, two methods can be identified for collecting monitoring information from the slices. The first method is collecting information periodically. Here, the period which is utilized to collect information needs to be a slice specific parameter, and it should be configurable. In the second method, slices can present their information to the SSME when needed. The SSME should expose APIs to receive monitoring information, or a Queue based system can be utilized to collect monitoring information. Security outputs from the deployed Security Virtual Network Functions (SVNFs) in the slices, outputs from deployed security related ML models, and performance matrices of the VNFs in the slices, can be identified as the monitoring information that needs to be collected for a particular network slice.

When considering security monitoring, two kinds of monitoring schemes can be identified: Active monitoring, which injects some test traffic into the system and monitors the behaviour, and Passive monitoring, which monitors the live network traffic [29]. Active monitoring helps to Validate the performed security configurations in the slices. Real attack scenarios can be identified by performing passive monitoring. The SSME needs to be able to perform both schemes.

B. Security Information Repository (SIR)

SIR stores all the required information for the functionality of the security orchestrator. Figure 4 shows the primitive Entity Relation (ER) diagram of the SIR. Details of the available security functions with resource requirements that can be

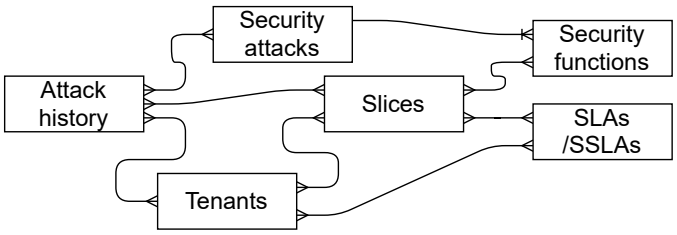

Fig. 4: Basic Entity Relation (ER) diagram of SIR

deployed in the slices, resource limits of the slices according to the SSLAs, potential security attacks and the corresponding security solutions, and history of attacks for slices are some of the information that should be maintained in the SIR.

C. Security Evaluation Element (SEE)

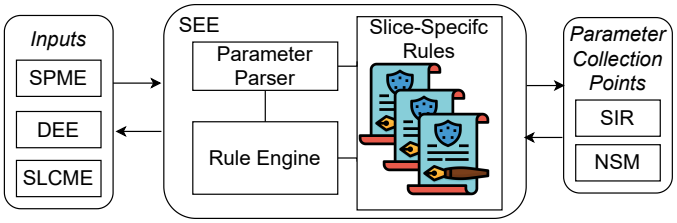

Fig. 5: Functional diagram of SEE

SEE is an essential element for the operation of the entire security orchestrator. Figure 5 shows the functionality of SEE. It mainly consists of a rule engine, parameter parser, and slice-specific rule set. It can access the real-time metrics such as available resource amounts in each cloud for every network slice, and imposed security policy information of network slices. When performing security operations in the system, the compatibility of the particular security operation with network slice-specific parameters should be considered. Therefore, DEE, SLCME, and SPME communicate with SEE to scrutinize selected security operations with slice-specific parameters. It identifies the incompatibilities, contradictions, and dependencies prior to enforcing security operations in the system. Slice-specific rule-set needs to be defined to avoid these. When SEE receives an input, it modernizes the slice-specific rule set with real-time parameters from parameter collection points, evaluates rules with rule-engine and provides the output.

D. Data Evaluation Element (DEE)

DEE acts as the brain of our proposed security orchestrator. After receiving the monitoring information across the SSME, DEE performs further analysis according to the type of received information. The functional flow diagram of the DEE is shown in figure 6.

After receiving performance matrices related to a particular slice, the DEE runs ML algorithms such as logistic regression, decision trees, deep neural networks, and reinforcement techniques to identify anomalies in the received matrices. If it identifies an anomaly, it examines potential security solutions

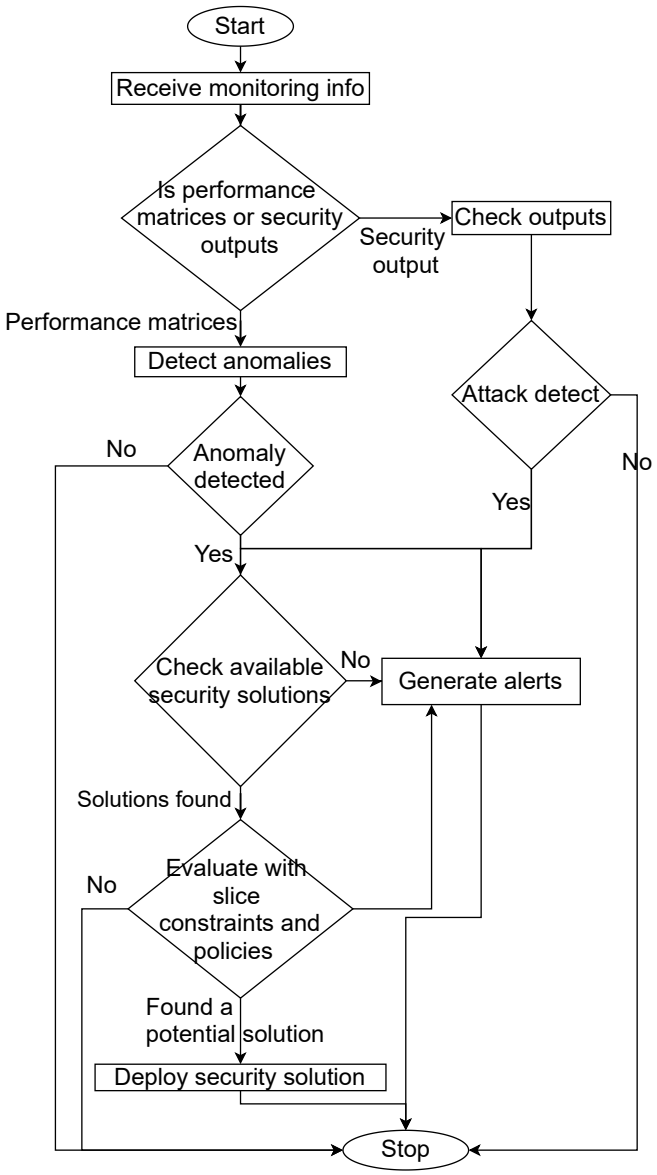

Fig. 6: Functional flow diagram of DEE

from existing security solutions and evaluates them with SEE. Then the security solution will be configured in the slice while educating the relevant parties. If the DEE can not identify a solution from existing solutions, it informs the security operations team immediately to perform necessary actions.

When receiving security outputs from the deployed SVNFs in slices, the DEE examines the outputs for identifying security attacks. If the DEE can identify security attacks from security outputs, it examines potential security solutions from the existing solutions in SIR and evaluates the selected solutions with slice-specific rules with SEE. At the same time, it notifies relevant parties about the attack. Finally, the DEE handovers the attack handling process to the SLCME.

E. Security Life-Cycle Management Element (SLCME)

Figure 7 shows the functional flow diagram of the SLCME. The responsibility of the SLCME is to manage the life-cycle

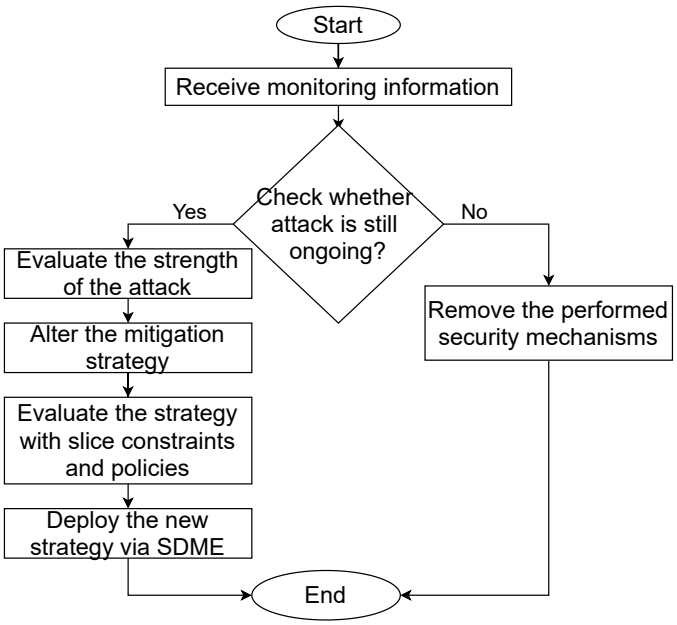

Fig. 7: Functional flow diagram of SLCME

of the security VNFs. In terms of performance improvement, the life-cycle management of SVNFs is an essential operation. SLCME is responsible for initialization, maintenance and termination of security VNFs in each network slice. According to the severity of the attack and the phase of the attacks, security VNFs need to be modified. Thus, the SLCME can dynamically update the security solution according to the different phases of the attack.

After detecting and performing initial action for a particular attack on a slice by the DEE, it instructs the SLCME to update the security VNFs according to a defined mitigation plan. After the initial deployment of SVNFs, the SLCME continuously monitors the information from the particular slice to identify the alterations in the security attack. If the strength of the particular attack is becoming severe, the deployed security configurations need to be updated. Finally, the deployed security configurations need to be restored to turn the particular slice into the normal state at the end of a particular attack.

F. Security Solution Deployment Element (SSDE)

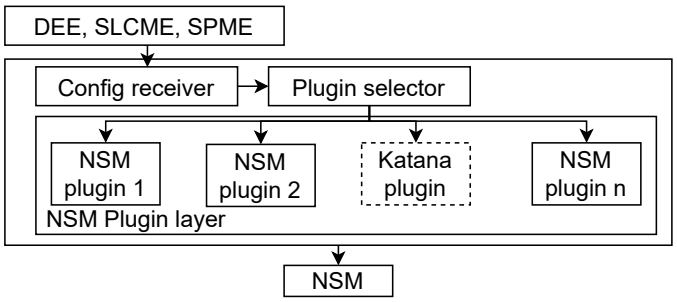

Fig. 8: Architectural diagram of SSDE

SSDE is the module that communicates with the NSM to perform alterations in the network. Primarily SSDE communicates with NSM to perform following actions.

- Deployment of new security VNFs in network slices
- Adjusting the resource allocation to security VNFs
- Altering the security related configurations in VNFs

A plugin-based architecture can be developed for the SSDE as shown in figure 8 to increase the support for multiple NSMs by different vendors. When there is a new NSM, a new plugin needs to be developed and deployed in the SSDE. After getting inputs from the SPME, DEE, and SLCME, SSDE determines the appropriate NSM plugin, which corresponds to the NSM in the slicing ecosystem, and sends the request to the plugin. Then the plugin parses these requests in an understandable manner to the NSM and sends them.

G. Security Policy Management Element (SPME)

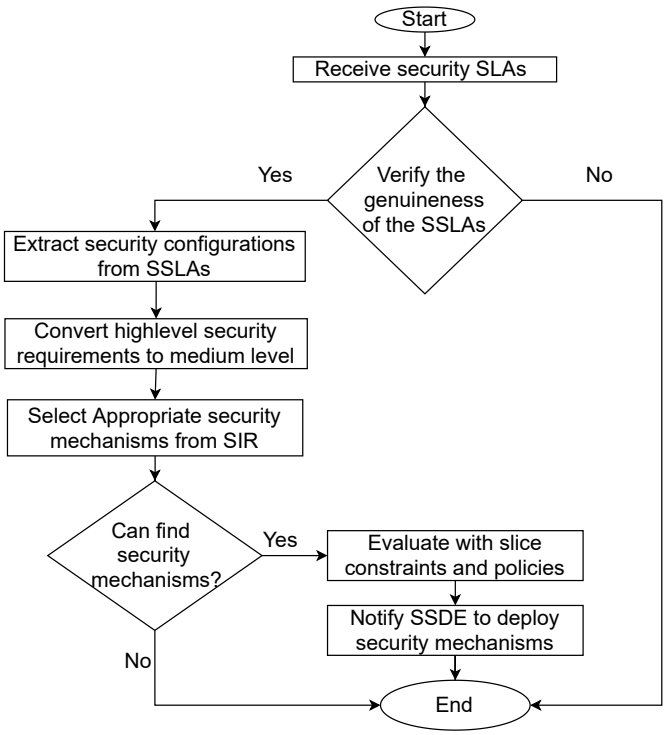

Fig. 9: Functional flow diagram of SPME

Figure 9 shows the functional flow diagram of the SPME. It is responsible for collecting the security requirements of third party tenants via SSLAs. The SPME converts the security requirements to network-level configurations, evaluates them with SEE, and notifies the SSDE to perform the configurations. As the security requirements are presented to the SPME through the trusted entities in the network, reviewing authentication and authorization is not required. Security requirements can be presented inside of Network Service Descriptors (NSDs) using standard NSD languages such as TOSCA [30].

A security policy priority scheme can be implemented to simplify the evaluation process of SEE. Priority values can be assigned to security policies to evaluate the policies with existing policies. Policies with higher priority values should be

deployed or configured earlier when there are difficulties such as resource limitations when performing security operations.

H. Federated Management Element (FME)

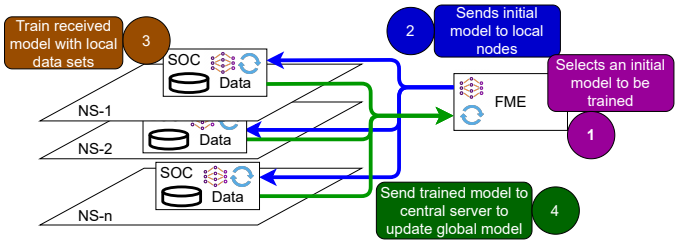

Fig. 10: Functional diagram of FME

The FME is responsible for federated training of security-related ML models in the network slices. Figure 10 shows the functionality of the FME with the support of SOC. Due to the requirement of network slice isolation and privacy requirements of network slices, collecting data in a centralized location to train ML models is not viable in an NS ecosystem. Therefore, the novel ML approach, known as federated learning, is required to train ML models in an NS-enabled environment. In our proposed security orchestrator, we allocated a dedicated module to perform these operations in the NS environment. The FME selects initial models and distributes them across network slices with the support of SOC. After that, the received models are trained using the slice-specific data, and model parameters are sent to the FME again. The FME aggregates the received model parameters and updates the central models. Finally, the updated central ML models can be deployed in network slices to perform security-related predictions.

VI. Services Offered by the Proposed Security Orchestrator

This section describes a set of potential services that can be facilitated by the proposed security orchestrator. Different combinations of the modules in the orchestrator are required to realize those services. Furthermore, more security services can be enabled using some other combinations of different modules, than the mentioned services.

A. Deployment Reactive Security Services

Reactive security mechanisms focus on responding to security incidents after they occur, for instance, hacks and data breaches. The security orchestrator can be utilized to deploy security services in a reactive manner to address the security challenges in the slicing system. The flow diagram of how the security orchestrator can be utilized to facilitate reactive security mechanisms is shown in figure 11. Mainly, SSME, DEE, SLCME, SEE, and SSDE participate in providing reactive security mechanisms. After collecting the security-related information by the SSME, the DEE analyses the collected data to identify ongoing attacks based on the information on the SIR or outputs from ML algorithms to identify anomalies

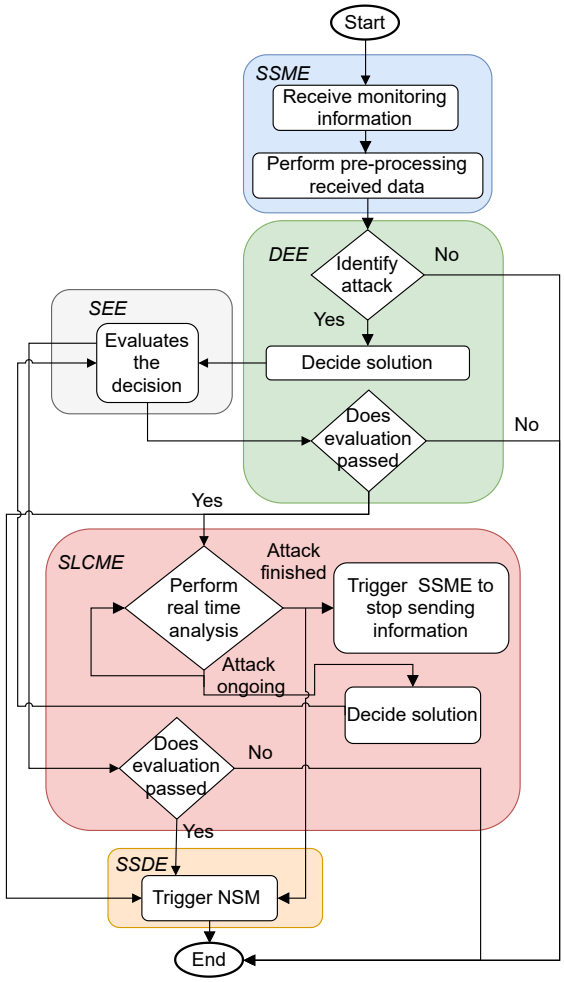

Fig. 11: Reactive Security flow diagram

in the normal behaviour. If a particular attack is identified successfully, the DEE decides a procedure to mitigate the attack, evaluates with SEE, and informs SSDE to perform the relevant operations in the slice. Moreover, the SSME and SLCME are notified to perform a continuous operation to manage the life cycle of the deployed mitigation mechanism.

B. Deployment of Proactive Security Services

Proactive security means implementing security mechanisms prior to security incidents occurring. Those can be implemented in the slicing ecosystem by using the security orchestrator. The flow diagram of implementing proactive security is shown in figure 12. Predominantly, implementing proactive security mechanisms in the NS ecosystem can be performed in two ways with the support of the security orchestrator. The first method is collecting the security requirements by the SPME module from tenants via SSLAs (Security Service Level Agreements). Then, the SPME decides the security functions and configurations required to establish those security requirements. After that, it evaluates them with SEE and notifies the SSDE to trigger the NSM to perform the required configuration in the slices. In the second method, after successively identifying security attacks in adjacent slices,

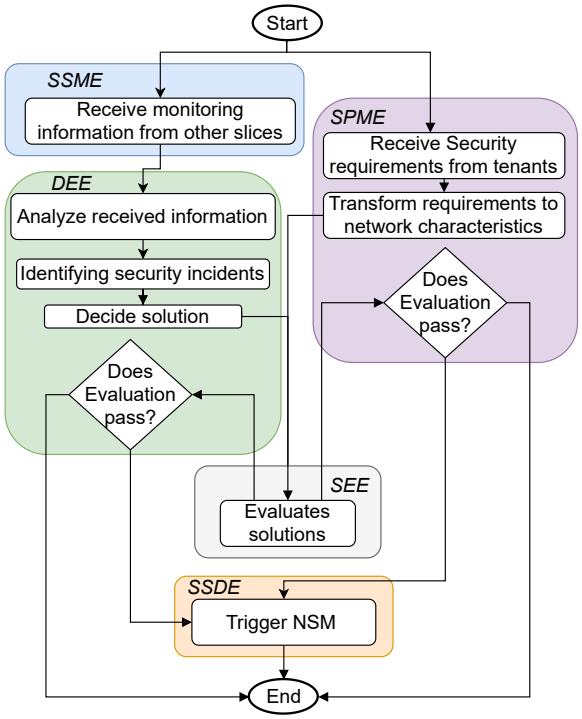

Fig. 12: Proactive Security flow diagram

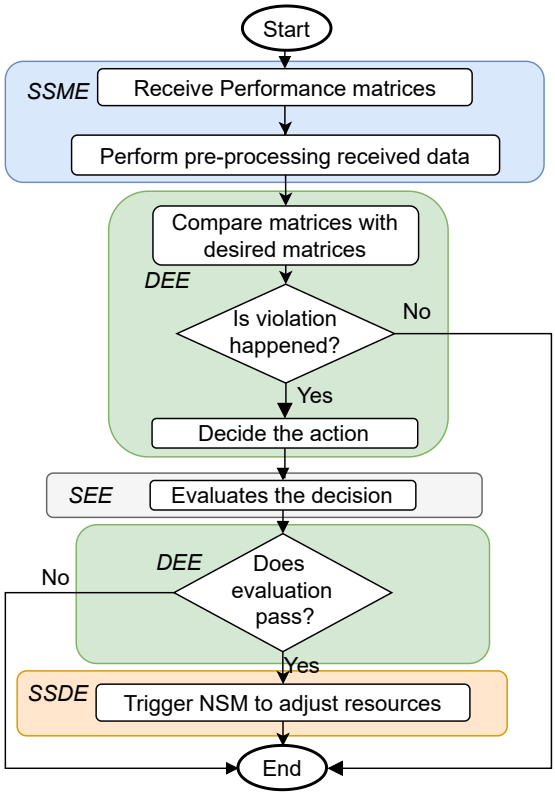

Fig. 13: Security resource management flow diagram

correspondent security countermeasures can be applied to other slices as well, as there is a potential to execute the same security attack after some time by the attackers to other slices. SSME, DEE, SEE, and SSDE engage in the second method.

C. Security Resource Management

Network resources are scarce. Hence optimal management of the available resources is vital in future networks. Allocating a dedicated amount of network resources for security operations is inefficient. Thus, resource allocation for security operations needs to be dynamically adjusted to improve resource utilization efficiency. The proposed orchestrator can adjust network resources allocated for security operations dynamically.

Figure 13 shows the flow diagram of the dynamic security resource management process of the orchestrator. The SSME can continuously monitor the performance matrices of the VNFs in the slices. The DEE uses the monitored information for identifying the violations of the desired performance matrices of a particular VNF. If it identifies a violation, it notifies the SSDE to correspondingly alter the resource allocation for the VNF through the NSM. In this way, the security orchestrator can dynamically adapt the resource allocation for security operations while finally optimizing the overall resource utilization efficiency of the slicing ecosystem.

D. Increased performance of security related ML models

In the traditional NS ecosystem, data sharing between network slices or collecting slice-specific data to a centralized location to perform ML operations is not possible due to the security and privacy requirements of slice data. However, our framework allows utilizing a novel ML technique known as federated learning to optimize the training process of security-related ML models while preserving the security and privacy of slice-specific data. The FME and the SOC deployed in each network slice, involve implementing federated learning for security operations in the NS ecosystem.

E. Life-cycle Management of the Security VNFs (SVNFs)

Life-cycle management of the Security VNFs (SVNFs) is a complex activity. Therefore, one of the primary tasks of the security orchestrator is to manage the life-cycle of SVNFs optimally. The following phases will be implemented under the life-cycle management service of security VNFs.

- **Acceptance:** Here, security requirements will be accepted from the external parties and identify the required security functions and the network configurations
- **Installation:** Identified security functions will be installed in the virtual machines or containers in this phase.
- **Deployment:** VNFs which consists of security functions will be deployed in the relevant slices in this phase.
- **Evaluation:** Continuous monitoring of the security VNF will be performed in this phase.
- **Maintenance:** Security functions will be continuously updated according to the monitored information in this phase. Evaluation phase and this phase will run simultaneously.
- **Disposal:** This is the last phase of the SVNF's life cycle. SVNF will be discontinued here and allocated resources for the SVNF will be released.

Most SVNFs deployments are only performed during the attack situations unless the slice owner has a specific requirement. Moreover, SLCME monitors the different stages of the ongoing attacks and manages the particular SVNFs accordingly.

F. Blacklist the Malicious Users

Quarantining malicious or infected devices is a vital requirement to reduce the effect of the attack on other devices or resources. The proposed security orchestrator can be utilized to perform this task. The security orchestrator can identify the malicious devices by analyzing the received monitoring information. Then, until performing the mitigation actions, those malicious devices can be isolated into a quarantine slice to reduce the impact of an ongoing attack. A rating system or a reputation system can be used to score users and to classify malicious users.

G. Inter-slice Communication Security

Secure inter-slice communication is another advantage that the security orchestrator can accomplish. The proposed security orchestrator can act as a certificate authority (a trusted entity) in the NS ecosystem. It can issue certificates to each slice when creating the slice. Then, they can use those certificates to ensure secure communication in the communications between slices. As different security requirements can be found in different network slices, security levels can be assigned to network slices. When a particular slice needs to communicate with some other slice, security levels of the correspondent slices should be considered. The proposed security orchestrator can perform these security level management of the slices.

H. Centralized security management

NS is an End-to-End (E2E) technology. Therefore, network slices can be spanned from the access network to the core network. Multiple parties engage in performing operations of network slices within these multiple domains. The impact of a particular security attack affects different domains differently. Hence, security teams corresponding to different domains apply heterogeneous security mechanisms to mitigate the same security attack. Even though security teams exchange information about security incidents, miscommunications can be occurred. It is an inefficient operation in terms of resources as well as effort. A centralized system for security management can be helpful to overcome this challenge. The proposed security orchestrator can perform as the required central entity in the NS ecosystem. The proposed orchestrator can monitor E2E network slices and perform necessary actions to mitigate security attacks in each domain.

VII. Feasibility Evaluation

In this section, we present a prototype of the proposed framework to discuss the feasibility of the solution. A couple of use cases of the orchestrator and a set of extensive simulations will be analysed here.

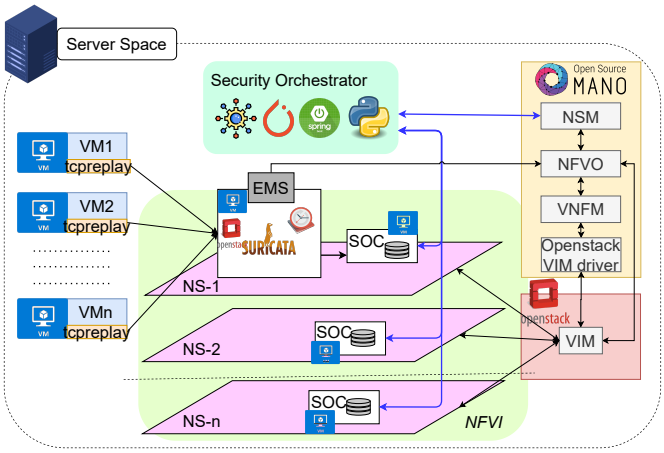

Fig. 14: Implementation setup

A. Implementation setup

Figure 14 demonstrates the implementation setup of the proposed security orchestrator with SOCs for an NS ecosystem. NS testbed implementation in [31] is used as the base of our implementation. The security orchestrator is implemented to follow the micro-service architecture using java spring boot, and python. Open Source MANO (OSM) [32], which is an open-source implementation of European Telecommunications Standards Institute (ETSI) NFV Management And Orchestration (MANO) framework specifications, is used as the NSM and the NFV Orchestrator (NFVO) in our implementation. NFVO is installed with OpenStack VIM(Virtual Infrastructure Manager) driver to communicate with OpenStack installation via APIs. Openstack [33] is an open-source cloud operating system that permits the management of enormous pools of computing, storage, and networking resources in a data centre. PyTorch, and scikit-learn are used to perform the federated learning implementation in the experiments.

NSL-KDD intrusion detection dataset [34] is used for the experiments performed to evaluate the functionality. We consider high-level attack classification, i.e. attack, normal, and mid-level classification, i.e. normal, DoS, User to Root (U2R), Root to User (R2U), and probe, to perform experiments. We ignore the low-level attack classification (more than 20 attack types) in the dataset as each attack type has a very less number of records. A deep neural network with two hidden layers which has 200 nodes in each hidden layer, is used as the model in the experiments. Moreover, Suricata [35] is the threat detection engine which is considered in the experiments.

B. Experiments

1) Increased attack detection

In this experiment, we investigate the impact of the training data distribution across the NS ecosystem on the accuracy of the security-related ML models for attack detection and how our framework can improve the performance. Here, the training data set is Independently and Identically distributed (IID) across the network slices. We compare the centralized security orchestrator-based approach and the legacy approach in this experiment. With our framework, a centralized attack

detection model can be trained using all the security data in the NS ecosystem. In the legacy approach, as the data cannot be shared between network slices, attack detection models are trained only using the slice-specific security data. When the number of network slices increases, the traffic per slice decreases since the number of users in the network is the same. We calculate the accuracy of the models that were trained under those two approaches. The experiment is performed several times for each number of network slices in the system and taken the average to increase the accuracy of the received results.

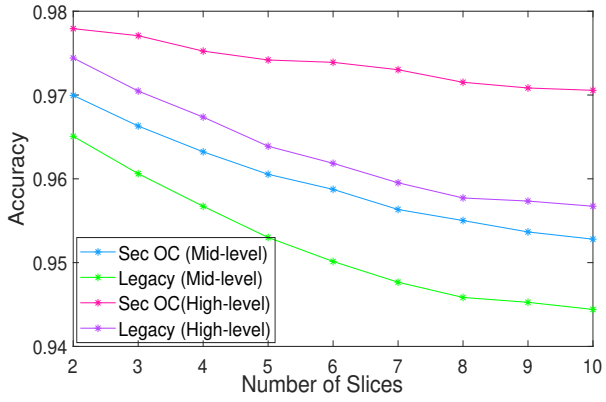

Fig. 15: Accuracy comparison number of network slices

Fig. 15 shows the received results for each model. In terms of accuracy, in both attack classification scenarios, our framework shows a higher accuracy when the number of slices in the network increases. High-level attack classification always shows a higher accuracy due to the lower number of classifiers in the data set. Moreover, when the number of network slices increases, the accuracy decreases in each scenario. In the legacy approach, the accuracy is reduced due to the deficit of the training data in the network slices. We use the FedAvg aggregation mechanism to calculate the model weights of the central model using the models in each node. The deviation from the optimal weight values of the ML models due to the FedAvg aggregation method is the cause of the accuracy reduction of our framework. This experiment shows the increased performance of security-related ML models with our proposed framework.

2) Ability to deploy Proactive security mechanisms

This experiment aims to investigate how proactive security mechanisms can be implemented using our framework, hence making the network slices capable of identifying unseen attacks. Also, we show how our framework can improve the performance of security-related ML models in this experiment. An NS ecosystem with five network slices is considered for this experiment since the considered data set has five label types under the mid-level attack classification. Attack data distribution across the NS ecosystem is shown in table I. As shown in the table I, we considered two cases in this experiment. In case 1, a specific attack type is allocated to each slice, and in case 2, an amalgam of attacks is allocated to each slice. However, the test set consists of all attack types in both cases.

TABLE I: Data distribution across slices in experiment C

	Normal		DoS		Probe		R2U		U2R	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
S1	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
S2	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
S3	✓	✓	✗	✓	✓	✓	✗	✗	✗	✗
S4	✓	✓	✗	✓	✗	✓	✓	✓	✗	✗
S5	✓	✓	✗	✓	✗	✓	✗	✓	✓	✓

C1 Case 1 C2 Case 2

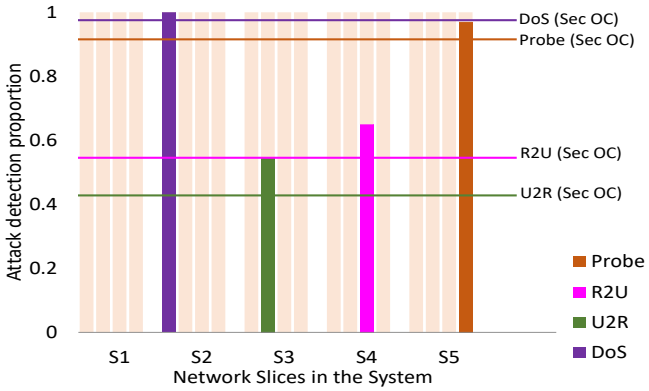

Fig. 16: Attack detection in Case 1

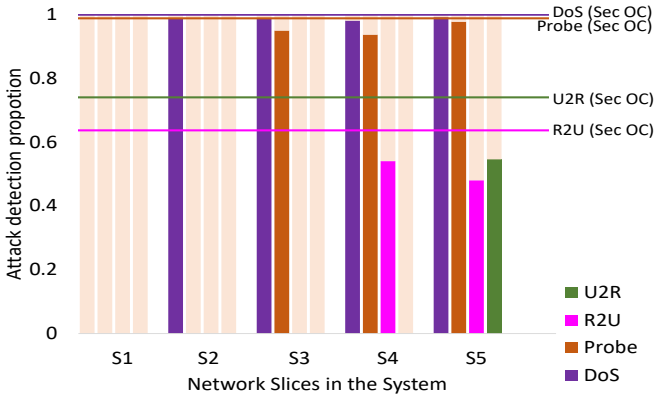

Fig. 17: Attack detection in Case 2

Fig. 16 and Fig. 17 , show the received results in this experiment. In case 1 and case 2, only slice-specific attacks can be identified in the legacy approach. However, with the centralized security orchestrator, all the slices can identify all the types of attacks and security mechanisms can be deployed in advance to protect the slices from unseen attacks. The data distribution across the slices is heavily affected forthe Attack Detection Proportions (ADPs). The ADPs in case 1 are very low than the ADPs in case 2 due to the higher data distribution across the ecosystem in case 2. The increased number of required training rounds (federated rounds) with the proposed framework to achieve a significant accuracy is a drawback in our framework and it can be mitigated to some extent through optimizing the model aggregation process in the FME.

3) Network resource management for security operations

Here, we present another usecase of the security orchestrator: network resource management for security operations. For this use case, Suricata, which is an open-source threat detection engine, is used as the security VNF in the slice. Tcpreplay [36] is executed in multiple VMs to replay pre-recorded pcap files to send at different data rates to Suricata VNF. VMs resource consumption, such as RAM, CPU, and the Dropped Packet Percentage (DPP), is periodically transmitted to the security orchestrator. Security orchestrator monitors the received information and dynamically alters the resource allocation of the Suricata VM with the support of NSM.

Initially, we allocated 4GB ram and two CPU cores to the Suricata VNF. The threshold level is set as 10% of the packet drop rate for this experiment. Once the threshold level is exceeded, the security orchestrator performs alterations in the resource allocation of the Suricata VNF to preserve the DPP below the threshold level. We gradually increase the data rate and measure the performance of the proposed security orchestrator. Figure 18 shows the DPP in the Suricata VNF along with different data rates.

The DPP increases when we increase the data rate. For the first time, the DPP passes the threshold level at the data rate of 54 Mbps. The security orchestrator decides to increase the amount of allocated RAM and the number of CPU cores for the VNF. Then, the security orchestrator alerts the NSM to increase the RAM of the Suricata VNF from 4GB to 6GB and the number of cores to four via Openbaton NFVO. As a result, the DPP of the Suricata VNF drops down to below the threshold level. This experiment is continued with gradually increasing data rates for the Suricata VNF. Within the experiment, the DPP passed the threshold level at three different data rates, and at each point, the security orchestrator decided to increase the resource allocation of the security VNF. Three saw teeth in the graph prove the operation of the security orchestrator. However, in the traditional scenario, DPP is continuously increasing with the data rate. Hence, we can conclude the operation of the security orchestrator in the mentioned use case. The security orchestrator allows maintaining the DPP within the desired range and increasing the attack detection while optimizing the network resource utilization.

C. Simulations

In this section, we evaluate the impact of security orchestrator in different scenarios via simulations. Matlab and python have been used to perform simulations.

1) The impact of different threshold levels selected by the security orchestrator

This simulation shows the impact of the DPP and the resource utilization of the security function for selecting the threshold level in the security orchestrator. In the experiment, we considered a static threshold level to allocate network resources dynamically to the security VNF. However, the impact of this threshold level will affect the performance of the system. Therefore, we evaluate the average DPP at a particular threshold level and the average resource requirement for maintaining that threshold level of the considered security

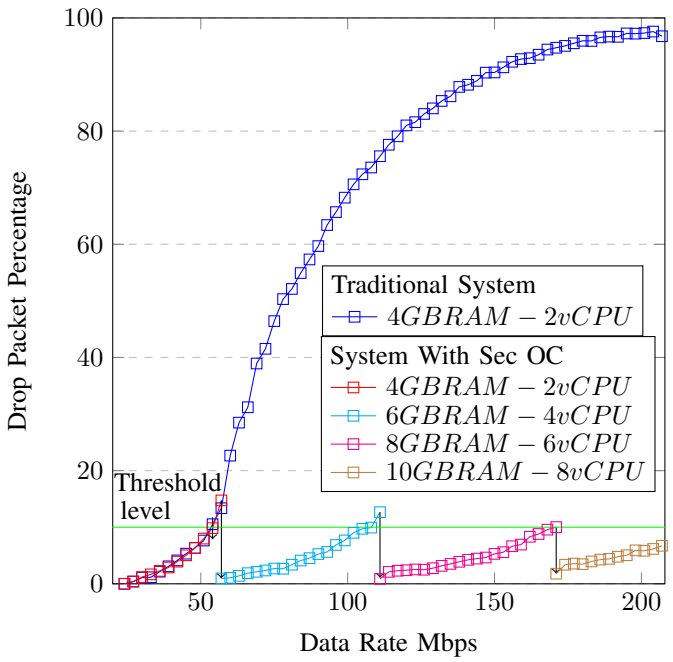

Fig. 18: Experiment Results

VNF. The data received in the experiment has been utilized to perform the simulation. Here, we generated sawtooth graphs as in the experiment for different threshold levels. After, we calculate the average DPP and the average resource requirement of the security VNF at considered threshold levels. We assume that the input data rate to the security VNF is increased by 1mbps per minute. The following equations are used for calculations, and the received results are shown in figure 19.

$$Average\ \ DPP = \frac{Total\ dropped\ packets}{Total\ simulated\ time} \tag{5}$$

$$Average\ resources = \frac{Total\ used\ resources}{Total\ simulated\ time} \tag{6}$$

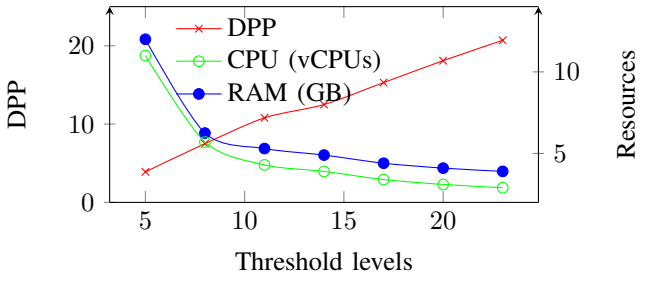

Fig. 19: Impact of different threshold levels for DPP and resource utilization

When the threshold level is very low, the average DPP is very low, but the average resource requirement is high. When the threshold level increases, the average DPP is also increasing. However, the average resource requirement is decreasing. The average DPP and resource requirement should be carefully selected when selecting the threshold level. The threshold level at the cross point of the DPP graph and RAM graph or DPP

graph and CPU graph, can be selected as the optimal threshold level for maintaining optimal resource utilization and DPP.

2) Efficient attack detection by the security orchestrator

Intrusion Detection (ID) is a critical requirement in a network environment. In [37], Lazka et al. discuss the way of selecting optimal threshold levels for Intrusion Detection Systems (IDSs). If very low threshold levels (sensitivity) are utilized, excessive losses can be experienced due to undetected losses. If high threshold levels are utilized, resource wastages can be experienced due to investigating increased false alarms. Therefore, selecting an optimal threshold level for detecting intrusions is essential. Here, we perform a simulation to show the significance of the security orchestrator in selecting the threshold level for ID. We assume that the attacker sends attack traffic 50packets/second per slice. In the traditional security system, slices are monitored individually. However, with the central security orchestrator, all network slices can be monitored at once. The received results are shown in the figure 20.

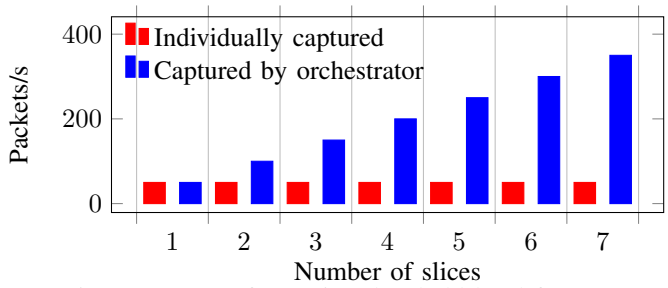

Fig. 20: Ranges for setting threshold level for ID

As shown in the figure, in the traditional scenario, when we perform ID in an individual slice, the threshold level is required to be set between 0-50. However, when we perform ID through the security orchestrator, we get an increased range to select the threshold. When the number of slices in the system increases, the range is also increasing. Moreover, since a higher amount of attacker traffic can be captured at a particular moment, the attack detection in the system is also becoming efficient. Therefore, the security orchestrator supports efficient attack detection as well as maintaining a proper threshold level for ID.

3) Cost for security operations

This experiment shows how the cost for security operations can be reduced through the proposed framework. We compare three scenarios here: traditional scenario (without security orchestrator), with security orchestrator, security orchestrator with SOC. In the traditional scenario, security functions need to be deployed in every slice all the time. Thus, security functions and monitoring functions have to be deployed. However, security functions can be dynamically deployed in the slices according to the situation with the orchestrator. Therefore, only slice monitoring functions are required to be deployed. In the third scenario, besides the monitoring functions, an element to collect matrices from slices needs to be deployed in each slice. We assume that 1-3 network functions need to be deployed for monitoring a slice, and 2-6 security functions are required to perform security operations in a particular slice. Moreover, we assume that the cost is the same for a security function and an element of the security orchestrator. Received results are shown in figure 21.

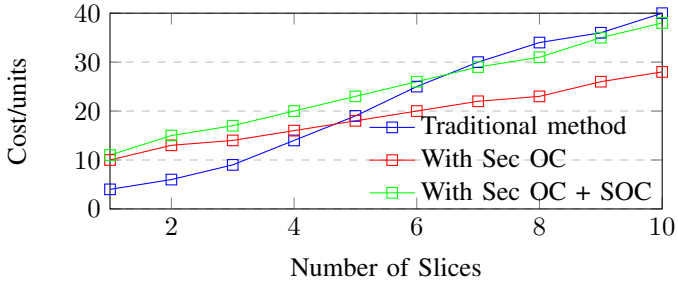

Fig. 21: Cost analysis in NS ecosystem

When the number of slices in the system is low, the cost is high for deploying the security orchestrator. When we use SOC, the cost is maximum. However, when the number of slices in the system increases, the cost becomes a lower value as the deployment of security functions is controlled by the security orchestrator. Therefore, we can conclude that when the number of network slices increases in the system, the cost for security operations becomes a lower value when the security orchestrator performs the security operations even though a SOC has been deployed in each slice.

VIII. Discussion

We compare a set of key features in our framework with existing related works here. Also, we present challenges related to the proposed security orchestrator in this section.

A. Comparison with related works

Table II shows a comparison of our security orchestrator with a set of selected key-related works. The related works are selected considering factors such as specificity for security orchestration, implementation, and key features of the frameworks.

TABLE II: Comparison with key related works

Features	[15]	[16]	[18]	[20]	Our
Modularized implementation	✗	✓	✓	✓	✓
Increased performance of security related ML models	✗	✗	✓	✗	✓
Alignment to ETSI MANO	✓	✓	✗	✗	✓
Pro-active security deployment	✗	✗	✓	✗	✓
NS-specific implementation	✗	✗	✗	✗	✓

From the comparison here, we can deliberate that our proposed security orchestrator performs well in an NS-enabled environment.

B. Design Challenges related to modules of proposed security orchestrator

Different elements in the security orchestrator have individual challenges that need to be investigated. As the NS ecosystem generates several kinds of data, identifying the types of data that are required to be collected by the SSME is a challenge. Moreover, developing standard interfaces to

collect data is another significant challenge. An Optimized ER diagram for SIR needs to be developed considering diverse aspects in the NS ecosystem. Collecting real-time telemetry data of network slices for evaluating rules in SEE needs to be investigated further. The DEE requires novel security attack detection and mitigation mechanisms for different kinds of security attacks. Artificial Intelligence/ Machine Learning (AI/ML) is a major area that can be utilized to develop such mechanisms. Identifying different phases of security attacks is a considerable challenge related to SLCME. Human and machine-readable security schemes and measurable security are exhilarating research areas related to SPME. The NSMs are required to be updated to work collaboratively with SSDE.

C. General Challenges

1) Extra Latency

Security functions are statically deployed in the slices in the conventional scenario. However, security functions can be dynamically deployed in the slices according to the attack situation after monitoring with the proposed solution. Hence, this procedure adds an additional latency for security operations in the slicing ecosystem. High-performance hardware for modules and high-speed links between modules can be used to reduce the time taken to determine the security solution and transfer the data within the NS ecosystem.

2) Scalability Issues

The number of slices in the network increases due to the increased application space such as metaverse, autonomous vehicles, and smart cities. Thus, the amount of information that needs to be handled by the security orchestrator increases. Moreover, due to the rapid expansion of IoT in almost all applications, simultaneous attacks can happen in different slices. Hence, scalability issues can arise in the security orchestrator. SSME, DEE, SEE, and SLCME are the most resource-consuming elements in the security orchestrator. A sufficient amount of resources can be allocated to these elements in advance to manage scalability issues. Furthermore, modular design on the security orchestrator can be utilized to scale the elements horizontally to handle scalability issues.

3) Automation

Automating the functionality of the security orchestrator is a mandatory requirement to increase the efficiency of the system. Attack detection, deciding the mitigation strategies, and LCM of security functions are required to be automated. Network automation is the goal of the ETSI's Zero-touch Service and Network Management (ZSM) [38]. Thus the security orchestrator can be optimized to be aligned with the ZSM. AI/ML techniques can play a significant role in automation.

4) Algorithms development for re-active and pro-active security mechanisms

Security orchestrator is responsible for facilitating re-active and pro-active security mechanisms. However, several algorithms, such as selecting optimal security mechanisms, extracting security requirements, and identifying attack life-cycles, are required to be developed to support these operations. Future research can be executed to identify and develop these required algorithms.

5) Multi-domain security management

Typically, network slices span over multiple domains. Hence security management needs to be performed over multiple domains. The current implementation of the security orchestrator can only manage the security aspects in a single domain. It should be upgraded to facilitate security services over multiple administrative domains. Inter-orchestrator communication schemes need to be developed to enable communication across multiple domains. Moreover, the proposed security architecture can be enhanced to operate in a hierarchical manner.

6) Standardization

Security orchestrator is not a standard component in the NS architecture. However, it is an essential component. Standards Developing Organizations (SDOs) need to consider including security orchestrator in the reference NS architecture. Moreover, standard interfaces are required to be designed for operations such as collecting monitoring information and communication between NSM and SSDE to perform security configurations.

IX. Conclusion

Since security is a pivotal element in modern NS enabled telecommunication networks, this paper presents a comprehensive analysis of how the concept of security orchestration can be utilized in the NS ecosystem. A modular-level architecture of the security orchestrator is provided. A descriptive analysis of the advantages of the security orchestrator and how those can be realized using the orchestrator are presented. Reactive and proactive security mechanisms, LCM of security functions, secure inter-slice communication, and isolation of infected devices are the identified advantages of the orchestrator. A testbed developed using standard tools, is utilized to analyze the feasibility of the proposed security orchestrator. A set of simulations are executed to investigate the impact of the security orchestrator. This feasibility evaluation illustrates the significance of the proposed framework. Moreover, the comparison with existing works shows that the security orchestrator outperforms compared existing work. The discussed implementation challenges open potential future research directions. In all essence, the paper proves the requirement of a security orchestrator for the NS ecosystem.

References

[1] F. Z. Yousaf, M. Gramaglia, V. Friderikos, B. Gajic, D. von Hugo, B. Sayadi, V. Sciancalepore, and M. R. Crippa, "Network slicing with flexible mobility and QoS/QoE support for 5G Networks," in 2017 IEEE International Conference on Communications Workshops. IEEE, 2017, pp. 1195–1201.

[2] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega et al., "Network slicing to enable scalability and flexibility in 5g mobile networks," IEEE Communications magazine, vol. 55, no. 5, pp. 72–79, 2017.

[3] S. Zhang, "An overview of network slicing for 5g," IEEE Wireless Communications, vol. 26, no. 3, pp. 111–117, 2019.

[4] L. U. Khan, I. Yaqoob, N. H. Tran, Z. Han, and C. S. Hong, "Network slicing: Recent advances, taxonomy, requirements, and open research challenges," IEEE Access, vol. 8, pp. 36 009–36 028, 2020.

[5] S. Wijethilaka and M. Liyanage, "Realizing internet of things with network slicing: Opportunities and challenges," 2021.

[6] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2429–2453, 2018.

[7] (2015, sep) Imt vision - framework and overall objectives of the future development of imt for 2020 and beyond. Accessed on 20.01.2021. [Online]. Available: https://www.itu.int/rec/R-REC-M.2083

[8] X. Zhou, R. Li, T. Chen, and H. Zhang, "Network slicing as a service: enabling enterprises' own software-defined cellular networks," IEEE Communications Magazine, vol. 54, no. 7, pp. 146–153, 2016.

[9] I. Afolabi, J. Prados-Garzon, M. Bagaa, T. Taleb, and P. Ameigeiras, "Dynamic resource provisioning of a scalable e2e network slicing orchestration system," IEEE Transactions on Mobile Computing, vol. 19, no. 11, pp. 2594–2608, 2019.

[10] R. P. Jover, "Security attacks against the availability of lte mobility networks: Overview and research directions," in 2013 16th international symposium on wireless personal multimedia communications (WPMC). IEEE, 2013, pp. 1–9.

[11] A. Mathew, "Network slicing in 5g and the security concerns," in 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC). IEEE, 2020, pp. 75–78.

[12] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain, "Network slicing for 5g: Challenges and opportunities," IEEE Internet Computing, vol. 21, no. 5, pp. 20–27, 2017.

[13] C. Islam, M. A. Babar, and S. Nepal, "A multi-vocal review of security orchestration," ACM Computing Surveys (CSUR), vol. 52, no. 2, pp. 1–45, 2019.

[14] Nokia. (2019, nov) Security orchestration and automation for telcos. [Online]. Available: https://on24static.akamaized.net/event/21/42/96/2/rt/1/documents/resourceList1574339646268/lrwebinarsecuritypresentationoperationsfinal1574339643356.pdf

[15] B. Jaeger, "Security orchestrator: Introducing a security orchestrator in the context of the etsi nfv reference architecture," in 2015 IEEE Trustcom/BigDataSE/ISPA, vol. 1. IEEE, 2015, pp. 1255–1260.

[16] M. Pattaranantakul, R. He, A. Meddahi, and Z. Zhang, "Secmano: Towards network functions virtualization (nfv) based security management and orchestration," in 2016 IEEE Trustcom/BigDataSE/ISPA. IEEE, 2016, pp. 598–605.

[17] M. Pattaranantakul, Y. Tseng, R. He, Z. Zhang, and A. Meddahi, "A first step towards security extension for nfv orchestrator," in Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, 2017, pp. 25–30.

[18] A. Molina Zarca, M. Bagaa, J. Bernal Bernabe, T. Taleb, and A. F. Skarmeta, "Semantic-aware security orchestration in sdn/nfv-enabled iot systems," Sensors, vol. 20, no. 13, p. 3622, 2020.

[19] G. Mirjalily and Z. Luo, "Optimal network function virtualization and service function chaining: A survey," Chinese Journal of Electronics, vol. 27, no. 4, pp. 704–717, 2018.

[20] A. Hermosilla, A. M. Zarca, J. B. Bernabe, J. Ortiz, and A. Skarmeta, "Security orchestration and enforcement in nfv/sdn-aware uav deployments," IEEE access, vol. 8, pp. 131 779–131 795, 2020.

[21] D. Sattar and A. Matrawy, "Towards secure slicing: Using slice isolation to mitigate ddos attacks on 5g core network slices," in 2019 IEEE Conference on Communications and Network Security (CNS). IEEE, 2019, pp. 82–90.

[22] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A survey on federated learning for resource-constrained iot devices," IEEE Internet of Things Journal, vol. 9, no. 1, pp. 1–24, 2021.

[23] R. F. Olimid and G. Nencioni, "5G Network Slicing: A Security Overview," IEEE Access, 2020.

[24] D. Alotaibi, "Survey on network slice isolation in 5g networks: fundamental challenges," Procedia Computer Science, vol. 182, pp. 38–45, 2021.

[25] V. A. Cunha, E. da Silva, M. B. de Carvalho, D. Corujo, J. P. Barraca, D. Gomes, L. Z. Granville, and R. L. Aguiar, "Network slicing security: Challenges and directions," Internet Technology Letters, vol. 2, no. 5, p. e125, 2019.

[26] M. Ehrlich, L. Wisniewski, H. Trsek, D. Mahrenholz, and J. Jasperneite, "Automatic mapping of cyber security requirements to support network slicing in software-defined networks," in 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2017, pp. 1–4.

[27] R. Harel and S. Babbage, "5g security recommendations package# 2: Network slicing," NGMN Alliance, Apr, 2016.

[28] F. Z. Yousaf, V. Sciancalepore, M. Liebsch, and X. Costa-Perez, "Manoaas: A multi-tenant nfv mano for 5g network slices," IEEE Communications Magazine, vol. 57, no. 5, pp. 103–109, 2019.

[29] K. Gold. (2019, feb) The role of active and passive monitoring in virtual networks. [Online]. Available: https://www.exfo.com/en/resources/blog/active-passive-network-monitoring/

[30] T. Binz, U. Breitenbücher, O. Kopp, and F. Leymann, "Tosca: portable automated deployment and management of cloud applications," in Advanced Web Services. Springer, 2014, pp. 527–549.

[31] T. Irshad et al., "Design and implementation of a testbed for network slicing," 2018.

[32] Osm. Accessed on 11.07.2022. [Online]. Available: https://osm.etsi.org/

[33] Openstack. Accessed on 22.01.2021. [Online]. Available: https://www.openstack.org/

[34] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in 2009 IEEE symposium on computational intelligence for security and defense applications. Ieee, 2009, pp. 1–6.

[35] Suricata. Accessed on 22.01.2021. [Online]. Available: https://suricata-ids.org/

[36] Tcpreplay. Accessed on 22.01.2021. [Online]. Available: https://tcpreplay.appneta.com/

[37] A. Laszka, W. Abbas, S. S. Sastry, Y. Vorobeychik, and X. Koutsoukos, "Optimal thresholds for intrusion detection systems," in Proceedings of the Symposium and Bootcamp on the Science of Security, 2016, pp. 72–81.

[38] ETSI. Zero touch network & service management (zsm). Accessed on 23.01.2021. [Online]. Available: https://www.etsi.org/technologies/zero-touch-network-service-management

Shalitha Wijethilaka is currently a PhD student at School of Computer Science of University College Dublin, Ireland. He obtained his Bachelor's degree (First Class Honours) in Electronics and Telecommunication Engineering from the University of Moratuwa, Sri Lanka in 2017. He has industrial experience in IoT and telecommunication networks from 2017 to 2020. His research mainly focuses on improving network slicing security in telecommunication networks. In addition, he is interested in blockchain and federated learning in telecommunication, metaverse realization, and IoT security. Moreover, he works as a teaching assistant at UCD and a reviewer at IEEE Access, Spring Nature, and IEEE ICC. For more info: https://ucdcs-research.ucd.ie/phd-student/shalitha-wijethilaka

Madhusanka Liyanage (Senior Member, IEEE) is an Assistant Professor/Ad Astra Fellow and Director of Graduate Research at the School of Computer Science, University College Dublin, Ireland. He is also acting as a Docent/Adjunct Professor at the Center for Wireless Communications, University of Oulu, Finland, and Honorary Adjunct Professor at the University of Ruhuna, Sri Lanka and the University of Sri Jayawardhanepura, Sri Lanka. He received his Doctor of Technology degree in communication engineering from the University of Oulu, Oulu, Finland, in 2016. He was also a recipient of the prestigious Marie Skłodowska-Curie Actions Individual Fellowship and Government of Ireland Postdoctoral Fellowship during 2018-2020. In 2020, he received the "2020 IEEE ComSoc Outstanding Young Researcher" award by IEEE ComSoc EMEA. In 2021 and 2022, he was ranked among the World's Top 2% Scientists (2020 and 2021) in the List prepared by Elsevier BV, Stanford University, USA. Also, he was awarded an Irish Research Council (IRC) Research Ally Prize as part of the IRC Researcher of the Year 2021 awards for the positive impact he has made as a supervisor. In 2022, he received "2022 The Tom Brazil Excellence in Research Award" by SFI CONNECT Center. Dr. Liyanage's research interests are 5G/6G, Blockchain, Network security, Artificial Intelligence (AI), Explainable AI, Federated Learning (FL), Network Slicing, Internet of Things (IoT) and Multi-access Edge Computing (MEC). More info: www.madhusanka.com