Service Chaining Using Software-Defined Networks

Nikolaos Petroulakis[1], Konstantinos Fysarakis[2], Andreas Miaoudakis[1], Konstantinos Ramantas[3], Panos Chatziadam[1], and Christos Verikoukis[4]

[1] *Foundation for Research and Technology-Hellas (FORTH), Heraklion, Greece*
[2] *Sphynx Technology Solutions AG, Zug, Switzerland*
[3] *iqudrat, Barcelona, Spain*
[4] *Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Barcelona, Spain*

Introduction

With anticipated exponential growth of connected devices, future networks require an open-solutions architecture, based on modular interoperable components facilitated by open standards, to form a strong and flexible ecosystem. These devices need a simple interface to the connected network to request communication service characterized by specific quality of service (QoS) guarantees such as bandwidth, delay, jitter, packet loss, or redundancy. In response, the network should grant the required network resources automatically and program the intermediate networking devices based on device profile and privileges. A similar requirement also comes from business applications where an application itself asks for particular network resources based on its needs. Software-defined networking (SDN) and network function virtualization (NFV), important parts of 5G networking, provide promising combination, leading to programmable connectivity, rapid service provisioning, and service chaining and can thus help lower capital operational expenditure (CAPEX) and operational expenditure (OPEX) costs in the control network infrastructure (ETSI 2012; Naudts et al. 2012). Nevertheless, SDN and NFV also expand the attack surface of the communication infrastructure due to the centralized control of the network, necessitating the introduction of additional security mechanisms. Moreover, many of the vertical domains that 5G technologies will cover, such as industrial networks, typically come with strict performance, security, and reliability requirements.

Service function chaining (SFC) is one important enabler in this context, as it provides the ability to define an ordered list of network services (Zhang et al. 2018) to create a service chain, without having to consider the underlying network infrastructure. These services are then "stitched" together to create a service chain, with numerous options for adaptations when required (e.g. to adapt to link failures). The flexible traffic steering toward network functions enabled by SFC can also be leveraged to integrate novel, adaptable security services, such as steering suspicious traffic to security appliances. The

Wiley 5G Ref. Edited by Rahim Tafazolli, Chin-Liang Wang and Periklis Chatzimisios.

DOI: 10.1002/9781119471509.w5GRef101

deployment of these enhanced security concepts is in line with the enhanced protection requirements of certain sensitive application domains, such as critical infrastructures, given that the old paradigm of perimeter defenses and trusted internal networks is obsolete, as recent attacks have demonstrated (Department for Homeland Security 2016).

In the above context, this article aims to highlight the potential of SDN, NFV, and SFC to provide adaptable networking infrastructures with enhanced security and performance characteristics, providing an overview of the involved technologies as well as implementation specifics. Moreover, the application of these technologies in a reactive security framework is considered in two representative use cases in the context of 5G-PPP European project VirtuWind (Mahmoodi et al. 2016; Petroulakis et al. 2018) and the IoT European project SEMIoTICS (www.semiotics-project.eu):

1) an industrial network in the wind power generation domain which has been implemented, demonstrated, and evaluated in a real wind park in Brande, Denmark, and
2) an ambient-assisted living scenario in a smart home environment, for the well-being and independent living of the elderly. In addition to the above, a full implementation of an SFC-based security framework based on these concepts is presented, along with a performance evaluation, on a realistic test bed featuring various services and operational security service functions.

The remainder of this article is organized as follows. In the section titled "Background – Key Technologies", the background, motivation, and key technologies on SFC are presented, highlighting security-related aspects. Section titled "Related Works" presents related works regarding SFC-related research efforts that can be identified in the literature. Section titled "Security Considerations" provides related security considerations. In the section titled "SFC-Based Reactive Security Framework", the reactive security framework and its key implementation elements (e.g. security services and controller modules) are presented, while in the sections titled "Use Case 1 – Industry 4.0" and "Use Case 2 – Ambient-Assisted Living", a study on two use cases and associated application of the proposed framework is presented. Finally, the section titled "Conclusions" concludes this article with some discussion and pointers to future work.

Background – Key Technologies

Software-Defined Networking

By definition, when referring to SDN architecture, we refer to the decoupling of the control from the data plane and the centralization of the network management and logic. This centralization offers a simpler and less error-prone method of establishing and maintaining network policies, instead of the traditional vendor-specific methods. Control plane programmability presents the ability of adaptability to network changes without establishing permanence, by adapting to situations at run-time while maintaining a status quo configuration as the basis of the network's normal behavior. Additionally, for the first time, computer networks can adapt and evolve by utilizing complex networking services and applications that may be the result of the combined abstraction of simpler networking functions. The Open Networking Foundation (ONF) was founded to

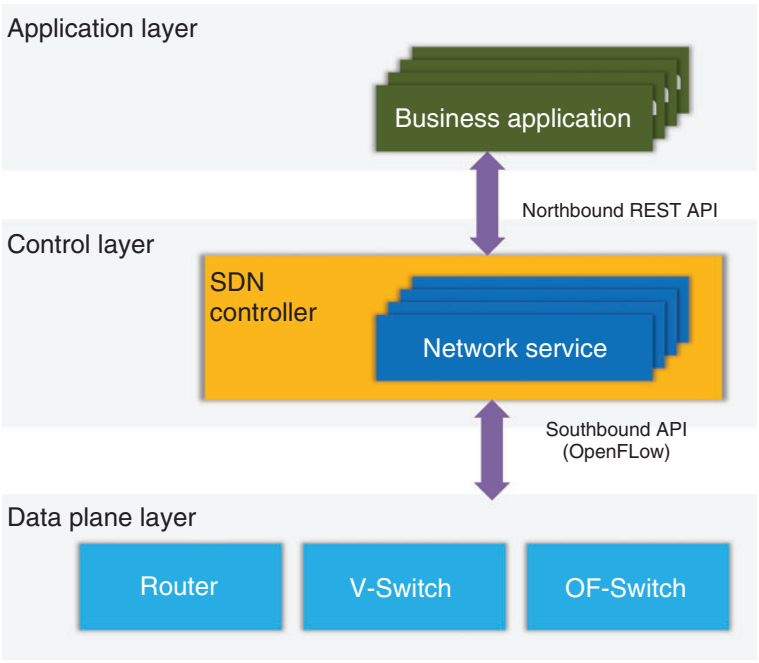

Figure 1 The three layers of the SDN architecture.

support the official architecture and technical specifications of SDN, as well as maintain a continuity in the development and improvement of the initial design. SDN networks consist of three logical layers as depicted in Figure 1.

- *The Application Layer*. Network applications, such as intrusion detection systems (IDSs), security systems, and monitoring systems, reside in this layer.
- *The Control Layer*. SDN controllers that provide centralized control logic to the data plane reside in this layer. SDN control plane functionality is maintained and propagated from this level.
- *The Infrastructure Layer*. All the networking elements, such as routers and switches, reside in this layer. SDN data plane functionality is executed at this level.

Communication between the three layers of the SDN architecture is established via application programming interfaces (APIs) that bridge the Infrastructure and Application layers to the Control layer. Specifically, we define two types of Interfaces, as follows (Prajapati et al. 2018):

- *Southbound Interface*. The API that facilitates the communication between the Control and Data planes. It is through this interface that the SDN controller, residing in the Control layer, sends flow rules to the networking elements at the Infrastructure layer. Aside from programmatic control, the Southbound interface provides controller capabilities advertisement, statistical information, and event notification (Open Networking Foundation 2014).
- *Northbound Interface*. The API that enables the communication between the Control and Application layers by utilizing representational state transfer (REST) APIs (Zhou et al. 2014). Through this interface, higher level applications, orchestration systems, and automation stacks may interact and request services from the SDN controllers at the Control layer.

In essence, SDN can be summarized by the following four principles that establish the foundation of the SDN architecture (Kreutz et al. 2015):

- *The Decoupling of Data and Control Planes*. The control logic is removed from the network devices, which become simple packet forwarders.
- *The Flow Rule Abstraction*. Forwarding decisions become flow based, which unifies heterogeneous network devices and becomes the new paradigm in the SDN realm.
- *The SDN Controller*. A traditional server system running specialized software that provides the necessary elements in order to empower the programmability of the forwarding devices.
- *Network Programmability*. The most important feature of SDN, which establishes the control of the Data plane where the forwarding devices reside.

Service Function Chaining

In typical network deployments, the end-to-end traffic of various applications typically must go through several network services (e.g. firewalls). It can also be referred to as Service Functions (or L4–L7 Services, or Network Functions, depending on the source/organization) that are placed along its path. This traditional networking concept and the associated service deployments have a number of constraints and inefficiencies (Quinn and Tom 2015), such as:

- *Topology Constraints*. Network services are highly dependent on a specific network topology, which is hard to update.
- *Complex Configuration and Scaling Out*. A consequence of topological dependencies, especially when trying to ensure consistent ordering of service functions and/or when symmetric traffic flows are needed; this complexity also hinders scaling out the infrastructure.
- *Constrained High Availability*. As alternative and/or redundant service functions must typically be placed on the same network location as the primary one.
- *Inconsistent or Inelastic Service Chains*. Network administrators have no consistent way to impose and verify the ordering of individual service functions, other than using strict topologies – on the other hand, these topology constraints necessitate that traffic goes through a rigid set of services functions, often imposing unnecessary capacity and latency costs, while changes to this service chain can introduce a significant administrative burden.
- *Coarse Policy Enforcement*. Classification capabilities and the associated policy enforcement mechanisms are of coarse nature, e.g. using topology information.
- *Coarse Traffic Selection Criteria*. As all traffic in a particular network segment typically has to traverse all the service functions along its path.

The above are exacerbated nowadays, with the ubiquitous use of virtual platforms, which necessitates the use of dynamic and flexible service environments. This is even more pronounced in service provider and/or cloud environments, with infrastructures spanning different domains and serving numerous tenants. Said tenants may share a subset of the providers' service functions, and may require dynamic changes to traffic and service function routing, to follow updates to their policies (e.g. security) or Service-Level Agreements.

SFC aims to address such issues via a service-specific overlay that creates a service-oriented topology, on top of the existing network topology, thus providing service function interoperability. An SDN-based SFC architecture, such as the one defined by the Open Networking Foundation (Open Networking Foundation 2015), can extend this concept, exploiting the flexibility and advanced capabilities of software-defined networks, to provide novel and comprehensive solutions for the abovestated presented weaknesses of the legacy networks.

Terms and Definitions

In this subsection, the terms and their definitions, as used in this article, are mentioned. The definitions of SFC terms are described in IETF, SFC architecture (Halpern and Pignataro 2015), and SFC environment security requirements (Migault et al. 2016). The key terms include:

- *Network Service Function*. A function that is responsible for specific treatment of received packets.
- *Service Function Chaining*. A service function chain defines an ordered set of abstract service functions and ordering constraints that must be applied to packets and/or frames and/or flows selected as a result of classification.
- *Service Function Forwarder*. An SFF is responsible for forwarding traffic to one or more connected service functions according to information carried in the SFC encapsulation, as well as handling traffic coming back from the service function (legacy or virtual).
- *Service Function Path*. A constrained specification of where packets assigned to a certain route must go. Any overlay or underlay technology can be used to create service paths (VLAN, ECMP, GRE, VXLAN, etc.).
- *Service Function Classifier*. An entity that classifies traffic flows for service chaining according to classification rules defined in an SFC Policy Table and to mark packets with the corresponding SF Chain Identifier. It can be on a data path, or run as an application on top of a network controller.
- *SFC Header*. A header that is embedded into the flow packet by the SFC to facilitate the forwarding of flow packets along the service function chain path. This header also allows the transport of metadata to support various service chain-related functionality.

Related Works

Several SFC-related research efforts can be identified in the literature. Nevertheless, a recent survey on the use of SFC (Bhamare et al. 2016) reveals a lack of work focusing on security-related applications, and this is a gap that the framework presented herein can cover. In terms of the key technological building blocks, Network Service Headers (NSHs) (Quinn and Guichard 2014) is an approach that involves the introduction of SFC-specific 4-byte headers that include all the information needed (including associated metadata) to reach a policy decision with regard to what service chain the traffic should follow. As part of the relevant IETF efforts, the NSH approach has been extended to define a new service plane protocol (a dedicated service plane) for the creation of

dynamic service chains (Quinn 2015); this NSH-based SFC approach is adopted in the framework presented herein.

StEERING (Zhang et al. 2013) is an OpenFlow-based alternative that allows for per-subscriber and per-traffic type/application traffic routing to the various service functions, via simple policies propagated from a centralized control point, but does not consider the security-based classification that forms the basis of the work presented here. Researchers have also introduced SIMPLE (Qazi et al. 2016), a policy enforcement layer that focuses on middleware-specific traffic steering and considers the inclusion of legacy service instances into the chain. It is based on monitoring and correlating packet headers before and after they traverse a specific service function, though this leads to a rather complex process (collecting packets for correlation, matching packets with high accuracy, etc.).

The chaining of virtual network functions (VNFs) is another aspect examined in the literature, which considers the trend of virtualizing networks and network functions in modern networks. More specifically, ETSI proposes a security management and monitoring specification in NFV that enable active and passive monitoring of the VNF and the SFC as provisioned in the NFV environment (ETSI GS NFV-SEC 013 2017). From this perspective, Mehraghdam et al. (2014) present a formal model for specifying VNF chains and propose a context-free language for denoting VNF compositions. Chain definitions in the work presented here are based on the structured format required by the test-bed controller (i.e. ODL), but a formal-based definition could be used if the corresponding module is appropriately extended, provided that the added complexity is justified by the application requirements. Blendin et al. (2014) exploit Linux namespaces to create isolated service instances per service chain, allowing one-to-one mapping of users to service instances; nevertheless, such an approach is not necessary in industrial environments, where, typically, the number of users is limited, and the management of multiple service instances can incur a significant administrative burden.

VNF chaining has a prominent role in 5G networks, and many works address the question of how vertical applications can be efficiently deployed on top of the virtualized 5G infrastructure. In Bruschi et al. (2019), a new architecture and data model for 5G-Ready adaptation is proposed, where monolithic apps are segmented into multiple VNFs, where each VNF implements a specific application functionality. A fundamental problem related to SFC is the deployment of service function chains over virtualized infrastructure. Bhamare et al. (2016) study VNF placement problem for the optimal SFC formation across geographically distributed clouds. Furthermore, Bhamare et al. (2016) propose frameworks and SDKs to automatically compose such chains leveraging SDN and NFV, aiming to reduce OPEX and CAPEX for network operators, while Parada et al. (2018) contribute tools and SDKs to simplify the development and deployment of 5G verticals on top of NFV Infrastructures. It must be noted that since the state of the art (SoA) of SFC focuses on VNFs and virtual machines (VMs), containers that can be seamlessly instantiated in a lightweight manner play an increasingly important role in modern virtualized infrastructures. Thus, hybrid Service Function Chains that converge VNFs and container network functions (CNFs) have been proposed in Mimidis et al. (2018). This allows container-based microservices (i.e. CNFs) to be chained with VNFs and jointly orchestrated by the same NFV Management and Network Orchestrator (MANO) (Ersue 2013) framework.

Security Considerations

Configuration errors, bugs, and security lapses are just some of the potential issues resulting from the time-consuming configuration process of legacy networking devices. The centralized network intelligence of SDN comes to the rescue by creating innovative ways to provide a layer of security, which is unattainable with traditional networks.

SDN is still evolving and is fast gaining momentum toward replacing legacy networks. While the low operational and capital expenditures are big selling points for the industrial community, the academic community appears to be excited about the potentials SDN can bring to the network. Security is one of the most recurrent fields of SDN-based research, and it is divided into two major branches (Ali et al. 2015), Protecting the Network and SDN Security as a Service.

Protecting the Network

The centralization of the control plane empowers SDN to provide a significant level of security, out of the box. By design, SDN has not only the potential of fast security event detection but also the potential for remediation and self-healing. In the case of a Distributed Denial of Service (DDoS) attack, the network can be dynamically reprogrammed to drop the malicious flows, while in the event of a malware infection, the compromised part of the network can be instantly isolated to prevent further spreading. The centralization of the Control plane reduces network exposure. Trust is no longer exposed by being distributed among the plethora of network devices, and therefore little intelligence can be gained from any of these devices. In addition, programmability provides the foundation for building flexible policies that may adapt according to the network state and usage.

Another advantage of the control plane centralization is the holistic visibility of the entire network. By collecting traffic statistics from the network devices, the SDN controller has a real-time view of the network state. The exploitation of such information can lead to the creation and enforcement of reaction mechanisms. Real-time traffic monitoring is one of the strong points of SDN and can easily be utilized to rapidly detect DDoS attacks and network traffic anomalies. Established policies combined with heuristic and even machine learning mechanisms (Nanda et al. 2017) not only can detect but also predict network attacks. Reaction to detected threats is also a strong point of SDN. Policy and programmability-based remediation can apply targeted, flexible, and smart countermeasures that no longer affect the entire network.

SDN Security as a Service

SDN programmability provides the ability to instantiate customized services on a per-need basis. When it comes to SDN security, an elastic model can be adopted empowering the creation of security services that are appropriate to exist under specific circumstances and not on a permanent basis. Security as a service is not a new concept when it comes to network security, and in fact it has been used extensively in the complex and mission critical environments of Cloud computing (Rittinghouse and Ransome 2016).

Nevertheless, when it comes to the SDN paradigm, there is a lot more to be achieved by security as a service than in traditional environments. Identity protection and anonymity have surfaced as two of the most critical factors to protect in the Internet world. SDN is able to provide an elegant and functioning approach, with almost no significant overhead, at line speeds. In such approach, the SDN controller can utilize flow rules to create routing policies directed at many networking elements that actually perform the anonymization function with minimal delays and overhead (Mendonca et al. 2012).

An additional application of SDN-based security as a service is the delegation of network security management to a third party. An obvious application for this is the protection of home networks where network security knowledge is either poor or even not existing. The majority of today's devices (i.e. broadband modems and routers) that connect the home users to their Internet Service Provider (ISP) already utilize some third-party security mechanisms using Domain Name System (DNS) services such as OpenDNS and Google DNS. Now if we consider that these home routers and modems are OpenFlow capable, we can easily imagine that these devices can potentially connect to an SDN controller, perhaps operated by the ISP or even a third-party service provider. The SDN controller will be able to receive traffic statistics from the home devices and will process these statistics utilizing malicious content detection algorithms and spam databases. Then, the SDN controller will be able to dispatch customized flow rules to keep the subscriber's network safe. The same approach may also be applied to small and medium businesses to provide security as a service option to accommodate their specific needs.

Moreover, SFC allows us to revisit old security concepts (e.g. firewalls) but introduce new techniques as well, by adopting moving target defense and other reactive techniques. For example, SFC can be used to route unknown/suspicious traffic via Intrusion Detection and Deep Packet Inspection service functions, to classify it (as either legitimate or malicious), allowing the isolation of malicious traffic in honeypot, and thus allowing to track and occupy the attacker, while identifying the purpose and means of the attack.

SFC-Based Reactive Security Framework

Motivated by the above, a Reactive Security Framework for next-generation 5G (and SDN/NFV in specific)-enabled networks leveraged by SFC is presented herein, building upon the concept presented in Fysarakis et al. (2017). More specifically, considering the key vertical domains of 5G networks, the framework features security and performance functions, such as firewalls and load balancers, and allows the continuous network monitoring, with provisions to reduce the performance impact of the security functions and to alleviate the burden of deploying and managing the security services themselves. Moreover, the framework's Honeynet facilitates the detailed analysis of potential attacks, isolating attackers and enabling the assessment of their level of sophistication (e.g. from script kiddies to state actors).

One of the goals of this effort is to provide a secure networking infrastructure, via the associated security mechanisms, such as network monitoring and intrusion detection for SDN. To achieve this, the presented security framework includes network

monitoring and intrusion detection for identification of attacks and run-time adaptation for attack response and mitigation mechanisms. By leveraging security network functions such as Firewalls, IDS, DPI, Honeypots, and Honeynets, the framework can create a number of service function chains, to forward traffic based on the traffic type or running application, overcoming constraints and inefficiencies, as mentioned previously. This can be used to provide security profiles per-application classification based on the originating application, or per-tenant classification serving multiple virtual tenant networks (VTNs) with the chaining of vital security functions or, alternatively, per-traffic classification, for both intra- and interdomain deployments, using predefined Service Function Paths for each traffic type.

In contrast to the proactive deployment of specific security mechanisms (typically at the network's design phase), reactive mechanisms are also employed to react in real time to changes in the network as well as the traffic traversing said network (e.g. to automatically mitigate attacks, block malicious entities, and route them to specific, dummy network components) or even trigger the deployment of new security functions to help alleviate the effects of an ongoing attack. By leveraging the flexibility of SDN-based deployments and the concept of SFC, a service-specific overlay creates a service-oriented topology, on top of the existing network topology, thus providing service function interoperability.

Architecture Sketch

In the current network infrastructures, network functions and applications are typically running on dedicated locations that were decided in the network planning phase and are very difficult to change. The approach proposed here enables future scenarios to add or delete functionalities or applications during runtime. In order to enhance this flexibility, the principles of NFV MANO can be combined with the presented framework. The expanded framework architecture can be aligned to the approach described in ETSI GS NFV 002 (2014), as depicted in Figure 2. This provides flexible deployment and instantiation of new VNFs and automated preparation of service function chains.

Implementation Approach

This work follows closely the standardization efforts of IETF, and the SFC Working Group (2019) in specific, building on top of the work of the Open Networking Foundation and the associated OpenDaylight Controller modules. Moreover, special care is given to the security of the SFC mechanisms, e.g. by guaranteeing the integrity of SFC-related data added to the packets for identifying the service functions chains, and by ensuring that no sensitive SFC data (and the associated metadata) crosses different SFC domains, or legacy networks, unprotected.

To implement the above functionality, other than the security service functions themselves (e.g. IDS and Honeypots) that need to be installed and setup appropriately, certain purpose-built modules as well as enhancements to existing SDN controller modules are needed.

SFC Manager

SFC Manager is a controller module that exposes a number of interfaces that various components can use to provide and receive information about service chains that need

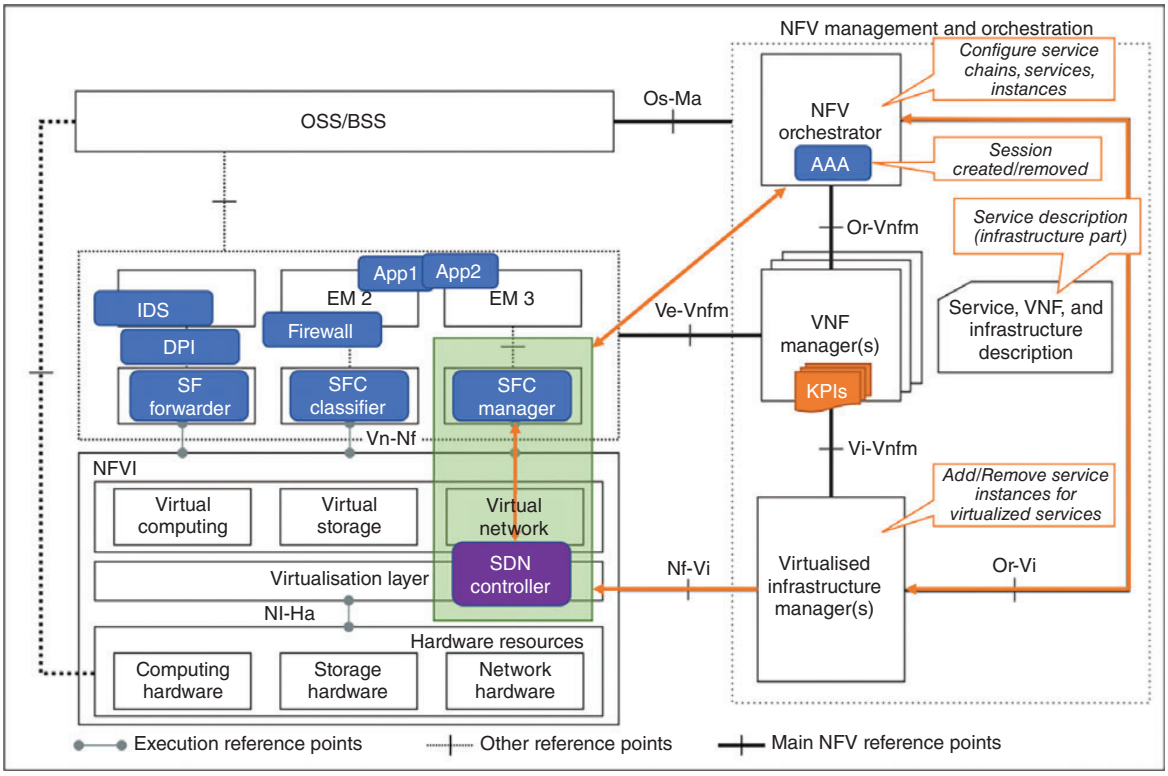

Figure 2 The proposed framework within the ETSI NFV architecture.

to be built, which tenants want to use them, which destinations are being accessed, what applications the traffic pertains to, and about the service instances of the network functions. Each SFC configuration includes a set of service nodes, a set of service functions, a set of service function forwarders, a set of service chains, a set of service paths, and a set of configurations for classifiers (access control list (ACL)/NSH). This information is generated by the NFV Orchestrator based on SFC requests and sent to the SFC Manager as depicted in Figure 2. The SFC Manager aggregates this information, combines it, and sends service chains in commands to the SDN controller (OpenDaylight 2018) and SFC-ODL (ODL wiki 2019). The SDN controller, in turn, programs the underlying forwarding elements that do the actual packet forwarding. In essence, the SDN controller converts commands from the high-level SFC language to the low-level flow filters expressed in the OpenFlow semantics.

The SDN controller provides an abstraction view of the network topology. This allows the SFC Manager to focus on the chaining itself and not on the internal topology of the Network Controller. This means that SFC Manager manages forwarding rules and flow filters on external ports, and thus configuration is significantly simplified. However, the SDN controller does the necessary transformations to put the paths (sequence of service instances) and filters (associate user based on his profile to its respective service chain) in the forwarding devices. In particular, the job of SFC Manager is to register external ports of the SDN transport network (which is being used for SFC) and to declare and associate service instances to those external ports. Such service instances may include vFirewall, IDS, DPI, and Honeypot. These services may be composed of one or multiple instances and may be physical appliances or virtual machines running in network function virtualization infrastructure (NFVI). At the Management and Control planes, the

SFC Manager and the SFC-enabled SDN controllers are responsible for administrating the services chains, i.e. for translating the operator's/tenant's/application's requirements into service chains. At the Data plane, Classifiers are responsible for assigning traffic to the appropriate service chain (based on various criteria, such as its maliciousness or the tenant that it belongs to, assuming tenant identities have already been validated by authentication/authorization components), and Service Forwarders and Proxies (where needed) are responsible for steering traffic accordingly to realize said Service Chains. If the Service Function Nodes are not OpenFlow-speaking or SFC-aware, or are in different domains, SFC Proxies are needed.

Security Service Chaining

Leveraging the benefits of SFC involves reversing this trend for monolithic "all-in-one" security services, which are now commonplace. In the context of SFC, the focus is on breaking up complex services into dedicated service functions, each providing a single task. This shift is not dissimilar to the emergence of the microservices as described in Thönes (2015), which moves developers away from the once dominant paradigm of building entire applications as a monolith toward applications made up from a number of smaller services, each of them performing a single function (adopting the "Do one thing and do it well" philosophy). Some key security mechanisms that are leveraged in the reactive security framework and deployed as virtualized network service functions are as follows.

Intrusion Detection System (IDS) is a service able to identify suspicious activities or attack violations. More specifically, IDS instances of Snort (2018) are deployed, with scripts to ensure that the most up-to-date rules are constantly active. A database for event monitoring is present, while provisions are made to allow for future extensions to transmit relevant information to a security backend (e.g. for more sophisticated pattern matching). Moreover, a SCADA-specific instance of Snort (Snort 2.9.2 2018) is also deployed for the wind park use case, where SCADA traffic will be routed. This limits the delay imposed on the SCADA traffic by the IDS functionality (a delay that significantly depends on the number of rules/patterns in the IDS's database, which will be significantly lower in the case of the IDS, which only has SCADA-specific rules installed).

Honeynet is formed by a set of functions (honeypots) emulating a production network deployment, able to attract and detect attacks, and acting as a decoy or dummy target. A honeypot is a decoy deployment that can fool attackers while at the same time used to collect information about the attacker and attack method. Network-based honeypots have been widely used to detect attacks and malware. A Honeynet is deployed in the framework, consisting of Honeypots emulating SDN and other network elements, as well as Honeypots emulating the operational systems and, more specifically, elements (such as the SCADA systems for the wind park use case). Moreover, passive Honeypots (early warning intrusion detection systems, EWIS, in specific (Chatziadam et al. 2014)) are also a part of the Honeynet, acting as a network telescope on the production part of the industrial network, to monitor all activities in normally unused parts of the network.

Firewall is a service or appliance running within a virtualized environment providing packet filtering. Firewall instances are also deployed to implement network perimeter security. It is worth to be noted that the type of firewall (software or hardware based), as well as its placement, is irrelevant in the context of the reactive security framework,

as the service plane view of the framework focuses on the type of service and not the underlying technology that is used to offer this service.

DPI is a function for advanced packet filtering (data and header) running at the application layer. In the proposed frameworks, an implementation of nDPI (2019) is employed to implement DPI functionality to the framework. The response of the *nDPI engine* is based on a set of rules that the engine has compiled to classify traffic types, and can be extended by writing additional rules (for instance, in the wind park use case, TCP and user datagram protocol (UDP) ports 502 associated with Modbus traffic can be defined as being malicious, if such traffic is not expected in the specific part of the network). The response from the *nDPI engine* classification of each packet is either the protocol/application/framework ID that the abovementioned rules define or *UNKNOWN* if it could not be determined.

Other than the ones employed above, other Service Functions could be included in a real deployment such as:

- *Network Virtualization*, via the use of Virtual eXtensible LocalAreaNetwork28, a VLAN-like encapsulation technique to encapsulate MAC-based OSI layer 2 Ethernet frames within layer 4 UDP packets, brings the scalability and isolation benefits needed in virtualized computing environments.
- *Access Control Lists (ACLs)* are used at the perimeter of each network domain to route traffic to the appropriate isolated virtual networks and the corresponding security service functions.
- *Packet Inspectors* are utilized to detect malformed packets or malicious activity (Internet protocol flow information export and denial-of-service).
- *Secure Communication Protocols* with packet encapsulation are used to provide secure communication channels (e.g. Internet protocol security). It is worth to be noted that, security functions as the above are some of the principal service functions considered by IETF when presenting SFC use cases pertaining to Data Centers (Kumar et al. 2015) and Mobile Networks (Haeffner et al. 2015).
- Others such as Load Balancers, HTTP header enrichment functions, TCP optimizers, and Resource Signaling.

Use Case 1 – Industry 4.0

One of the main objectives in industrial networks is faster service provisioning. The time to provide the service is foreseen to be reduced from several days to several minutes. The concept of SFC has already shown promising results in enabling the faster time-to-market for the new services in the domain of telecom operators. This also implies the potential to reduce CAPEX and OPEX, especially for short-lived service. Depending on the focused aspect, which is of relevance for each deployment of the proposed SFC-enabled framework, some use cases flavors (subuse cases) can be identified.

Per-Tenant Type Classification

In the context of next-generation industrial networks, one of the promised services is the possibility to instantiate VTNs on demand. The purpose of virtual tenant networks

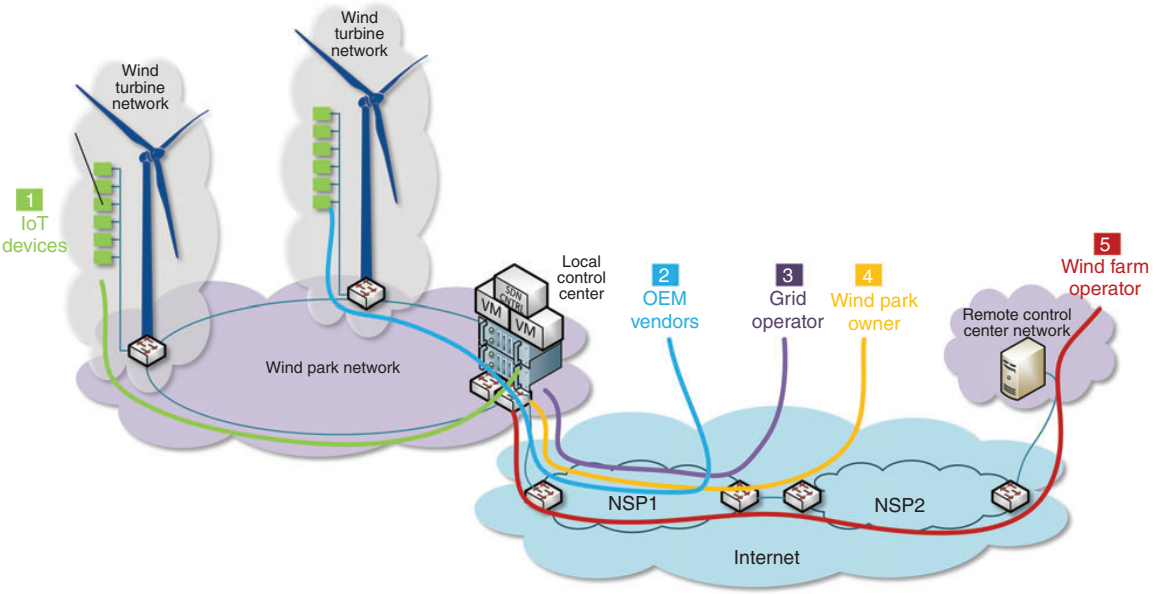

Figure 3 Wind park stakeholders.

is to set up virtual networks which contains a set of functionalities on the same physical infrastructure. One objective is that different VTNs are not influencing each other. From industrial networks perspective the tenants can be related to different stakeholders in a wind park, as shown in Figure 3. The stakeholders may include wind farm operator (WPO), transmission system grid operator (TSO), original equipment manufacturer (OEM), IoT device vendors, wind farm owner, and SCADA application.

Each of the stakeholders can have different requirements and constrains for setting up a dedicated VTN for their purposes. The different requirements and constrains can lead to different VTN flavors when talking about the VTN setup and network configuration. SFC could be exploited to instantiate VTN, but beyond that aspect, each tenant should be able to add functions on demand in its network, and control the function of its own VTN. In the above context, two different flavors of tenant-based chain classification can be envisioned, which are presented below.

Security Services On-Demand Tenant Use Case

As an example, let us assume that equipment vendor (e.g. Siemens) requests a VTN to inspect the turbines of wind park operator (e.g. EON). Siemens would request a slice of the operator's network allowing it to access (only) the turbine configuration and log files, with certain quality of service (e.g. high availability). Additionally, Siemens must be able to create the list of the engineers and the technicians who are authorized to do a troubleshooting on turbines of a given wind park. Siemens and/or EON would also like to make sure that members of VTN are performing only the authorized tasks (upgrade, traces, etc.). Hence, tenant-specific security components should be added to a security function chain of the wind park. Examples of tenant-specific functions are ACL and DPI. The other security functions, such as firewall and IDS, might be still shared with other VTNs in the wind park.

In another scenario, the flexibility of function chaining could allow tenants to change the deployed security mechanisms dynamically. Thus, for example using tenant-based classification, Tenant 2 could only be using a firewall protection, while Tenant 1 could

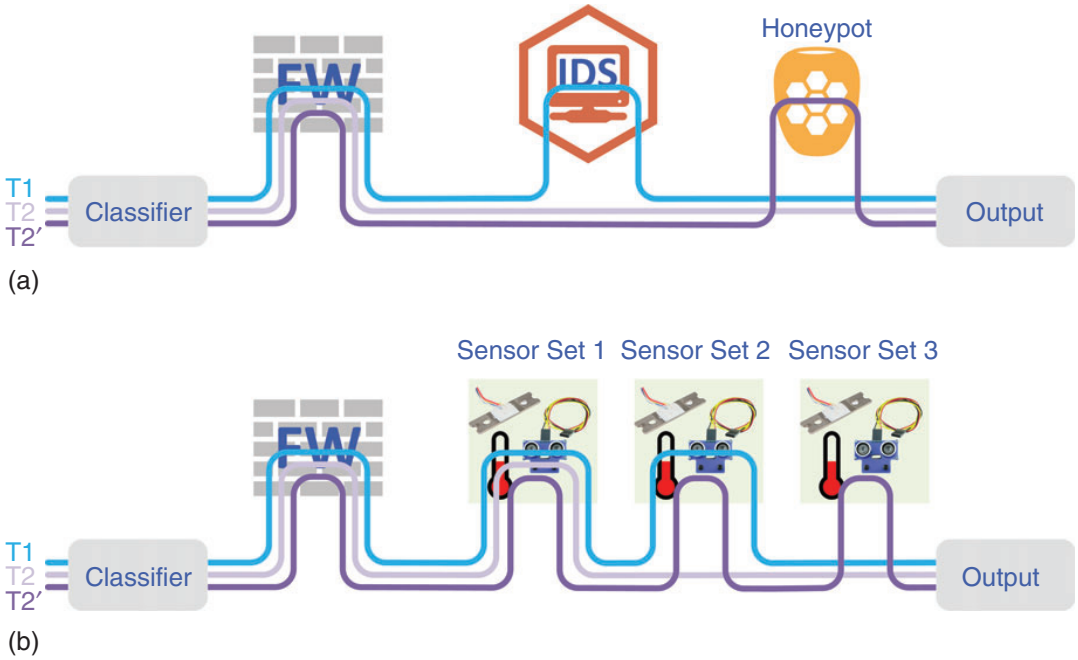

Figure 4 Per-tenant classification options.

be using a firewall and an IDS appliance. During operation, the preferences of Tenant 2 could be updated (Tenant 2′), to also necessitate the presence of a Honeypot or Honeynet, triggering the corresponding update to his/her function chain. This is depicted in Figure 4a, which presents an example of such a setup with the following chain definitions:

- Chain 1 – Tenant 1: Firewall → IDS → Output
- Chain 2 – Tenant 2: Firewall → Output
- Chain 3 – Tenant 2': (after update): Firewall → Honeypot/Honeynet → Output

Industrial Internet of Things Tenant Use Case

Considering the Industrial Internet of Things (IIoTs) in the wind park use case, and in the context of Tenant-based classification, SFC could also be exploited at the application level, in order to provide dynamic, real-time access to the required data of the IIoT sensors. Therefore, depending on the tenant's requirements/agreement, etc., each of them get access to a different subset of data monitored by IIoT sensors, even though all tenants will reach the same resource (e.g. web interface on backend monitoring server). This could be achieved by a simple HTTP header enrichment service running on each of the Service Functions, with each of these services adding the corresponding subset of sensed data into the final web pages that the tenants will see on their web browsers. An example of the above concept is depicted in Figure 4b, whereby the following chains are defined:

- Chain 1 – IIoT Tenant 1: Sensors' Set 1 → Output
- Chain 2 – IIoT Tenant 2: Sensors' Set 1 → Sensors' Set 2 → Output
- Chain 3 – IIoT Tenant 3: Sensors' Set 1 → Sensors' Set 2 → Sensors' Set 3 → Output

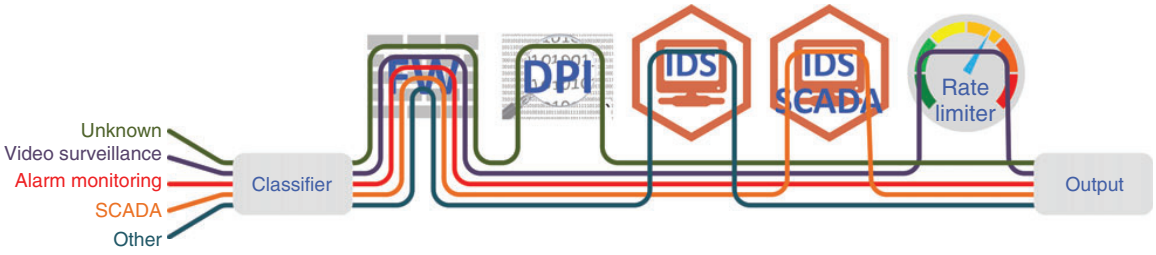

Figure 5 Per-application classification.

Per-Application Type Classification

This variation of the SFC application classifies traffic based on the originating application. Thus, after a stage of Deep Packet Inspection, the Application is identified and the corresponding chain is assigned. An example of chains tailored to specific applications could include forwarding SCADA traffic to a SCADA-specific IDS, and a generic IDS for other traffic, thus limiting the delay imposed on the SCADA traffic by the IDS (as it depends on the number of rules/patterns in the IDS's database, which could be significantly lower in the case of an IDS which only has SCADA-specific rules installed). Another example could be having video surveillance traffic go through Firewall and a Rate Limiter, to lower the transmission rate, respecting QoS requirements. Thus, potential chains in this case (Figure 5) could include:

- Chain 1 – Unknown application: Firewall → DPI → Output
- Chain 2 – Video surveillance: Firewall → Rate Limiter → Output
- Chain 3 – Alarm monitoring: Output
- Chain 4 – SCADA: Firewall → SCADA IDS → Output
- Chain 5 – Other application: Firewall → IDS → Output

Per-Traffic Type Classification

This scenario includes a security SFC-based enhancement, for both intra- and interdomain deployments, with the ability to forward traffic based on its security classification (e.g. unknown/malicious/legitimate), following predefined Service Function Paths for each traffic type. This type of classification opens up various possibilities for the integration of advanced malicious traffic detection techniques (e.g. exploiting machine learning). As an example, let us assume that a data packet enters the intradomain wind park deployment. Based on its classification (from the categories listed above), the traffic will be directed to one of the three different paths as depicted in Figure 6a. The aim for this process is to route unknown/suspicious traffic via the Intrusion Detection and Deep Packet Inspection Service Functions, in order to classify it (as either legitimate or malicious), thus allowing us to forward it to the wind park or the honeypot, accordingly.

For the interdomain use case, see Figure 6b, the procedure is similar to the intradomain scenario. However, a more sophisticated honeypot deployment, such as a Honeynet, can be used as an emulated wind park, having similar services and functions as the original wind park. Moreover, in this case, having acquired the needed tag (as malicious or legitimate) in other parts of the larger wind park deployment, the traffic can avoid

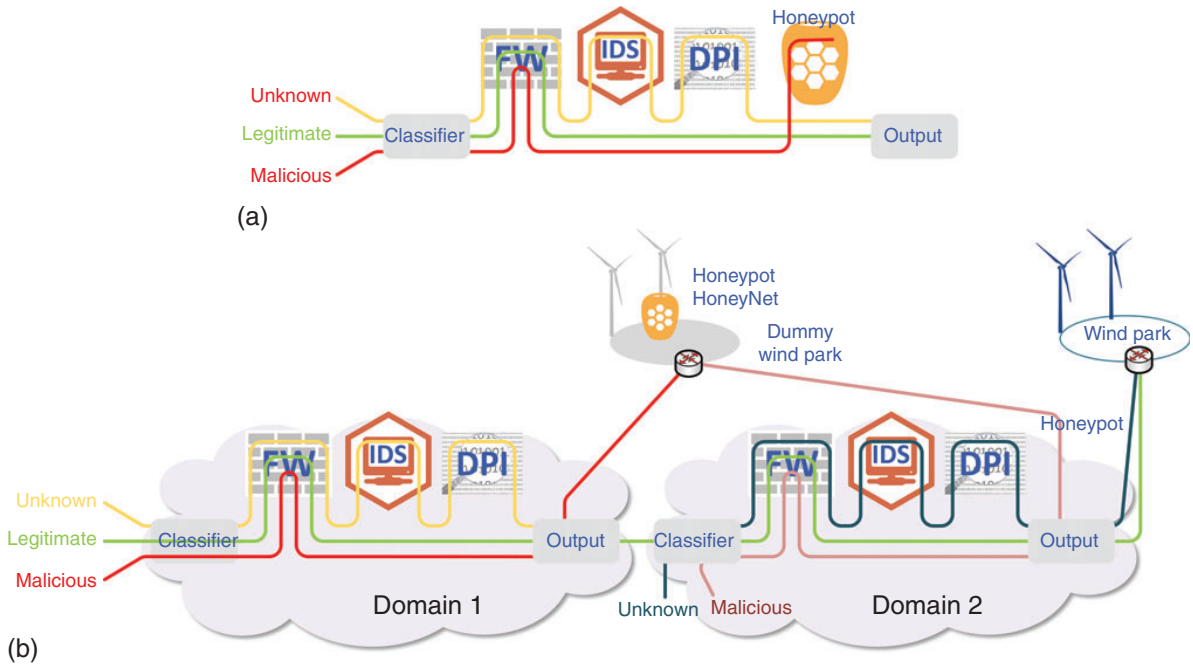

Figure 6 Per-traffic type classification options for inter- and intradomain.

going through the same procedures (i.e. Service Functions) again, better highlighting the benefits of SFC in terms of potential performance gains. A core part of this use case is the classifier. The classifier is responsible for classifying and forwarding packets based on predefined rules, exploiting pattern matching and tags found on the packet headers. The classifier (attached to the SFF) forwards the packets through one of the predefined function chains. In more details, based on the classification of each packet, the traffic can be classified as legitimate, unknown (suspicious), or malicious. Thus, three different chains are defined:

- Chain 1 – Legitimate (known) traffic: Firewall → Output
- Chain 2 – Suspicious traffic: Firewall → IDS → DPI
- Chain 3 – Malicious traffic: Honeypot/Honeynet

Use Case 2 – Ambient-Assisted Living

The European population growth is slowing down, while population aging accelerates (Chłoń-Domińczak et al. 2014). Eurostat projects that the ratio of people aged 65 and over relative to those of working age (15–64 years), namely the old-age dependency ratio, is projected to increase from 28.8% to 51.6% between 2015 and 2060 in the EU-28's populations (OECD 2015). The progressive decline in physical and cognitive skills prevents elderly people from living independently and from performing basic instrumental activities of daily living. Over recent years, there has been an increase in interest in e-health monitoring systems situated at homes, leading to the creation of Health Smart Homes. Such technologies can facilitate the monitoring of patients' activities and enable healthcare services at home. They improve the quality of elder population well-being in a nonobtrusive way, allowing greater independence, maintaining good health, and

preventing social isolation for individuals, and delay their placement in institutions such as nursing homes and hospitals.

In this context, the second use case for the proposed framework focuses on an ambient-assisted living scenario, whereby a smart home environment includes the following:

- *Body Area Network (BAN).* Short-range network of wearables (e.g. sensors and identification tags carried or worn on the patient's person) for fall detection, fall risk assessment, and other mobility-related data.
- *Robotic Rollator*. A powered, wheeled walking frame, primarily used for physical support, but also equipped with various sensors and computational units, and capable of identifying a patient (the user of the rollator) and monitoring their behavior (e.g. gait & posture).
- *Mobile Phone.* User's mobile phone that acts a gateway for the BAN devices, as well as the Robotic Rollator devices (but only in case of outdoors use).
- *Smart Home Infrastructure.* Sensors, actuators, lighting, climate control, and other smart devices, as well as the corresponding gateway(s), that comprise a smart living environment.
- *Robotic Assistant*. A robotic component for monitoring a patient's activities (ADL data), health status, and treatment/training progress, as well as for supporting cognitive skills training, notifying/reminding the patient of upcoming treatments (e.g. medication & training schedules) and visits.
- *Backend*. The backend system providing an assortment of assistance services for the elderly, and being monitored by caregivers and healthcare professionals.

The above scenario sketches a complex environment, requiring support for integration of heterogeneous devices and communication protocols, high degrees of interoperability, and support for distributed services and applications (each with its own set of intrinsic requirements), while guaranteeing the safety of the patient and the security and privacy of her patient data. This use case is visualized in Figure 7b, which depicts the various types of devices, their interactions, and the involved communication technologies.

Investigating this use case, and considering the different types of traffic reaching the backend where the chaining of services will take place, the following intricacies are observed: traffic originating from the mobile phone is of low trust and low priority, as the mobile device is not trusted (e.g. can be easily targeted by malicious software) and the reporting from the BAN devices has low bandwidth and latency requirements; traffic from the Robotic Rollator is of medium trust (relatively restricted devices) but high priority, as messages need to arrive in a timely fashion (e.g. in case a patient fall is detected); the smart home traffic is of medium trust (commercial devices that may be vulnerable to, e.g. incorrect configuration) and of low priority; and finally traffic from the robot are of high trust (closed/restricted device) and high priority, as low latency and relatively high bandwidth is required to enable seamless interactions with the robot. Considering the above, the defined service chains (Figure 7a) are as follows:

- Chain 1 – Mobile Phone: Firewall → DPI → IDS → Output
- Chain 2 – Robotic Rollator: Firewall → IDS → Load Balancer → Output
- Chain 3 – Smart Home: Firewall → IDS → Output

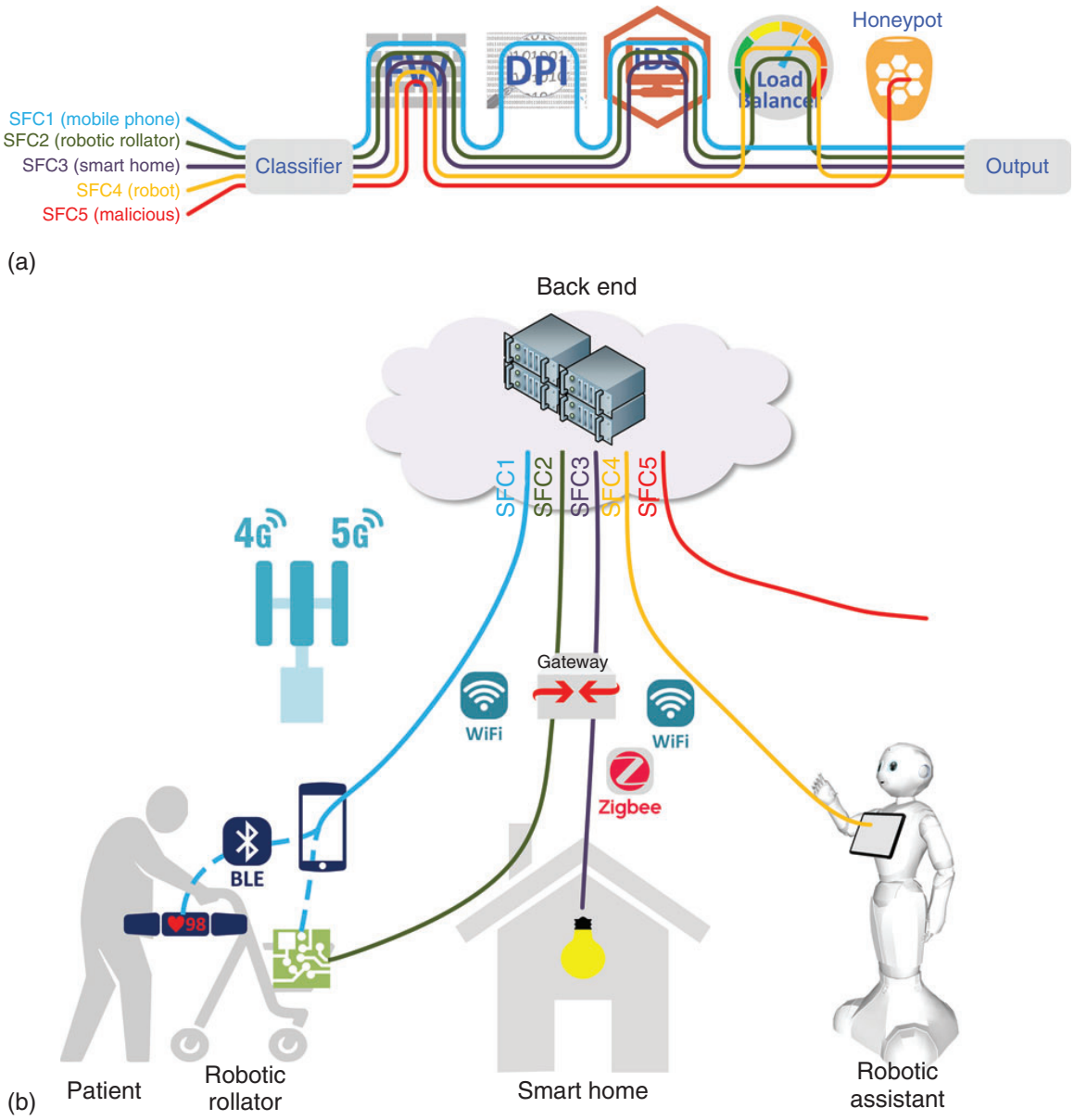

Figure 7 Ambient-assisted living scenario and traffic classification.

- Chain 4 – Robot: Firewall → Load Balancer → Output
- Chain 5 – Malicious: Firewall → Honeypot

Thus, through the definition of the said chains, each of the traffic types gets routed through a chain of service functions tailored to its intrinsic requirements and characteristics, such as QoS and trust levels.

Conclusions

This article presented an approach to achieve reactive security for SDN/NFV-enabled 5G networks, based on the use of SFC to dynamically chain various security functions, classify traffic, and steer traffic accordingly. Two use cases were analyzed, covering industrial network and ambient-assisted living environments. Moreover, a

proof-of-concept application of this approach was developed, which led to deployment of a reactive security framework modeled on (and deployable to) an actual, operating wind park, allowing continuous monitoring of the industrial network and detailed analysis of potential attacks, thus isolating attackers and enabling the assessment of their level of sophistication (e.g. from script kiddies to state actors). The deployment of this reactive security framework not only enhances the industrial network's security, but also decreases the performance impact of the security functions. The DPI's performance impact is minimized as the traffic only has to go through one DPI instance, and the same can be said for the IDS/IPS functionality, as, e.g. SCADA traffic only has to go through a faster performing, SCADA-specific IDS instance.

As future work, improvements will be investigated in both the security service functions and the implementation of the DPI functionality (essential for traffic-type classification), to minimize the impact of the framework on the network's performance and enable its use in more time-critical industrial applications. Moreover, the framework will be enhanced via the use of an open source NFV MANO software stack, which, via the definition of the service templates at the MANO, will be responsible for the boot-up of the necessary VMs using a Virtual Infrastructure Management (VIM) software (e.g. OpenStack). In turn, the MANO will be used to program the ODL Controller accordingly, passing the necessary information to the SFC Manager. This will also enable a more accurate monitoring of the Service Functions' resources (e.g. allowing the instantiation of additional VMs when one of the existing functions is overloaded).

Acknowledgment

This work has been supported by the European Union's Horizon2020 research and innovation programme project SEMIoTICS with Grant Agreement number 780315.

Related Articles

5G-Core Network Security
5G Enabling Technologies and Autonomic Networking (SDN, MEC, NFV, SFC)
5G Security
5G Security Architecture
5G Security Lifecycle Functions
5G Security Requirements
5G Security Standardization
Access Control to 5G
Network Slicing/Virtualization Security
NFV based security services
Precoding/Beamforming
SDN Security
Security Management in 5G
Security Monitoring and Management

References

Ali, S.T., Sivaraman, V., Radford, A., and Jha, S. (2015). A survey of securing networks using software defined networking. *IEEE Transactions on Reliability*. doi: 10.1109/TR.2015.2421391.

Bhamare, D., Jain, R., Samaka, M. et al. (2016). A survey on service function chaining. *Journal of Network and Computer Applications*. doi: 10.1016/j.jnca.2016.09.001.

Blendin, J., Rückert, J., Leymann, N. et al. (2014). Position paper: software-defined network service chaining. Proceedings – 2014 3rd European Workshop on Software-Defined Networks, EWSDN 2014. doi: 10.1109/EWSDN.2014.14.

Bruschi, R., Bolla, R., Davoli, F. et al. (2019). Mobile edge vertical computing over 5G network sliced infrastructures: an insight into integration approaches. *IEEE Communications Magazine* 57 (7): 78–84.

Chatziadam, P., Askoxylakis, I.G., and Fragkiadakis, A. (2014). A network telescope for early warning intrusion detection. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). doi: 10.1007/978-3-319-07620-1_2.

Chłoń-Domińczak, A., Kotowsa, I., Kurkiewicz, J. et al. (2014). Population ageing in Europe. Facts, implications and policies. *Procedia – Social and Behavioral Sciences*. doi: 10.1016/j.sbspro.2011.05.106.

Department for Homeland Security (2016). Cyber-Attack Against Ukrainian Critical Infrastructure | ICS-CERT. ICS-CERT.

Ersue, M. (2013). ETSI NFV management and orchestration – an overview. Proceedings of 88th IETF Meeting.

ETSI (2012). Network Functions Virtualisation (NFV). ETSI. http://www.etsi.org/technologies-clusters/technologies/nfv (accessed 25 April 2019).

ETSI GS NFV 002 (2014). GS NFV 002 – V1.2.1 – Network Functions Virtualisation (NFV); Architectural Framework, ETSI GS NFV 002.

ETSI GS NFV-SEC 013 (2017). GS NFV-SEC 013 ETSI. Network Functions Virtualisation (NFV) Release 3; Security; Security Management and Monitoring Specification.

Fysarakis, K., Petroulakis, N., Roos, A. et al. (2017). A Reactive Security Framework for operational wind parks using Service Function Chaining. Proceedings – IEEE Symposium on Computers and Communications. doi: 10.1109/ISCC.2017.8024604

Haeffner, W. and Napper J. (2015). Service function chaining use cases in mobile networks. Internet Engineering Task Force.

Halpern, J. and Pignataro, C. (2015). Service function chaining (sfc) architecture. Internet Engineering Task Force.

Kreutz, D., Ramos, F., Verissimo, P. et al. (2015). Software-defined networking: a comprehensive survey. *Proceedings of the IEEE*. doi: 10.1109/JPROC.2014.2371999.

Kumar, S., Tufail, M., Majee, S. et al. (2015). Service function chaining use cases in data centers. IETF SFC WG.

Mahmoodi, T., Kulkarni, V., Kellerer, W. et al. (2016). VirtuWind: virtual and programmable industrial network prototype deployed in operational wind park. *Transactions on Emerging Telecommunications Technologies*. doi: 10.1002/ett.3057.

Mehraghdam, S., Keller, M., and Karl, H. (2014). Specifying and placing chains of virtual network functions. 2014 IEEE 3rd International Conference on Cloud Networking, CloudNet 2014. doi: 10.1109/CloudNet.2014.6968961.

Mendonca, M., Seetharaman, S., and Obraczka, K. (2012). A flexible in-network IP anonymization service. IEEE International Conference on Communications. doi: 10.1109/ICC.2012.6364931.

Migault, D., Pignataro C., Reddy, T. et al. (2016). SFC environment Security requirements. https://tools.ietf.org/html/draft-mglt-sfc/securityenvironment-req-01 (accessed 5 March 2019).

Mimidis, A., Ollora, E., Soler, J. et al. (2018). The next generation platform as a service cloudifying service deployments in telco-operators infrastructure. 2018 25th International Conference on Telecommunications, ICT 2018. doi: 10.1109/ICT.2018.8464838.

Nanda, S., Zafari, F., DeCusatis, C. et al. (2017). Predicting network attack patterns in SDN using machine learning approach. 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2016. doi: 10.1109/NFV-SDN.2016.7919493.

Naudts, B., Kind, M., Westphal, F. J. et al. (2012). Techno-economic analysis of software defined networking as architecture for the virtualization of a mobile network. Proceedings – European Workshop on Software Defined Networks, EWSDN 2012. doi: 10.1109/EWSDN.2012.27.

nDPI (2019). Open and Extensible LGPLv3 Deep Packet Inspection Library. http://www.ntop.org/products/deep-packet-inspection/ndpi/ (accessed 9 February 2019).

ODL wiki (2019). Service function chaining. https://wiki.opendaylight.org/view/Service_Function_Chaining:Main (accessed 12 March 2019).

OECD (2015). Old-age dependency ratio. OECD Pensions at a Glance. doi: 10.1787/pension_glance-2015-23-en.

ONF (2019). Open Networking Foundation. www.opennetworking.org (accessed 15 January 2019).

Open Networking Foundation (2014). SDN architecture overview (HPE). *White Paper*. doi: 10.1017/CBO9781107415324.004.

Open Networking Foundation (2015). L4-L7 service function chaining solution architecture. Technical specification, https://www.opennetworking.org/wp-content/uploads/2014/10/L4-L7_Service_Function_Chaining_Solution_Architecture.pdf (accessed 4 December 2018).

OpenDaylight (2018). Open source SDN platform. www.opendaylight.org (accessed 21 June 2018).

Parada, C., Bonnet, J., Fotopoulou, E. et al. (2018). 5Gtango: a beyond-mano service platform. 2018 European Conference on Networks and Communications, EuCNC 2018. doi: 10.1109/EuCNC.2018.8443232.

Petroulakis, N.E., Fysarakis, K., Askoxylakis, I. et al. (2018). Reactive security for SDN/NFV-enabled industrial networks leveraging service function chaining. *Transactions on Emerging Telecommunications Technologies*. doi: 10.1002/ett.3269.

Prajapati, A., Sakadasariya, A., and Patel, J. (2018). Software defined network: future of networking. Proceedings of the 2nd International Conference on Inventive Systems and Control, ICISC 2018. doi: 10.1109/ICISC.2018.8399028.

Qazi, Z.A., Tu, C.C., Chiang, L. et al. (2016). SIMPLE-fying middlebox policy enforcement using SDN. *ACM SIGCOMM Computer Communication Review*. doi: 10.1145/2534169.2486022.

Quinn, P. (2015). Network service header. *Ietf.* doi: 10.1017/CBO9781107415324.004.

Quinn, P. and Guichard, J. (2014). Service function chaining: creating a service plane via network service headers. *Computer*. doi: 10.1109/MC.2014.328.

Quinn, P. and Tom, N. (2015). Problem Statement for Service Function Chaining. doi: 10.17487/rfc7498.

Rittinghouse, J.W. and Ransome, J.F. (2016). *Cloud Computing: Implementation, Management, and Security*. CRC Press.

SFC Working Group (2019). Charter for working group. https://datatracker.ietf.org/wg/sfc/about (accessed 2 September 2018).

Snort (2018). Network intrusion detection and prevention system. https://www.snort.org (accessed 15 November 2018).

Snort 2.9.2 (2018). SCADA Preprocessors. http://blog.snort.org/2012/01/snort-292-scada-preprocessors.html (accessed 15 November 2018).

Thönes, J. (2015). Microservices. *IEEE Software* 32 (1): 116. doi: 10.1109/MS.2015.11.

Zhang, Y., Beheshti, N., Beliveau, L. et al. (2013). StEERING: A software-defined networking for inline service chaining. Proceedings – International Conference on Network Protocols, ICNP. doi: 10.1109/ICNP.2013.6733615.

Zhang, J., Wang, Z., Ma, N. et al. (2018). Enabling efficient service function chaining by integrating NFV and SDN: architecture, challenges and opportunities. *IEEE Network* 32 (6): 152–159. doi: 10.1109/MNET.2018.1700467.

Zhou, W., Li, L., Luo, M. et al. (2014). REST API design patterns for SDN northbound API. Proceedings – 2014 IEEE 28th International Conference on Advanced Information Networking and Applications Workshops, IEEE WAINA 2014. doi: 10.1109/WAINA.2014.153.

Further Reading

Bhamare, D., Jain, R., Samaka, M., and Erbad, A. (2016). A survey on service function chaining. *Journal of Network and Computer Applications* 75: 138–155.

Bhamare, D., Samaka, M., Erbad, A. et al. (2017). Optimal virtual network function placement in multi-cloud service function chaining architecture. *Computer Communications* 102: 1–16.

ETSI (2014). Network functions virtualisation (NFV) management and orchestration. https://www.etsi.org/deliver/etsi_gs/nfv-man/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf (accessed 4 September 2019).

ETSI (2015). Network functions virtualisation (NFV) infrastructure overview. https://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/001/01.01.01_60/gs_NFV-INF001v010101p.pdf (accessed 4 September 2019).

John, W., Pentikousis, K., Agapiou, G. et al. (2013). Research directions in network service chaining. IEEE SDN for Future Networks and Services (SDN4FNS), Trento, Italy.

Lal, S., Taleb, T., and Dutta, A. (2017). NFV: security threats and best practices. *IEEE Communications Magazine* 55 (8): 211–217.

Li, Y. and Chen, M. (2015). Software-defined network function virtualization: a survey. *IEEE Access* 3: 2542–2553.

Mechtri, M., Ghribi, C., Soualah, O. et al. (2017). NFV orchestration framework addressing SFC challenges. *IEEE Communications Magazine* 55 (6): 16–23.

OSM (2019). OSM PoC 1: devOps in service chains and 5G network slices. https://osm.etsi.org/wikipub/index.php/OSM_PoC_1_-_DevOps_in_Service_Chains_and_5G_Network_Slices (accessed September 2019).

Sendi, A.S., Jarraya, Y., Pourzandi, M. et al. (2016). Efficient provisioning of security service function chaining using network security defense patterns. *IEEE Transactions on Services Computing* 12 (4): 534–549.

Xie, Y., Liu, Z., Wang, S. et al. (2016). Service function chaining resource allocation: a survey. arXiv preprint arXiv:1608.00095.