Towards a 5G Security Architecture: Articulating Software-Defined Security and Security as a Service

Gregory Blanc
SAMOVAR, CNRS, Télécom SudParis
Evry, France
gregory.blanc@telecom-sudparis.eu

Nizar Kheir
Thales Group
Paris, France
nizar.kheir@thalesgroup.com

Dhouha Ayed
Thales Group
Paris, France
dhouha.ayed@thalesgroup.com

Vincent Lefebvre
Tages SAS
Le Cannet, France
vincent@solidshield.com

Edgardo Montes de Oca
Montimage
Paris, France
edgardo.montesdeoca@montimage.com

Pascal Bisson
Thales Group
Paris, France
pascal.bisson@thalesgroup.com

ABSTRACT

5G is envisioned as a transformation of the communications architecture towards multi-tenant, scalable and flexible infrastructure, which heavily relies on virtualised network functions and programmable networks. In particular, orchestration will advance one step further in blending both compute and data resources, usually dedicated to virtualisation technologies, and network resources into so-called slices. Although 5G security is being developed in current working groups, slice security is seldom addressed.

In this work, we propose to integrate security in the slice life cycle, impacting its management and orchestration that relies on the virtualization/softwarisation infrastructure. The proposed security architecture connects the demands specified by the tenants through as-a-service mechanisms with built-in security functions relying on the ability to combine enforcement and monitoring functions within the software-defined network infrastructure. The architecture exhibits desirable properties such as isolating slices down to the hardware resources or monitoring service-level performance.

CCS CONCEPTS

• **Security and privacy → Security services**; **Network security**; **Software and application security**; • **Networks → Network design principles**; • **General and reference** → *Reference works*;

KEYWORDS

Network Slicing, Software-Defined Security, Security as a Service

ACM Reference Format:
Gregory Blanc, Nizar Kheir, Dhouha Ayed, Vincent Lefebvre, Edgardo Montes de Oca, and Pascal Bisson. 2018. Towards a 5G Security Architecture: Articulating Software-Defined Security and Security as a Service. In *ARES 2018: International Conference on Availability, Reliability and Security, August 27–30, 2018, Hamburg, Germany*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3230833.3233251

1 INTRODUCTION

From a business point of view, a network slice is composed of a collection of network functions and services as well as all the processing resources that are necessary for a specific use case or business model [14]. According to 3GPP [4], a network slice instance is a set of network functions that are either physical or virtualized, as well as the resources for these network functions which are arranged and configured, forming a complete logical network to meet certain network characteristics that satisfy specific needs of vertical domains requiring dedicated services. Security has already been considered by 3GPP regarding architecture and procedures [3]. 5G slices embed security functions on a per slice basis as detailed in [4]. Additionally, security should be tuned as per tenant/vertical policies for a given slice which implies in turn to manage/duplicate the security functions on logical slices. A user may access different slices for different services, but the confidentiality, integrity and availability has to be preserved for any slice [8].

Related working groups in 5G-PPP, or projects such as 5G-ENSURE, have started contributing to slicing security by extending what was already proposed in 3GPP. However, we attempt to go beyond the state-of-the-art by integrating security management and orchestration within the life cycle of 5G slices. By doing so, we have the ability to individualize the security of tenants and their services in a multi-tenant, multi-provider infrastructure. It enables fine-grained policies to be deployed and security enablers to collect tenants' needs per application service. Slices are built upon requests that define security function chains. These are fulfilled through, on one hand , the selection of infrastructure resources, and the set up of trusted paths and communications on the other hand. In order to realize autonomic security management, monitoring has to be deployed concurrently to security functions. Incidently, as slices are built upon virtualization and softwarization technologies, security itself highly relies on paradigms such as software-defined networking (SDN) and network function virtualization (NFV) to control the traffic steering throughout the slice, as well as, deploy security functions within the slice, whether they are the result of tenants'

ARES 2018, August 27–30, 2018, Hamburg, Germany
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6448-5/18/08…$15.00
https://doi.org/10.1145/3230833.3233251

demands (*Security as a Service*) or the reaction to security threats (*Software-Defined Security*).

In this paper, we propose such architecture that is able to fully integrate within the slice life cycle. After reviewing the main requirements with respect to 5G slices, we describe the major components of the security architecture, able to address the challenges of multi-tenant, multi-provider infrastructure, per-tenant policy management, and dynamic resource allocation and chaining.

2 REQUIREMENTS

Network slicing relies on network virtualization technologies such as SDN and NFV to isolate multiple tenants over a federation of heterogeneous network domains owned by multiple infrastructure providers. The advent of 5G is characterized by the merge of cloud and network technologies, which extends slicing beyond the network domain and makes the slice a more global service concept that provides a set of processing, storage and network resources in order to respond to specific vertical domains or tenants. The convergence of clouds and communication networks introduces the programmability of the network infrastructure (SDN), and the virtualization of the network functions (NFV) in the slice. However, the general trend of softwarization and the massive usage of virtualization bring intrinsic security issues, in particular software specific. Therefore, a specific attention should be paid to the life cycle of software predominant systems that will also affect security sustainability, including critical update/upgrade phases when the level of vulnerability is typically higher (R1). In addition, the possibility of virtualizing the security functions (i.e., NFV of security) to act themselves on virtualized networks and functions is needed to deal with the dynamism of the infrastructure and the multi-party context (R2). Naturally, a concomitant smart provisioning/orchestration of the security features, both for the enforcement of the security policies and the security monitoring is also required (R3). This virtualization of security functions is aptly named *Software-Defined Security* (SDSec).

The heterogeneity of technologies and involved parties in the provision of a slice leads to a need to maintain consistency in security SLAs and policies across the slice using the *as a service* model (R4). One of the main advantages that the slice management could benefit from the SecaaS (*Security as a Service*) approach is the deployment flexibility, which is more adapted to supporting distributed locations than complex multi-site hardware installations. Migration and dynamic events such as unavailabilities and attacks are also better supported. Finally, SecaaS is an enabler for intelligence-sharing that is important for the enhancement of the security service level, for example by taking protective actions for all tenants as soon as an incident is detected or according to monitored events.

New components introduced by SDN and cloud technologies, such as controllers and orchestrators, become a security concern with regard to the correct management of their integrity, availability, location, access and administration/control (R5).

Another important requirement is the isolation of slices (R6). This implies that:

- the VNFs of a slice are not shared with another slice and are instantiated separately on different virtual machine containers.
- Secure and authenticated links between VNF instances is necessary to ensure an isolation of virtual links connecting the VNFs.
- Obviously, the capability of isolation of the hypervisor, the integrity and trustworthiness of the infrastructure platform are important to be verified prior to the instantiation of a VNF in a slice.
- Critical functions could need hardware isolation, with an instantiation in Hardware-Mediated Execution Enclave (HMEE) for example.

Based on a set of functional and security requirements, Slice Providers provide an end-to-end slice instance as a service that satisfy these requirements and the Security and Service Level Agreements (SLA and SSLA) concluded with the customer (R7). After activating the slice instance, the slice management system has to provide monitoring capabilities to check the satisfiability of the SLA/SSLA (R8), to notify the customer about it (R9), in order to be able to take appropriate actions to guarantee end-to-end security of the slice instance (R10).

3 STATE OF THE ART

5G Architecture and more specifically 5G security architecture is in scope of a number of active working groups ranging from standards developing organizations (e.g. 3GPP[1], ETSI[2]) till 5G PPP[3]. This is especially true for what concerns the 5G security architecture defined by 5G-ENSURE project[4] and further promoted by both 5G Security WG and 5G Architecture WG and which builds on the 3GPP security architecture.

The core of the 5G-ENSURE architecture (see Fig. 1) for 5G networks extends and revises the 3GPP security architecture from TS 33.401 [1] to integrate key features and the *domain* concept from 3GPP TS 23.101 [2] to support trust models for a 5G vision beyond "telecom" and "mobile broadband".

The following domains can be listed:

- Infrastructure domains and tenant domains to capture the physical and logical aspects;
- Management domains to capture orchestration and security management;
- Identity management (IM) domains to re-use existing industrial AAA for device authentication;
- Internet Protocol (IP) domains to model external IP networks;
- *Slice domains* to capture network slicing, application domains transversal to the others.

Focusing on a logical and functional architecture, motivated by general trends such as network de-perimeterisation as well as the strong dependency of 5G systems on software defined networking and virtualization in general, the 5G-ENSURE Security Architecture[15] is one of the first addressing the whole spectrum and requirements to generate trust and confidence in 5G to widen its adoption. Furthermore, not only does it deliver detailed designs but also significantly contributes to its implementation through 5G security enablers specified and/or released in areas of major

[1] The 3rd Generation Partnership Project. http://www.3gpp.org/about-3gpp
[2] The European Telecommunications Standards Institute. http://www.etsi.org/
[3] The 5G Infrastructure Public Private Partnership. https://5g-ppp.eu/
[4] 5G enablers for network and system security and resilience. http://5gensure.eu/

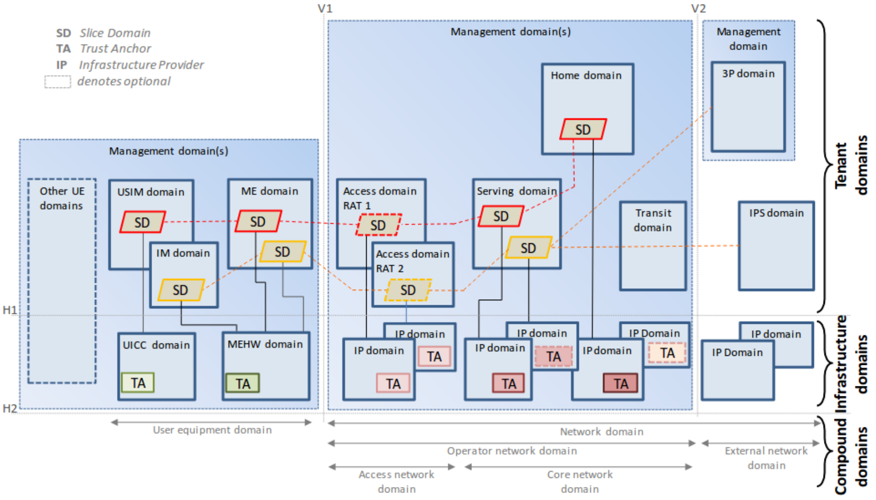

Figure 1: 5G-ENSURE 5G Security Architecture [15]

concerns (i.e., AAA, privacy, trust, security monitoring, network management and virtualization isolation).

Another interesting work is the one done through the Celtic-Plus Project called SENDATE[5] that further promotes concepts of security as a service (SecAAS) and software-defined security (SD-SEC). It has a special focus on slice security which is of utmost importance to satisfy vertical markets' requirements.

Work on-going through 5G-PPP Phase 2 project called SLICENET[6] is also interesting here since it truly targets E2E slicing through a highly innovative slice provisioning, control, management and orchestration framework, oriented to the vertical markets' quality of user experience (QoE).

The vision of network slicing will satisfy the demand of vertical sectors that request dedicated telecommunication services by providing on-demand network slice requirement descriptions to operators as depicted in Figure 2 extracted from 5G-PPP Architecture WG Whitepaper V2.0 [5]. It can also be considered as a reference since it shows how a 5G security architecture is expected to be integrated within the overall 5G architecture, highlighting some of the major components enabling it (e.g., E2E security Service Orchestrator).

4 5G SECURITY ARCHITECTURE

4.1 Architecture

The 5G security architecture aims at securing the 5G slice from the service layer to the network layer, including the management of functions and the orchestration of resources. As described in Fig. 3, it is articulated around three planes, namely the *service* plane, the *management and orchestration* plane and the *infrastructure* plane, which is itself a dual plane, involving both logical and physical functions.

4.1.1 Service Plane. Since our security architecture aims to offer the ability to tenants to describe and enforce their own security

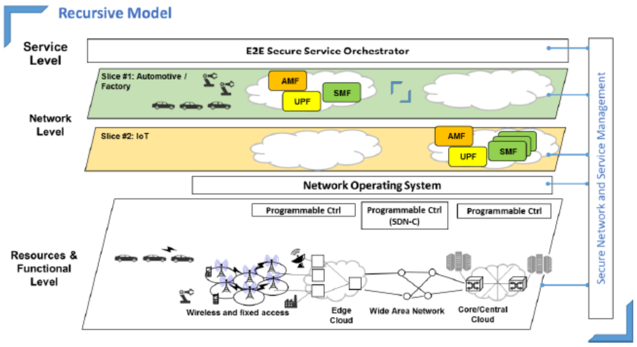
Recursive Model
Service Level
E2E Secure Service Orchestrator
Slice #1: Automotive / Factory
AMF
UPF
SMF
Network Level
Slice #2: IoT
AMF
UPF
SMF
Network Operating System
Programmable Ctrl
Programmable Ctrl (SDN-C)
Programmable Ctrl
Resources & Functional Level
Wireless and fixed access
Edge Cloud
Wide Area Network
Core/Central Cloud
Secure Network and Service Management

Figure 2: 5GPPP Architecture WG's Overall Architecture [5]

policies, this plane exposes applications and interfaces to services enabling them to have control on the slice resources (virtual or physical) they require, and the policies they will deploy in the slice. The SSLA management component will parse the requirements made by the tenant in terms of security. Topology and required services can be derived from the tenant's inputs. The topology can be fed to the cybersecurity monitoring component to initiate a slice instance for the particular tenant, which would allow to monitor the security during the lifetime of the requested services. The services themselves would be placed along paths on the topology.

Alternatively, a tenant can interact with applications that would rely on some services, hitting the exposed APIs. Typically, an Identity and Access Management as a Service (IAMaaS) application would allow the tenant to identify and authenticate to her slice at the service plane, and have subsequent orchestrated services and flows automatically authorized at the control and data plane of the slice.

On the other hand, a service provider could also rely on applications and/or services exposed at the service plane to request

[5] Secure Networking for a Data Center Cloud in Europe. http://www.sendate.eu/
[6] SLICENET. https://5g-ppp.eu/slicenet/

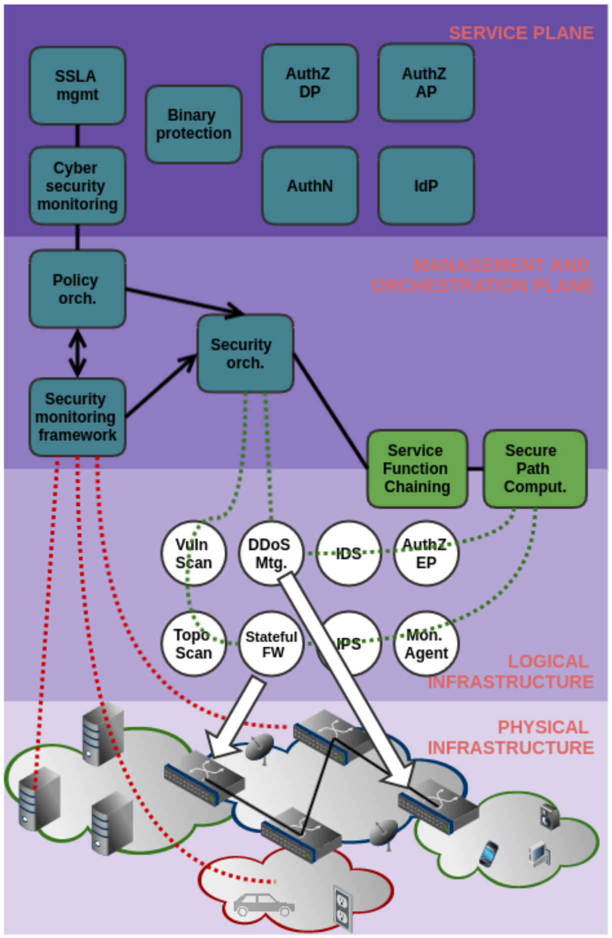

Figure 3: 5G Security Architecture

for the protection of the provider's services. The protection is enforced at the binary level, i.e., when the (virtual) network function is deployed and executed in the slice.

4.1.2 Management and Orchestration Plane. In order to fulfill the demands of tenants, the architecture needs to orchestrate the services according to the tenants' requirements, and guarantee the monitoring and management of the provided resources and functions during the whole life cycle of the slice, and across the providers, along trusted paths. The management and orchestration plane is responsible for computing the sequence of actions to be taken resulting from security policies, in order to provision, create, instantiate, manage, order, maintain, monitor and terminate the resources and functions required to fulfill the services provided in the slice.

The architecture combines a security monitoring framework, to continuously detects the satisfiability of SLA/SSLA across domains, and a security orchestrator that instantiates network security functions in the slice, according to policies and events. The intercommunication between these two components is guaranteed by the policy orchestration component.

4.1.3 Logical and Physical Infrastructures. These infrastructure layers are complementary to the usual slice's control and data planes. They constitute vertical layers that both feature control and data planes. Indeed, within the logical infrastructure, network functions are virtualised and chained, implementing services logically.

Control Plane. Once instantiated, security and monitoring functions need to be deployed to the data plane. The control plane aims at instructing the underlying network on how to deal with the tenants' traffic. As an abstraction plane, the control plane configures the network with (either virtual or physical) functions and build paths to circulate tenants' traffic, distribute policies and virtual machines, effectively executing the requested services.

Data Plane. This plane is the collection of computing, storage and network resources across the providers' cloud and network devices responsible for hosting functions and forwarding traffic.

4.2 Security Orchestration

The resource and service orchestration is a key function in 5G that should be capable of supporting multi-tenancy and network slicing to make possible differentiated services with varying quality of service and security characteristics over the same network infrastructure or even over different network domains. The orchestration needs to manage the complete life cycle of both the slices and the services provided. For this, it needs to have an end-to-end view but also to be able to independently optimize and ensure the security of the slices through specific configurations of the resources, network functions and service chains. In this context, the security orchestrator needs to manage the security of all the slices, VNFs and services, acting as a decision point for reducing security risks and reacting to security breaches.

In our security architecture, mainly two components share these responsibilities. On one hand, a security monitoring framework allows the detection of security events specific to tenants' and clients' requirements (i.e., Security SLAs) and verifies the correct deployment and execution of tenants' policies across the domains. On the other hand, the security orchestrator covers the selection of network security functions, their instantiation in the tenant's slice. In particular, a service function chaining component is responsible for computing the sequence of functions in a cost-effective manner. Additionally, the secure path computation component guarantees that the tenant's traffic in the slice will be secured in compliance with the tenant's requirements and successfully led through the requested security network functions.

Thus, both monitoring and enforcement aspects of security policies need to be simultaneously addressed. The monitoring framework instantiates and deploys monitoring probes to the slices and virtual machines, while the security orchestrator instantiates and deploys reaction policies and functions. To achieve an autonomic security architecture, we combine the monitoring and reaction functions. This is the role of the security policy orchestration component: it interacts with the service plane to collect policy decisions and requirements and organizes two parallel policy streams (*monitoring* and *policy enforcement*). It provides visibility on the monitoring events and the deployed countermeasures from the network to the service plane, and gets updates whenever new policies are pushed, or when the security status of the network changes. Placed at the interface of the monitoring and policy enforcement components, it allows one to react to the other, effectively completing the autonomic security loop.

4.3 Security Service Function Chaining

The placement and chaining of security services in the 5G context add new challenges to the slice operator. These challenges are in part driven by the convergence of IT and networking technologies, where the slice operator leverages the multiple Points of Presence (PoPs) in the 5G network in order to provision and chain security services, in the form of virtual appliances, both while keeping a delicate balance between cost, overhead, quality of service, and security. This is a generic state of the art optimization problem that drives the slice operator to conciliate between the cost of deploying new virtual security functions, the network overhead, the quality of service, and the end-user experience. Nonetheless, in the specific context of provisioning and chaining network security functions, the slice operator needs to handle additional requirements that, to the best of our knowledge, are only partially represented in existing state of the art service chain optimization models.

Security functions add ordering constraints, such as the fact that a firewall service must be supplied in the chain before any subsequent Intrusion Detection Service (IDS), otherwise the IDS would detect attacks that are further blocked by the firewall, leading to an excessive amount of false alerts. Moreover, as long as end users are concerned with their security, they may ask the slice operator to avoid specific PoPs that are operated by non-trusted parties (e.g. providing no guarantees that the data centers are being located within a given geographical area or country). They may even ask the operator not to route their traffic at all through these specific PoPs. Additional security requirements, such as the compatibility between services and the lower security guarantees when hosting them within the same PoP, may also drive the slice operator to take decisions that are not optimal at a first glance. If not correctly represented in the optimization model, these requirements may lead to inconsistent decisions, including excessive costs for the slice operator and a degraded experience for end-users.

In the remainder of this section, we discuss more in details the main challenges and optimization objectives that need to be addressed when provisioning and chaining virtual security functions in the 5G context.

4.3.1 Cost management. The slice operator affords direct costs for the hosting of virtual services over infrastructure datacenters. They include the cost for hardware resources such as storage and computation, and running costs such as maintenance. The slice operator also affords licensing costs for the security editors in charge of editing the virtual security appliances, and that are specified based on multiple pricing models that may be agreed upon between the slice operator and the security editors. Note that these costs are not specific to the security services chaining problem. Rather, they are generic as they apply to other service chaining use cases.

Similarly, we may refer to the costs for the slice operator when routing traffic over a given network link or through a given provider network. The routing costs are due to the different pricing agreements between the slice operator and the 5G network providers. Such agreements may drive the operator to favor specific links or network providers because of their competitive costs.

4.3.2 Network overhead and QoS management. The slice operator aims to reduce network overhead through avoiding unnecessary or excessively long network paths, which may affect the quality of service perceived by end users. In particular, the use of long network paths when chaining network security services may add excessive packet delays. Moreover, overloading network links or exceeding the nominal capacity for a virtual appliance or datacenter may also increase the packet drop rate. These are all important objectives that need to be considered as part of the security service chaining problem.

4.3.3 Security management. The chaining of network security services adds specific challenges and requirements that we classify in this paper into two generic categories: *service-specific* and *user-specific* challenges.

Service-specific requirements. are related to the operation of security services and the way they are chained over the network. First, the *ordering requirement* sets a partially ordered relation over the set of security services. One example is the chaining of a proxy encryption function (`ENC`) and a deep packet inspection function (`DPI`). The `DPI` function requires direct access to the traffic content, whereas the `ENC` function encrypts the content to make it unaccessible over the network. These functions are chained together only in the exact order where the `DPI` function is supplied in the chain *before* the `ENC` function.

Second, *compatibility constraints* prevent security functions from being collocated within a same datacenter. Certain functions may provide less security guarantees when being collocated within the same platform, as they expose the traffic to specific attacks like side-channels and memory leakage. This is a fundamental requirement, especially when the functions are being supplied by security editors that are not equally trusted by the end-user.

User-specific requirements. enable users to obtain custom security guarantees. In this scope, *trust constraints* may drive the slice operator into selecting or avoiding specific network providers or specific PoPs. The main reason is that the provider may be submitted to different regulations (e.g. a network provider operating datacenters across multiple countries). Hence, the users may constrain the slice operator into hosting their security functions and / or routing their traffic within a given geographical area or inside a given country.

Finally, *isolation constraints* enable a group of users to prevent their respective functions from being collocated within the same datacenters, and / or their traffic from being routed across the same network links. The isolation constraints formulate to some extent a *slicing policy* that is particularly relevant in the context of a slice operator, and that may capture security requirements for sensitive users like defence and critical service providers such as electricity and public transport. In particular, they enable such sensitive users to specify flows that are either logically or physically isolated from other user flows that are being managed by the operator. As long as these users may need such high level security guarantees, they may constrain the operator into using dedicated data centers and network PoPs to host their supplied cyber security functions.

4.3.4 Dynamicity management. A key requirement when provisioning and chaining network security functions is the ability for the operator to optimize on the fly new security function chains, or yet to modify an existing chain of security functions, while minimizing the impact on previously installed chains. This is driven by the fact that there may not be a one-size-fits-all security configuration, and the slice operator may need to constantly update or modify the service function chains in order to adapt to new security demands (a.k.a. security as-a-service), or yet to face new security threats (a.k.a. software-defined security).

Two different alternatives may be adopted by the slice operator. It may either choose to optimize from scratch a whole new optimal strategy, even though the new computed strategy may modify the already existing service chains. Alternatively, it may avoid changing the existing chains in order to minimize the impact on end-user experience, and find a *local optimum* for the new function chains. While the second strategy seems more appropriate, it cannot guarantee an optimal chaining on the long term, as long as new function chains will need to be optimized. To address this issue, the security function chaining requires the ability to incrementally optimize new security chains, taking into account the service chains already implemented by the operator. The objective would be to minimize the cost and network overhead for all security chains, while also minimizing the changes for the already existing ones.

4.4 Secure Virtual Network Embedding

Slice operators, regardless of whether they are the slice tenants or not, are responsible for embedding the service slice on top of a substrate network. The substrate network is often a composition of multiple provider networks, with a different set of resources and constraints. The embedding often consists in two distinct phases: (a) mapping: in which a solution to a virtual network request, including both physical network nodes and links, is computed to satisfy the tenant's needs. *This phase finds all possible solutions satisfying the tenant's needs*; (b) allocation: in which the computed mapping is enforced on top of the selected resources, under the providers' constraints. *This phase finds the optimal solutions satisfying the providers' policies.* Indeed, these two phases are not usually distinct but such partition allows to simplify the problem of virtual network embedding (VNE).

4.4.1 Mapping. As described above, a tenant usually describes in a service request the functions necessary to process a specific traffic flow. Therefore, there is often as many slices as there are specific flow treatments, with the deployment of the required functions on the path, as made possible by the network function virtualization (NFV) paradigm. Such demands actually induce consistent resource provisioning in terms of compute, storage and network. VNE itself is interested on how a virtual network is provisioned to support the service slice. The tenant, being more and more concerned about security issues related to service provisioning and execution, tends to express more fine-grained needs with regard to security. Legacy embedding approaches were simply concerned with resource optimization, which was often transparent to the tenant, inducing over-provisioning among providers. With security in mind, additional constraints apply, and the mapping phase becomes explicit: selection of nodes and paths that will satisfy properties of confidentiality, integrity, availability, etc.

Tenant constraints. In order for a slice operator to negotiate over such constraints with the multiple infrastructure providers, the mapping request should be expressed in a unified and actionable way. Given the virtual network can be represented as a graph, an attribute-based graph could fulfill these goals, as demonstrated in [7]. In particular, different classes of attributes are considered as some attributes may (a) be single-valued or multi-valued, (b) vary over time while others remain constant, (c) have impact on the embedding or be innocuous.

Provider privacy. An additional benefit of the approach described in [7] is that the providers do not need to disclose their topology. The slice operator only needs to know the interdomain, i.e., how the different provider networks are interconnected. The resolution of the mapping problem is then a 2-stage problem, with a parallel resolution of the mapping request by individual providers and a composition of obtained individual "pieces" by the slice operator.

4.4.2 Allocation. Resource allocation is a widely studied topic in machine virtualization, and has also been actively investigated in network virtualization, in particular with regard to NFV [10]. The heterogeneity of resources to allocate (compute, storage, network) and their dynamics make their allocation quite challenging. This accounts for the fact that state-of-the-art approaches are mostly static. The slice is made of not only network resources, i.e., forwarding elements and paths, but also of computing resources that support the services the slice is dedicated to. Such computing resources may be placed at diverse locations depending on the computation model (mobile edge computing, fog computing, cloud computing, etc.). This highlights the fact that the slice operator has indeed to accommodate conflicting requirements, with the service owner needs on one hand, and the infrastructure owners constraints on the other hand. Once the slice request, the virtual network one, has been formulated and satisfied, it can only be completely fulfilled when resources are allocated for enforcing the deployment.

In order to make current resource allocation strategies dynamic, the state of the substrate network and cloud network infrastructures need to be continuously monitored. As embedding requests are limited in time, it is also important to estimate resources with respect to their sunset date. Some efforts have already been carried out towards machine-learning based optimization of resource allocation of VNF [12].

Regarding location of resources, the environment being not only multi-provider, but also multi-tenant, resource allocation needs to consider possible mutualisation through traffic aggregation, eventually leading to migrating network functions or service functions. This advocates the fact that the substrate network owned by the infrastructure owners is indeed a moving target, which should not affect the slice in any way. In particular, tenants collocated on the infrastructure network may also have conflicting or competing policies, requiring policy reasoning at a high level and appropriate translation to network configuration and functions.

These policies and functions encompass security and monitoring ones, which can also be resource-intensive. On one hand, continuous monitoring addresses network state updates, and anomaly detection. On the other hand, security functions may be mutualised and placed at aggregation points, depending on the sensitivity of the processing or of the processed data. A first approach towards modeling overlapping areas of security coverage was proposed in [6] where a network function's configuration enabled the definition of a "responsibility domain". Learning these domains could help optimizing the coverage of the attack surface by security functions, without overspending resources.

4.5 Virtual Network Functions Security

VNFs are operated within the boundaries of a slice, which is itself defined by its PoPs, geographical limitations (area exclusions), QoS and security attributes, such as isolation. On one hand, slice operators shall operate the slice within its boundaries, at minimum costs. On the other hand, VNF vendors shall duly profit from the usage of their VNFs by their customers within the slice. Ensuring the security of VNFs shall benefit to both slice operators and VNF vendors, guaranteeing the operation of VNFs within the slice boundaries at the lowest cost, while protecting the VNF vendors business interests.

VNFs are indeed victim to a number of threats:

- when slices are operated off-premises, a VNF may be locally attacked by a maintenance operator. This so called *introspection* risk exposes the VNF to be analyzed and modified if no technical measures are implemented. Additionally, side-channel attacks can also be carried out locally against data in transit, to break confidentiality, but rarely impact the VNF code itself, i.e., the software.
- when a VNF expects data inputs from users, remote attacks can be mounted that exploit vulnerabilities or misconfigurations. Typical instances include buffer overflows and denial of service attacks by resource exhaustion.
- when a VNF is distributed as a VM payload, there is a risk that is being abused by operators or even competitors beyond the limits of the license rights. Such payload can be copied and used elsewhere, especially when the software is no longer bound to run on a specific hardware platform.

  These three VNF threats can today and will be tomorrow hold back by various techniques providing confidentiality, integrity or right enforcement

4.5.1 VNF Integrity. Current state of the art is based on software file integrity verification, i.e., the software executable is verified to be the original only at rest, before being loaded by the processor in RAM, then executed. Hence, this VNF file authentication does not prevent local attackers to alter software in memory by introspection attacks. To secure cloud computing, processor vendors have developed techniques to ensure data and code integrity even when executed in so-called adversary conditions (malicious operator or kernel). Techniques, such as the Trusted Execution Environment (TEE), are in fact protected memory areas that are only accessible by the program itself (even debuggers are excluded). Thus, a TEE guarantees integrity to any code that it executes. However, some recent publications have demonstrated that the TEE is no silver bullet [9, 11, 13, 16] but they still offer a security level that software solution could not bring. They are today just starting their life cycle and will undoubtedly evolve and improve. Intel's SGX version 2.0 has for instance just brought software confidentiality which was missing in version 1.0. However, their adoption by the VNF and telecom market in general faces a strong obstacle, which is the processor market fragmentation. As a matter of fact, investing on one TEE technology to build NFV security requires the assurance that this type of TEE will be available on the cloud servers that will run the NFV (which can be seen as a bet for the future by the NFV vendor). Therefore, there is a need to bridge all different types of TEEs to break this strong resistance.

Another route for bringing code integrity at runtime is offered by the insertion of self-checks. A change in the program footprint content results either in a crash or a graceful program exit. These self-checks are no silver bulletand can be circumvented by talented attackers. However, they do not impact performance so they can be massively inserted. Self-checks bring an applicable, processor-agnostic defense against software modification attacks made on processor working memory.

4.5.2 VNF Confidentiality and Right Enforcement. Encrypting a VNF addresses confidentiality for a static analysis but it cannot prevent tracing the code with a debugger if this is made possible for the attacker. Actually, she will trace the code execution until it appears in clear in memory since encrypted code cannot be directly executed by the processor. An additional level of protection is code obfuscation which results in more complex to analyze and slower to execute software. Performance degradation is inevitably related to the strength of the obfuscation.

It is possible to attach the code with a hardware anchor, present where it is executed. Even for VNFs executed in a Virtual Machines, it is possible to attach each of these VNF to a single VNF-launch token hardware device, plugged somewhere in the data centre. In practice, this attachment is a strong obstruction to the attack which would have been done much more comfortably at attacker's home. Alternatively, One can imagine a remote or split code solution not yet part of the state of the art marketed solutions. By use of code interpretation technique, which in practice transforms machine code instructions into a hardware abstract "payload" (e.g., a data set) executed through an interpreter, one can build a split or remote partial execution scheme providing both code integrity and confidentiality. The code executed remotely through interpretation cannot be analyzed since the attacker does not have read and write access to it. Code right enforcement, intellectual property and protection against cyber attack can be met with the highest security by moving sensitive parts of the software in a fully-controlled machine, which can be either a locally-based closed machine or an open machine located in a safe, remote location. One could argue that this technique opposes to cloud computing principle, as well as it may degrade the software performance and last but not least is dependant on the internet connexion quality. But the remotely executed piece of code actually represents a small part of the complete software and is run on a proprietary location, which could be the private part of a hybrid cloud, leaving the vast majority of the software to be still run into the cloud. Second, the delay induced by code interpretation could actually be acceptable in comparison to the latency induced by the communication between the two machines (local and remote interpreters). Last, as introduced previously, we consider that TEE of different types shall be bridged to develop a TEE-agnostic solution that would operate on any different types of TEE. The solution would be the ultimate security solution offering altogether confidentiality, integrity and right management with encompassed strength with the required deploy ability for NFV operation. Compared to the code split solution, a TEE-agnostic solution is not dependant on internet quality and is shows a better deploy ability, whereas it offers the same level of security.

4.5.3 Recommended VNF security. Code confidentiality, code integrity and right enforcement techniques shall all be applied on the VNF. Our survey shows that these techniques are not independent but reversely technically intricated. They are combined and grouped inside today's marketed or future security solutions. One shall keep in mind that each of them can be offered with high variations in terms of resilience. The techniques should not be viewed as all equivalent. Some vendor security promises are fragile. Only a technical analysis of what a technique does on the code provides a first appreciation of its real resilience strength. Security level is not the only aspect to consider. Techniques vary also with their performance impact and their deployability.

When considering the best technique to cope with VNF introspection, one shall first defines the security threat (modus operandi of the attack, methodology and tools, objectives) as well as NFV market main criteria (e.g.,throughput, deployability). It is more practical to hinder the attacker, i.e., making the attack more complex to carry out, than to employ the most robust techniques that could slow down the VNF. Considering practical efficiency, code machine binding technique could be preferred over sophisticated code obfuscation.

TEE is expected to be an important part of NFV security as it allows to combine seemingly opposing aspects: high efficiency and low performance impact. Offering the most efficient security with minimal impact on performance is considered a breakthrough in software security. However, deployability is questionable with the fragmentation of the current processor vendor market. Until TEE-based security solution deployability is addressed (if ever), we would recommend a security solution that use encryption for code confidentiality, self-checks insertions for runtime code integrity and last but not least, machine binding to obstruct code dump and code rights infringement. This choice is motivated by the VNF market where high throughput is favored over resource-intensive obfuscation techniques. With the suggested solution, we concede that the attacker can still perform dynamic code tracing using a debugger. Binding the code on the machine is however a serious obstacle to the attacker, forcing her to trace the code on-site, i.e., directly on the hardware, which is impractical when such hardware is housed at physically protected data centers. Additionally, once bound to the machine, the code can not be abused and the VNF vendor intellectual property is guaranteed to be secure. Last, inserted self-checks will significantly augment the efforts for the attacker to modify the code.

5 CONCLUSION

This paper has introduced an integrated security architecture to support the slice security within its life cycle. In particular, it articulates around the 5G slice planes and exposes interfaces for tenants to specify their security needs (Security as a Service). It relies on a software-defined networking and monitoring infrastructure to build security functions into the slice (Software-defined Security).

Through its different building blocks, the 5G security architecture we presented is able to address a number of requirements we described, including the security of VNFs (R1), the integration of security functions in the slice (R2), the composition of enforcement and monitoring functions (R3), the on-demand security provisioning per tenant's needs (R4/R7), the security-aware virtual network embedding providing isolation of slices (R6), and the global security orchestration (R8-R10).

Future work includes the definition of metrics and a methodology to evaluate the infrastructure. In particular, cross-layer interfaces would rely on existing virtual infrastructure management as well as management and orchestration ones, with the challenge of effectively bridging network and compute resource control with network control.

ACKNOWLEDGMENTS

The work is supported by the Celtic-Plus SENDATE-TANDEM project (2016-2019).

REFERENCES

[1] 3GPP. [n. d.]. *3GPP System Architecture Evolution (SAE); Security architecture.* Technical Report 33.401.

[2] 3GPP. [n. d.]. *General Universal Mobile Telecommunications System (UMTS) architecture.* Technical Report 23.101.

[3] 3GPP. [n. d.]. *Security architecture and procedures for 5G System.* Technical Report 33.501.

[4] 3GPP. [n. d.]. *System Architecture for the 5G System Stage 2.* Technical Report 23.501.

[5] 5G-PPP Architecture Working Group. 2017. *View of 5G Architecture.* Technical Report. Version 2.0. https://5g-ppp.eu/wp-content/uploads/2018/01/5G-PPP-5G-Architecture-White-Paper-Jan-2018-v2.0.pdf.

[6] Y. Ben Mustapha, H. Debar, and G. Blanc. 2014. Policy enforcement point model. In *International Conference on Security and Privacy in Communication Systems.* Springer, 278–286.

[7] F. Boutigny, S. Betgé-Brezetz, H. Debar, G. Blanc, A. Lavignotte, and I. Popescu. 2018. Multi-Provider Secure Virtual Network Embedding. In *9th IFIP International Conference on New Technologies, Mobility and Security (NTMS).* IEEE, 1–5.

[8] E. Dotaro. 2018. 5G Network Slicing and Security. *IEEE Softwarization* January (2018).

[9] J. Götzfried, M. Eckert, S. Schinzel, and T. Müller. 2017. Cache Attacks on Intel SGX. In *10th European Workshop on Systems Security (EUROSEC).* ACM, 2:1–2:6.

[10] J.G. Herrera and J.F. Botero. 2016. Resource allocation in NFV: A comprehensive survey. *IEEE Transactions on Network and Service Management* 13, 3 (2016), 518–532.

[11] J. Hiser, A. Nguyen-Tuong, M. Co, M. Hall, and J.W. Davidson. 2012. ILR: Where'd my Gadgets Go?. In *IEEE Symposium on Security and Privacy (S&P).* IEEE, 571–585.

[12] H. Jmila, M. Ibn Khedher, and M.A. El Yacoubi. 2017. Estimating VNF Resource Requirements Using Machine Learning Techniques. In *International Conference on Neural Information Processing.* Springer, 883–892.

[13] A. Moghimi, G. Irazoqui, and T. Eisenbarth. 2017. Cachezoom: How SGX Amplifies the Power of Cache Attacks. In *19th International Conference (CHES).* 69–90.

[14] NGMN Alliance. 2016. *Description of Network Slicing Concept.* Technical Report.

[15] R. Blom et al. 2017. *Security Architecture (Final) .* Technical Report D2.7. 5G-ENSURE.

[16] M. Shih, S. Lee, T. Kim, and M. Peinado. 2017. T-SGX: Eradicating Controlled-Channel Attacks against Enclave Programs. In *24th Annual Network and Distributed System Security Symposium (NDSS).* The Internet Society.