Modeling and Mitigating Security Threats in Network Functions Virtualization (NFV)

Nawaf Alhebaishi[1,2](✉), Lingyu Wang[1], and Sushil Jajodia[3]

[1] Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada
{n_alheb,wang}@ciise.concordia.ca

[2] Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

[3] Center for Secure Information Systems, George Mason University, Fairfax, US
jajodia@gmu.edu

Abstract. By virtualizing proprietary hardware networking devices, Network Functions Virtualization (NFV) allows agile and cost-effective deployment of diverse network services for multiple tenants on top of the same physical infrastructure. As NFV relies on virtualization, and as an NFV stack typically involves several levels of abstraction and multiple co-resident tenants, this new technology also unavoidably leads to new security threats. In this paper, we take the first step toward modeling and mitigating security threats unique to NFV. Specifically, we model both cross-layer and co-residency attacks on the NFV stack. Additionally, we mitigate such threats through optimizing the virtual machine (VM) placement with respect to given constraints. The simulation results demonstrate the effectiveness of our solution.

1 Introduction

As a cornerstone of cloud computing, virtualization has enabled providers to deliver various cloud services to different tenants using shared resources in a cost-efficient way. The trend of virtualization has also led to many innovations in networking in and outside clouds. In particular, traditional networks heavily rely on vendor specific hardware devices with integrated software, such as routers, switches, firewalls, IDSs, etc., which lacks sufficient flexibility demanded by today's businesses. Consequently, the need for decoupling software from hardware in network devices has led to Network Functions Virtualization (NFV) [14], which basically virtualizes proprietary hardware networking devices. As a key enabling technology of 5G, NFV has seen an increased adoption among cloud service providers, especially in the telecommunication industry [22].

However, the reliance on virtualization and the increased complexity together imply that NFV may unavoidably introduce new security concerns. First, as an NFV stack involves several abstraction levels covering the physical and virtual

A. Singhal and J. Vaidya (Eds.): DBSec 2020, LNCS 12122, pp. 3–23, 2020.
https://doi.org/10.1007/978-3-030-49669-2_1

infrastructures as well as the virtual network functions [14], it naturally has a larger attack surface, opening doors to new security threats such as cross-layer attacks. Second, as one of the main advantages of NFV is to provide diverse network services to different tenants using the same hardware infrastructure, NFV would also share similar cross-tenant attacks as those seen in clouds (e.g., [44]). Therefore, security threats introduced by the multi-layer and multi-tenant nature of NFV need to be better understood and mitigated.

Attack modeling and mitigation in NFV has only received limited attention (a more detailed review of related work will be given in Sect. 6). Existing works have focused on specific vulnerabilities caused by orchestration and management complexities [43] and vulnerabilities resulting from the lack of interoperability [15] or the lack of proper synchronization between different abstraction levels [24]. There also exist works on dynamically managing security functions in NFV [41] and verifying Service Function Chaining (SFC)-related properties [17,28,38,45]. Existing works on co-residency attacks mostly focus on clouds [4,6] instead of NFV. To the best of our knowledge, there lack a general approach to modeling and mitigating NFV attacks.

In this paper, we take the first step toward modeling and mitigating security attacks in NFV. Our key ideas are threefold. First, we propose a multi-layer resource graph model for NFV in order to capture the co-existence of network services, VMs, and physical resources at different abstraction levels inside an NFV stack, and how those could potentially be exploited by attackers. This model allows us to capture not only attacks that target each layer of the NFV stack, but also attacks that go across different layers by exploiting the inter-dependencies between corresponding resources at those layers. Second, we also model the insider threats posed by malicious or compromised users of co-resident tenants inside the same NFV stack. The model allows us to captures how a co-residency attack may allow insiders to satisfy certain initial security conditions, such as privileges or connectivity, which are normally not accessible to external attackers. Third, we propose a solution to mitigate security attacks in NFV through VM placement and migration, which is a low cost option already available in NFV. The aforementioned model allows us to formulate the attack mitigation in NFV as an optimization problem and solve it using standard optimization techniques. We evaluate our approach through simulations to demonstrate its effectiveness under various situations. In summary, the main contribution of this paper is twofold:

- To the best of our knowledge, this is the first study on the modeling of security threats in an NFV stack. Our multi-layer resource graph model demonstrates the possibility of novel security threats in NFV, such as cross-layer and co-residency attacks, and also provides a systematic way to capture and quantify such threats.
- By formulating the optimization problem of mitigating attacks on NFV through optimal VM placement and migration, we provide an effective solution, as evidenced by our simulation results, for achieving a better trade-off between security and other constraints using standard optimization techniques.

The remainder of this paper is organized as follows; Sect. 2 presents background information on NFV and co-residency in NFV, and provides a motivating example. In Sect. 3, we present our multi-layer resource graph and insider attack model, and describe the application of a security metric. Section 4 formulates the optimization problem and discusses several use cases. Section 5 gives the simulation results. Section 6 reviews related works. Section 7 concludes the paper.

2 Preliminaries

This section first provides background information on NFV and co-residency in NFV, and then gives a motivating example.

2.1 Background on NFV

As a main technology pillar of network softwarization and 5G, NFV provides network functions through software running on standard hardware. NFV enables network service providers to deploy dynamic, agile and scalable Network Services (NS). Such benefits come from the fact that an NFV deployment stack is usually an integration of various virtualization technologies including cloud and SDN together with various network orchestration and automation tools.

Specifically, the left side of Fig. 1 shows the European Telecommunications Standards Institute (ETSI) NFV reference architecture [14]. The architecture builds on three main blocks, i.e., virtual network function (VNF), NFV infrastructure (NFVI), and NFV management and orchestration (MANO). First, the VNF provides network functions, such as router, switch, firewall, and load balancer, running on top of VMs through software. Second, NFVI represents the cloud infrastructure that provides basic computations, network, and storage needs for the execution of VNFs. Lastly, MANO has three management components, virtual infrastructure manager (VIM), virtual Network function manager (VNFM), and network function virtualization orchestrator (NFVO), which together manage and orchestrate the lifecycle of physical and virtual resources.

In addition, the right side of Fig. 1 shows a multi-layer NFV deployment model [24] which complements the aforementioned ETSI architecture with deployment details, and divides an NFV stack into four layers, i.e., service orchestration (layer 1), resource management (layer 2), virtual infrastructure (layer 3), and physical infrastructure (layer 4).

2.2 Co-Residency in NFV

As an NFV stack is multi-layer and multi-tenant in nature, placing and migrating a VM or VNF can be a challenging task for the provider due to the issue of co-residency. It is well known that co-residency may lead to various security issues, such as side-channel attacks, and additionally the tenant may have specific requirements in terms of (the lack of) co-residency. Co-residency may occur in an NFV environment when a new VM or VNF is first placed, or when an

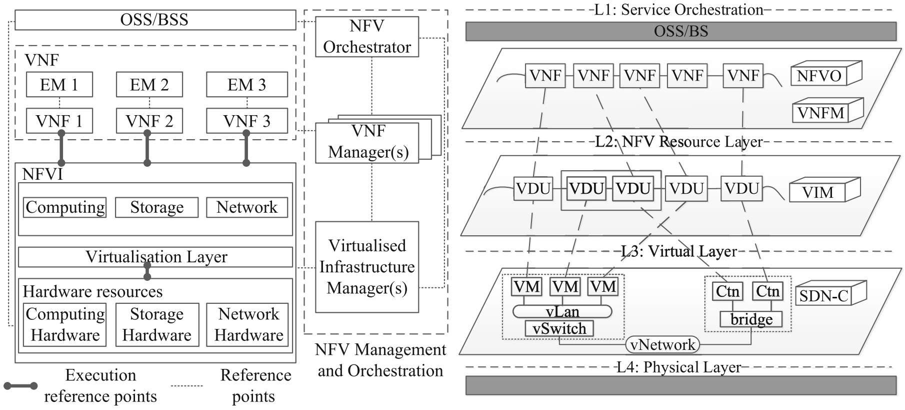

Fig. 1. ETSI reference architecture [14] (left) and multi-layer NFV model [24] (right)

existing VM or VNF is migrated. The tenant requirements may specify that certain VNFs are to be placed on a dedicated host, or a VM needs to have the auto-scaling feature such that its need for more space can be quickly fulfilled. In terms of security, co-resident VMs or VNFs may belong to tenants with conflicting interests, and the co-residency may enable an insider attack with increased privileges and connectivity not available to regular attackers.

A unique aspect of co-residency in NFV is that, in an NFV stack, co-residency can happen between more layers, such as between VNFs and physical hosts, between VNFs and VMs, or between VMs and physical hosts. The co-residency of VNFs or VMs on the same physical host can occur due to placement or migration, which is known to lead to side-channel or resource depletion attacks due to the shared physical resources such as CPU, memory, or cache. The co-residency of VNFs on the same VM can also occur when different tenants employ the same VM to run similar network functions, such as virtual firewall or virtual IDS [21,23,32].

2.3 Motivating Example

In the following, we present a concrete example of NFV stack to demonstrate the challenges in modeling and mitigating security threats for NFV. First of all, as NFV is a relatively new concept, there lack public access to information regarding the detailed hardware and software configurations used in real NFV environments. As can be seen in Fig. 1, both the ETSI architecture [14] and the multi-layer deployment model [24] are quite high level, and lack such details. Most other existing works either focus on high-level frameworks and guidelines for risk and impact assessment [26,29,33], or very specific vulnerabilities [25,30,31], with a clear gap in between.

To address such limitations, we design a concrete example of NFV stack, as shown in Fig. 2, based on both the ETSI architecture [14] and the multi-layer

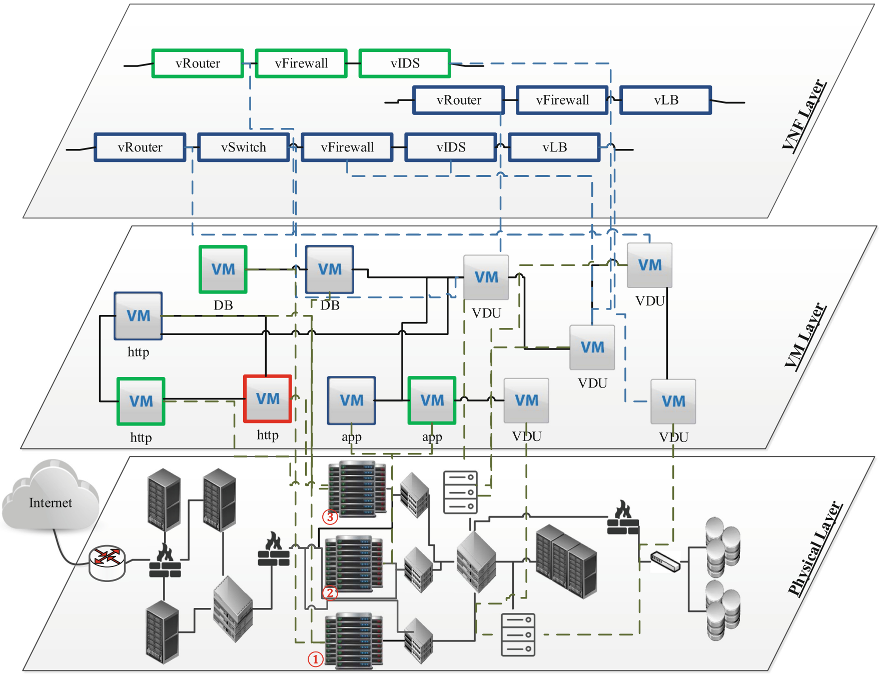

Fig. 2. A concrete example of NFV stack (vFirewall: virtual firewall, LB: load balancer, DB: database, VM: virtual machine, VDU: virtual deployment unit)

deployment model [24], as well as other public available information gathered from various providers and vendors. As shown in Fig. 2, the NFV stack is depicted on three layers where the VNF layer shows three service function chains (SFCs), and the VM and physical layers show the corresponding virtual and physical infrastructures used to implement those SFCs, respectively. The dashed lines between layers demonstrate the correspondence between the services and the virtual or physical resources. We assume there are three tenants, shown in Fig. 2 through different colors, i.e., Alice (blue), Bob (green), and Mallory (red)), that are hosted on this NFV stack.

In such a scenario, both the NFV provider and each tenant may want to understand and mitigate potential security threats. While existing threat models such as various attack trees and attack graphs may be applied, there are some unique challenges and opportunities as follows.

- First, as can be seen in Fig. 2, the NFV stack is composed of different layers, and the inter-dependencies between resources on those layers may lead to novel cross-layer attacks. The NFV tenant and provider need to consider all layers and the inter-dependencies between layers when analyzing potential security threats because of the possibility of such cross-layer attacks.

- Second, the fact that multiple tenants are sharing both virtual and physical resources in the same NFV stack poses another challenge, i.e., co-residency attacks. In contrast to clouds, NFV may have an increased attack surface in terms of co-residency attacks. As demonstrated in Fig. 2, unlike in clouds, co-residency in NFV may occur in terms of both shared physical resources and shared virtual resources such as VMs, which must be considered in modeling the threat of co-residency attacks.
- On the other hand, virtualization in NFV also provides an opportunity for mitigating security threats through a unique hardening option, i.e., through VM placement or migration. In contrast to other hardening options, such as patching vulnerabilities, disabling services, or stricter firewall and access control rules [8], VM placement or migration provides a lower cost option as it is a built-in feature already employed by providers for other purposes such as maintenance or resource consolidation.

More specifically, we consider concrete problems of threat modeling and attack mitigation based on Fig. 2 as follows.

- First, we would like to model potential multi-step attacks that could occur in this NFV stack (Fig. 2), by assuming Mallory (whose resources are shown in red color) is a malicious tenant, and the database VM belonging to Alice (whose resources are shown in blue color) is the critical asset in question. The modeling process must consider the multi-layer and multi-tenant nature of NFV and its many security implications, e.g., an attacker's VM placed on the database server (node # 1) or on the *http* server (node # 3) would certainly incur very different security threats, and an attacker co-residing with the target tenant on the VM level could have a better chance of compromising the target than one co-residing on the physical host level.
- Second, we would like to mitigate the modeled threats posed by Mallory to Alice's database VM, through optimal placement or migration of virtual resources in this NFV stack. The solution must quantify the security threats before and after the hardening process in order to show the amount of improvement in terms of security, and the solution should be able to accommodate other considerations or constraints, e.g., limiting the scope to one layer or multiple layers, supporting different VM placement policies (such as those used in CloudSim [11]), and limiting the cost of placement or migration (such as the maximum number of VM migrations).

To this end, we will present our threat modeling solution in Sect. 3 and our attack mitigation solution in Sect. 4.

3 Modeling Security Threats in NFV

This section presents our solutions for modeling potential multi-step attacks on an NFV stack (Sect. 3.1), for modeling insider attacks from co-residing tenants (Sect. 3.2), and for quantifying the threats using a security metric (Sect. 3.3).

3.1 Multi-layer Resource Graph

Threat Model: Our goal is to help the NFV provider or tenants to better understand and mitigate the security threats from an external attacker, a dishonest or compromised user or tenant, or a tenant administrator or cloud operator with limited privileges. We assume the NFV provider and its administrators are trusted and consequently the inputs to our threat modeling process are intact. Our in-scope threats are security attacks used to escalate privileges or gain remote accesses through exploiting known or zero day vulnerabilities in the physical or virtual entities inside an NFV stack. Out-of-scope threats include attacks which do not involve exploiting vulnerabilities (e.g., phishing or social engineering attacks) or do not propagate through networks (e.g., flash drive-based malware).

To model the security threats in an NFV stack, our key idea is to apply the resource graph concept [34] (which is syntactically similar to attack graphs, but focuses on modeling zero day attacks exploiting unknown vulnerabilities) while considering the multi-layer nature of NFV (as explained in Sect. 2.1). Specifically, based on a model of the NFV stack with three layers, i.e., the VNF layer, VM layer, and physical layer, as previously demonstrated in Fig. 2, we propose the concept of *cross-layer resource graph* to represent the causal relationships between different resources both inside each layer, and across different layers, in a given NFV stack.

We first illustrate the concept through an example, before giving the formal definitions. Figure 3 shows an example of a cross-layer resource graph for our running example (only a portion of the cross-layer resource graph is shown here due to space limitations). The VNF layer maps to layer 1 and layer 2 in the multi-layer NFV deployment model [24], which depicts exploits of various virtual network functions, such as virtual firewalls, load-balancers, switches, etc. The VM layer corresponds to layer 3 in the NFV deployment model, which includes exploits of the VMs that are used to implement the virtual network functions in the cloud layer. Finally, the physical layer includes exploits of the physical hosts. The left-hand side of Fig. 3 shows the cross-layer resource graph for one tenant, whereas the right-hand side shows co-residency attacks from other tenants (which will be explained in next section).

Each triple inside an oval indicates a potential zero day exploit (in which case the unknown vulnerability is represented by the exploited service itself) or an exploit of known vulnerabilities, in the form of <service or vulnerability, source host, destination host>. For example, <Xen, 3, h3> indicates an exploit of Xen coming from a VM on physical host 3 to that physical host itself, and the plaintext pairs indicate the pre- or post-conditions of those exploits in the form of <condition, host>, where a condition can be a privilege on the host, e.g., <root,3> means that the root privilege on the VM runs on host 3, and <user,vFW> means the user privilege is on the VNF layer for the virtual firewall. Additionally, conditions may include the existence of a service on the host (e.g., <Xen,3>), or a connectivity (e.g., <0,3> means that attackers can connect to a VM on host 3 from host 0, and <2,h2> means a local exploit is occurring

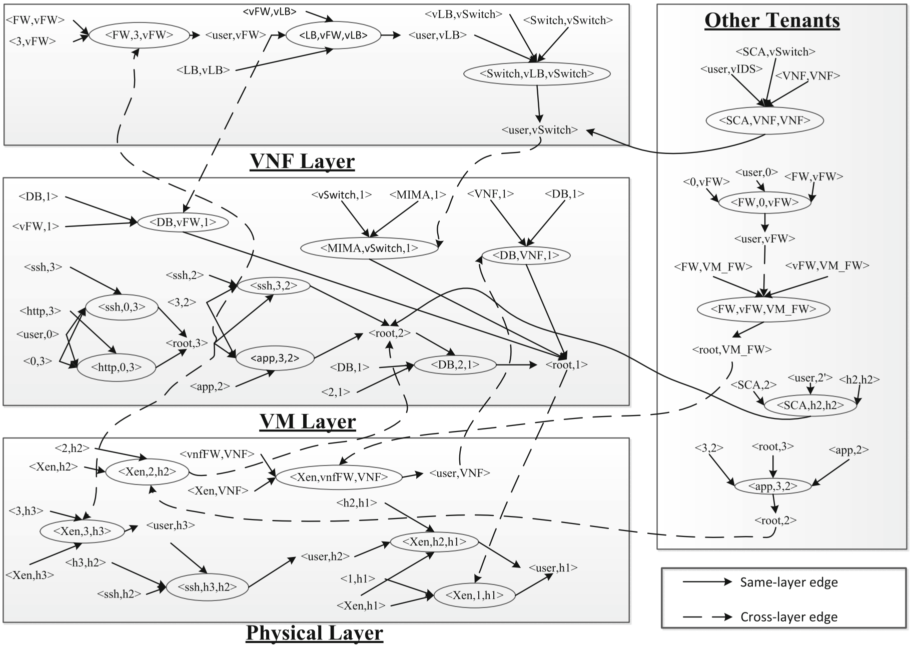

Fig. 3. An example of cross-layer resource graph (FW: firewall, LB: load balancer, DB: database, MIMA: man in the middle attack, SCA: side-channel attack)

on host 2). The edges point from pre-conditions to an exploit and then to its post-conditions, which indicate that any exploit can be executed if and only if all of its pre-conditions are satisfied, whereas executing an exploit is enough to satisfy all of its post-conditions. The following provides formal definitions of those concepts.

Definition 1 (Same-layer Resource Graph). *Given a collection of hosts (physical hosts or VMs) H and the set of resources HR (services or VNFs running on a physical host or a VM) with the resource mapping $rm(.) : H \rightarrow 2^{HR}$, and also given a set of zero day exploits $E = \{< r, h_s, h_d > | h_s \in H, h_d \in H, h_r \in rm(h_d)\}$ and the set of their pre- and post-condition C, a same-layer resource graph is a directed graph $G(E \cup C, HR_r \cup HR_i)$ where $pre \subseteq C \times E$ and $post \subseteq E \times C$ are the pre- and post-condition relations, respectively.*

Definition 2 (Cross-layer Resource Graph). *Given the same-layer resource graph for the three layers, $G_i(E_i \cup C_i, pre_i \cup post_i)(1 \leq i \leq 3)$, and given the cross-layer edges $pre_c \subseteq \bigcup_1^3 C_i \times \bigcup_1^3 E_i$ and $post_c \subseteq \bigcup_1^3 E_i \times \bigcup_1^3 C_i$, a cross-layer resource graph is a directed graph $G(\bigcup_1^3(E_i \cup C_i), \bigcup_1^3(pre_i \cup post_i) \cup pre_c \cup post_c)$.*

3.2 Modeling Co-Residency Attacks

In modeling co-residency attacks, our main idea is to capture the consequences of such attacks as satisfying certain conditions inside the cross-layer resource graph of the targeted tenant. For example, in Fig. 3, the left-hand side shows the cross-layer resource graph of the targeted tenant, which depicts what an external attacker may do to compromise the critical asset <user, h1>. On the other hand, the right-hand side of the figure depicts the insider threat coming from potential co-residency attacks launched by other tenants. The (dashed) lines pointing from the right to the left side of the figure show that, as the consequence of the co-residency attacks (right), some conditions inside the targeted tenant's resource graph (left) may be satisfied, either within the same layer or across different layers. The co-residency could occur w.r.t. the physical layer, which is similar to the co-residency in clouds (when tenants share the same physical host). The co-residency could also occur w.r.t. the VMs when tenants employ the same VM to run their VNFs, which is unique to NFV.

For example, as shown on the right-hand side of Fig. 3, a malicious co-resident tenant can potentially gain a user privilege on the vSwitch service of the targeted tenant through a local exploit <SCA, VNF, VNF>, or he/she can gain a root privilege on a VM on host 2 through a similar attack (<SCA, h3, h3>) (where <user, 3'> means the privilege of the malicious tenant on a VM on host 3). A malicious tenant who shares VNFs running on the same VM may attack the virtual firewall VNF and subsequently the corresponding VM to eventually be able to control the firewall (<root, VM_FW>) and modify its rules in order to gain access to the critical asset. A malicious tenant can also exploit an application running on VM 2 to gain control of that VM, and subsequently attack the co-residing host h2. These examples show how our model can capture co-residency attacks between different layers of the NFV stack.

3.3 Applying the Security Metric

Before we could mitigate the modeled security threats, we need to first quantify them such that we could evaluate the level of threats before and after we apply a hardening option. For this purpose, we apply the k-zero day safety security metric [39,40] originally proposed for traditional networks. The metric basically counts how many distinct unknown vulnerabilities must be exploited in order to compromise a given critical asset. A larger k value will indicate a relatively more secure network because the possibility of having more unknown vulnerabilities occurring at the same time, inside the same network, and exploitable by the same attacker would be significantly lower. The metric can be evaluated on the resource graph of a network, which basically gives the length of the shortest path (in terms of the number of distinct zero day exploits). The exploits of known vulnerabilities can be either regarded as a shortcut (i.e., they do not count toward k) or assigned with a significantly lower weight in the calculation of k.

On the basis of the cross-layer resource graph model introduced in previous sections, the k-zero day safety metric (k0d) can be applied in several ways. First, we could evaluate the metric on the cross-layer resource graph of the targeted tenant, without considering others tenants, whose result provides an estimation for the threat coming from external attackers. Second, we could also evaluate the metric on the cross-layer resource graph including co-resident attacks from others tenants, and we could consider one particular malicious tenant, or multiple such tenants either separately (assuming they do not collude) or collectively (as one, assuming they may collude). Third, we could evaluate the metric before, and after applying a placement or migration-based hardening option, and the difference in the results will indicate the effectiveness of such a hardening option (which we will further investigate in next section).

For example, in Fig. 3, by considering a malicious tenant sharing the same physical host with the targeted tenant (indicated by privilege <root, VM_FW>) would yield a k0d value of 2 since two zero day exploits <Xen, vnfFW, NFV>, and <DB, VNF, 1> are needed to reach the critical asset. Whereas considering a malicious tenant with privilege <user, 2'> (here 2' indicates the privilege belongs to a tenant different from the targeted one) would yield a k value of 3 since three zero day exploits, <SCA, h2, h2>, <DB, 2, 1>, and <Xen, 1, h1> are needed.

4 Attack Mitigation

In this section, we present the optimization-based mitigation through placement and migration of VNFs and VMs, and demonstrate its applicability through discussing several use cases.

4.1 Optimization-Based Mitigation

Based on our previous definition of cross-layer resource graph model and the discussions on modeling co-residency and applying the $k0d$ metric, we can define the problem of optimal placement and migration of VMs and VNFs. As shown in Definition 3, hosts and resources are defined in a way such that the placement and migration may apply to both VMs (on physical hosts) and VNFs (on VMs) through the resource mapping function. The objective function is the application of the $k0d$ metric to the cross-layer resource graph (which can under a value assignment of the resource mapping function. Note that the application of the $k0d$ metric could take several forms for different purposes, as discussed in Sect. 3.3), which gives different variations of the optimization problem. Although not specified in the definition, constraints may be given in terms of possible value assignments to the resource mapping function, e.g., which VM (or VNF) may be placed or migrated to which physical host (or VM), or a threshold for the maximum number of migrated VMs.

Definition 3 (The optimal NFV co-residency problem). *Given a collection of hosts (physical hosts or VMs) H, the set of resources HR (services* *or VNFs running on a physical host or a VM), and the collection of tenants T with the tenant mapping function $tm(.) : HR \rightarrow T$, the optimal NFV co-residency problem is to find a resource mapping function $rm(.) : H \rightarrow 2^{HR}$ to maximize $k0d(G)$ where G is the cross-layer resource graph, and $k0d(G)$ denotes the application of the $k0d$ metric to G.*

The optimal NFV co-residency problem we have defined is intractable, since it can be easily reduced to the NP-hard problem of network hardening through diversity in traditional networks [7]. Specifically, the goal in the diversity problem is to maximize the $k0d$ metric by changing the instance of services (e.g., from IIS to Apache for web service), assuming that different instances of the same service along the shortest path would both count toward the k value (conversely, two identical instances would only count as one). Our problem can be reduced to this since, for any given resource graph G under the diversity problem, we can construct a special case of our problem by regarding G as the VM-layer resource graph, and regarding the instance of a service as the physical host on which that service resides (such that identical instances of a service are always co-resident). By further assuming that the attacker can always trivially exploit all co-resident services as long as he/she can exploit one (i.e., co-resident services only count as one toward the k value), the two problems then become equivalent.

In our study, we use the genetic algorithm (GA) [18] to optimize the VM (VNF) placement and migration for maximizing k. Our choice of GA is inspired by [13] and based on the fact that GA provides a simple way to encode candidate solutions and requires little information to search effectively in a large search space [18]. Specifically, the cross-layer resource graph is taken as input to the optimization algorithm, with k (averaged between tenants) as the fitness function. We try to find the best VM placement within a reasonable number of generations. The constraints we have considered include defining the resource mapping function in the case that specific VMs can be assigned to each host (e.g., firewall only), enforcing a given placement policy (e.g., CloudSim [11] placement policy), or satisfying a maximum number of migrating VMs. In our simulations, we choose the probability of 0.8 for crossover and 0.2 for mutation based on our experiences.

4.2 Use Cases

We demonstrate the applicability of our solution through several use cases with different types of attackers and while considering different layers. The first use case contrasts an external attacker to an insider launching cross-tenant attacks. The second use case compares a same-layer attack versus a cross-layer attack. The last use case is based on the motivating example shown in Sect. 2.3.

- Use Case A: In this case, we have an external attacker using a victim tenant's resources, and an insider malicious tenant co-residing with the victim tenant. Figure 4 shows the cross-layer resource graph for the external attacker (AE) and the insider attacker (AI). The figure shows the shortest path (dashed

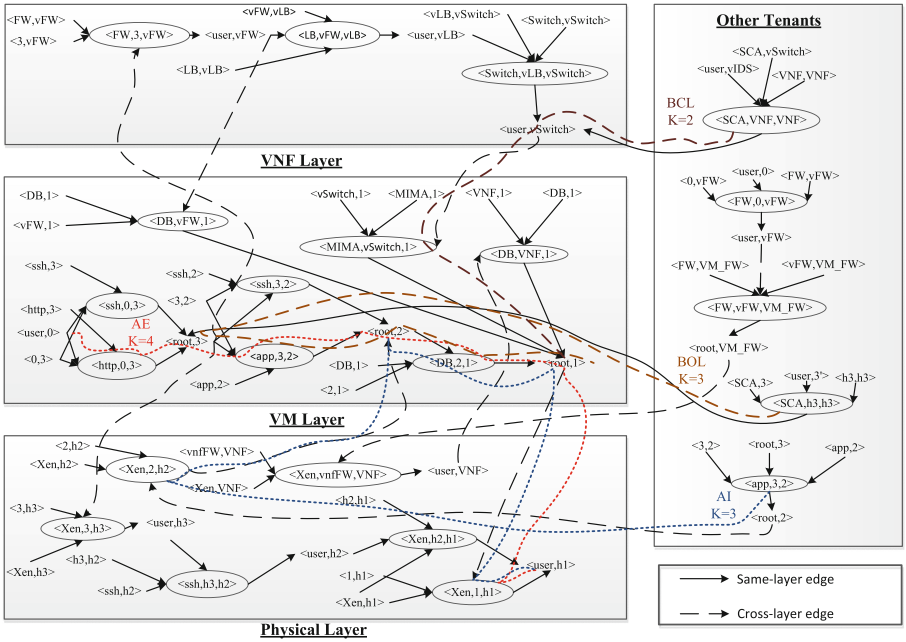

Fig. 4. Use Cases A and B (FW: firewall, LB: load balancer, DB: database, MIMA: man in the middle attack, SCA: side-channel attack, AE: external attacker, AI: insider attacker, BCL: cross-layer attack, BOL: one-layer attack)

lines) for calculating the $k0d$ metric, and the critical asset is represented as $< user, h1 >$. After optimization, the value of k for the external attacker (AE) is 4, and for the insider attacker (AI) is 3 (which means there is less room to mitigate the insider threat).

- Use Case B: In this case, we compare a one-layer attack (BOL) to a cross-layer attack (BCL) for an insider malicious tenant. The BOL and BCL dashed lines shown in Fig. 4 show the shortest paths for the malicious tenant using his/her co-residency with the victim tenant to reach the target $< root, 1 >$. After optimization, the value $k = 3$ for BOL and $k = 2$ for BCL show that there is less room to mitigate the insider threat when the attack may go across layers.
- Use Case C: This case shows the optimal placement result for our motivating example discussed in Sect. 2.3. We consider three tenants (Alice (A), Bob (B), and Mallory (M)) and three servers each of which can host four VMs. We consider Mallory a malicious tenant and the database VM belonging to Alice a target. Table 1 shows three different placements. The upper left table shows the placement before applying our optimization solution and the value of $k = 2$. The right upper table and the bottom table show the optimal placement after we apply our solution by the victim tenant (where the migration is limited to the tenant's resource) and the provider (where the

Table 1. The optimal solution to the motivating example (Sect. 2.3)

Host	VM / VDU				VM / VDU			
1	app A	app B	DB A	DB B	app A	app B	DB A	DB B
2	LB A / Switch A	Router A	http B	http A	LB A / Switch A	FW A / IDS A	http B	http A
3	FW A / IDS A	FW B / IDS B	http M	Router B	Router A	FW B / IDS B	http M	Router B
	Before mitigation $k = 2$				After mitigation by tenant $k = 4$			

Host	VM / VDU			
1	app A	app B	DB A	FW A / IDS A
2	LB A / Switch A	Router A	http B	http A
3	DB B	FW B / IDS B	http M	Router B
	After mitigation by provider $k = 5$			

migration is applied to all resources), respectively. The value of k increases to $k = 4$ and $k = 5$ for mitigation by the tenant and provider, respectively. The result of mitigation by the provider is slightly better than by the tenant because more VMs may be migrated.

5 Simulations

This section shows the simulation results of applying our mitigation solution under various constraints. All VM placement in the simulations are based on CloudSim [9,11,20]. We applied the three placement policies in CloudSim (i.e., the random, least, and most policies) to our NFV environment. We have 300 hosts and 7,000 VMs, and the following shows the default configurations for the host and VMs from CloudSim.

- For the physical machine, we specify the capacity of the hosts as having 16 GB of RAM, 1000 GB of storage space and a 10,000 MB/s bandwidth
- The virtual machine's resource requirements are 512 MB of RAM, 10 GB of storage space and a 1,000 MB/s bandwidth

Moreover, we use a virtual machine equipped with a 3.4 GHz CPU and 8 GB RAM in the Python 2.7.10 environment under Ubuntu 12.04 LTS and the MATLAB R2017b's GA toolbox. To generate a large number of resource graphs for simulations, we start with seed graphs with realistic configurations similar to Fig. 2 and the cloud infrastructure configurations presented in [1,2], and then generate random resource graphs by injecting new nodes and edges into those seed graphs based on the VM placement results of CloudSim. Those resource graphs were used as the input to the optimization toolbox where the fitness function maximizes the average insider threat value k under various constraints. The parameters and constraints used in our simulations include the VMs placement policy, size of the network, type and number of attackers, and maximum number of VMs migrating to malicious users. We repeat each simulation on 400 different resource graphs to obtain the average result.

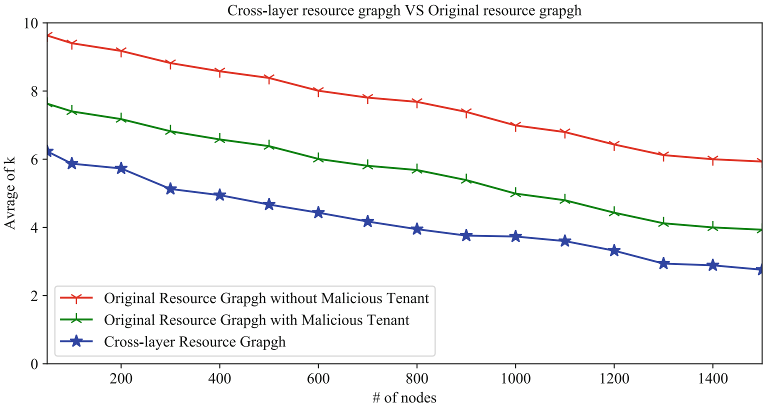

Fig. 5. Comparing the original resource graph and cross-layer resource graph

The objective of the first simulation is to study how cross-layer attacks may affect the security of the NFV stack. We compare the $k0d$ metric on the original resource graph (without any cross-layer attacks), and cross-layer resource graphs. In Fig. 5, the number of malicious users (external attackers or insider tenants) is between 5 and 15, while the size of the network varies between 50 and 1,500 along the X-axis. The Y-axis shows the average of k among all malicious users. The red line represents the results of the original resource graph without considering malicious tenants in this particular case. The green line represents the results of the original resource graph while considering malicious tenants. The blue line shows the result of cross-layer resource graph (with both malicious tenants and cross-layer attacks considered).

Results and Implications: From the results, we can make the following observations. First, the value of k decreases in all cases almost linearly; this is expected because, as the size of the network increases, there is a higher chance for the length of the shortest path to decrease, which means attackers may require less attack steps. Second, the value of k drops on the original resource graph (without considering cross-layer attacks) after considering the presence of malicious tenants (i.e., co-residency attacks), which is as expected. Third, the value of k drops by approximately 55% between the original resource graph without considering the malicious tenant, and the cross-layer resource graph, which shows the additional threat of cross-layer attacks.

In Fig. 6 the objective is to show how different placement policies can affect the value of k. In this simulation, we employ the cross-layer resource graph to measure the value of k for three types of attackers (external, malicious tenant, and lower-layer provider who has access to all the hosts) under three different placement policies used in CloudSim (i.e., the most, least, and random policies). The three figures show how the three placement policies can slightly affect the value of k. Each trend on the figure shows a different type of attackers, while the

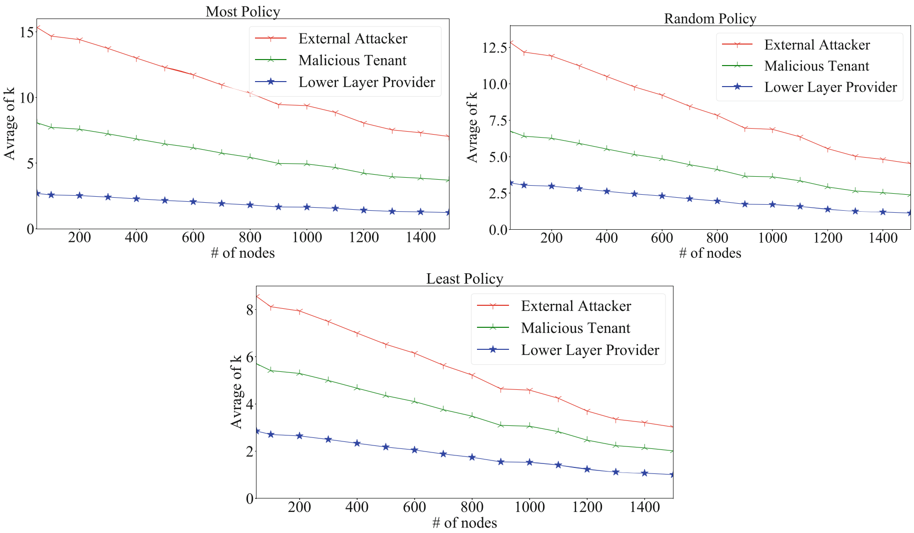

Fig. 6. Comparing the three placement policies in CloudSim

total number of attackers stays between 5 and 20. The X-axis depicts the size of the network and the Y-axis shows the average value of k among all attackers.

Results and Implications: From the three figures, we can make the following observations. First, similar as in previous results, the value of k in all trends decreases almost linearly as the size of the network increases. Second, the trends of external attackers and malicious tenants decrease faster than the lower-layer provider. This is expected because the lower-layer provider already has access to all the hosts, which enables him/her to either use his/her privileges to attack higher layers, which means much lower k values and hence less room for further decrease as the network size increases. Finally, the most placement policy has the highest value of k both external attackers and malicious tenants and the lowest k for lower layer provider. This is because, under the most policy, the target tenant's VMs tend to stay closer to each other, which renders them less vulnerable to external attackers or malicious tenants, but more so to a lower-layer provider managing the hosts of such VMs.

The objective of the next three simulations is to study how the different types of attackers behave under our attack mitigation solution. Figure 7 shows the simulation of applying the mitigation solution on the least placement policy for external attackers and malicious tenants, and the placement policy for the lower-layer provider. The three simulations are based on similar X and Y axis as in previous simulations. The solid lines represent the results after applying our mitigation solution under the constraints of the maximum number of VMs migration. The dashed lines represent the results before applying the mitigation solution.

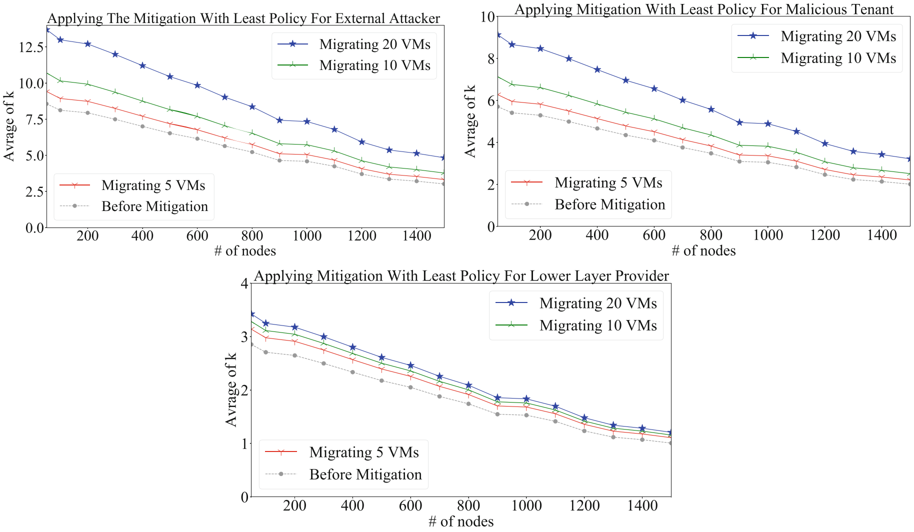

Fig. 7. Applying mitigation solution with the maximum number of VMs migrating

Results and Implications: From the simulation results, we can make the following observations. First, our solution is improving the value of k in all cases. Second, all three simulations follow the same trend and the value of k improves when we increase the maximum number of VMs that are allowed to migrate, i.e., the cost of migration. Finally, improving the result for the lower-layer provider is difficult to attain because the low-layer provider already is assumed to have the power to access more than one host (based on the privilege he/she has) so migration has less effect.

The objective of the last simulation is to study how the number of malicious tenants can increase insider threat under different placement policies, and how the mitigation solution may improve the value of k in each case. In Fig. 8, the size of the network is fixed at 700 nodes, while the number of malicious tenants is varied between 0 and 25 along the X-axis. The Y-axis shows the average value of k among all malicious tenants. The solid lines represent the results after applying the mitigation solution, and the dashed lines are for the corresponding results before applying the solution.

Results and Implications: From the results, we can make the following observations. First, the mitigation solution successfully reduces the insider threat (increasing the average of k values) in all cases. Second, the results before and after applying the solution start with a sharp decrease prior to following similar linear trends (meaning increased insider threat) as the number of malicious tenants increase from zero. Finally, the result of the random placement policy after applying the solution is slightly better than the result of the most placement policy before applying the solution, which means that the mitigation solution may improve the placement algorithm w.r.t. the co-residency attack.

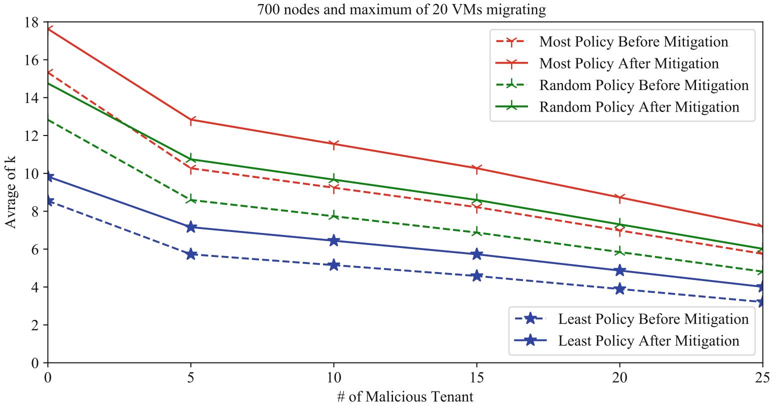

Fig. 8. The results of the mitigation solution under different placement policies

6 Related Work

To the best of our knowledge, this is the first work proposing a threat model specifically for NFV environments. On the other hand, many exists works focus on clouds. In particular, our previous work applies different threat modeling techniques to cloud data center infrastructures for different types of attackers [2]. Gruschka and Jensen devise a high level attack surface framework to show from where the attack can start [19]. The NIST emphasizes the importance of security measuring and metrics for cloud providers in [29]. A framework is propose by Luna et al. for cloud security metrics using basic building blocks [26]. There exist other works focusing on insider threats in clouds [1,10]. Chinchani et al. proposed a graph-based model for insider attacks and measure the threat [10]. There are many works that focus on the co-residency attacks by improving the placement policy. Han et al. introduces a new strategy to prevent attackers from achieving co-residency by modifying the placement policy on CloudSim [20]. Madi et al. propose a quantitative model and security metric for multi-tenancy in the cloud at different layers [27]. Atya et al. study co-residency in clouds and suggest solutions for the victim tenant to avoid co-residency with malicious users [4].

Unlike our work which focuses on the cross-layer and co-resident threats and the application of threat models and security metric, existing studies on NFV security [16,25,31,42] mostly focus on issues related to virtualization. Lal et al. [25] propose to adapt several well-known best practices like VM separation, hypervisor introspection, and remote attestation to NFV. Pattaranantakul et al. [31] adapt best practices like access control to address virtualization-related threats in NFV. Our cross-layer resource graph model is partially inspired by existing works [35,36] in which Sun et al. use a cross-layer Bayesian network to measure security threats for enterprise networks [35] and additionally they employ a multi-layer attack graph to measure security in clouds [36].

There also exist related works on other aspects of NFV security. Firoozjaei et al. [16] show how multi-tenancy and live migration can affect the security on NFV by using a side-channel and shared resource misuse attack. Alnaim et al. [3] uses architectural modeling to analyze security threats and the possible mitigating solution for NFV; their model is relatively abstract and only considers a malicious tenant to exploit vulnerability when he/she co-resides with the target VM on the same physical machine. Tian et al. propose a framework that uses a hierarchical attack and defense model which divides the 5G network to four layers (physical layer, virtual layer, service layer, and application layer) [37]. Basile et al. [5] propose to add a new policy manager component to enforce security policies during deployment and configuration of security functions. Coughlin et al. [12] integrate trusted computing solution based on Intel SGX to enforce privacy with secure packet processing.

7 Conclusion

In this paper, we have modeled cross-level and co-residency attacks in the NFV stack. We have also formulated the optimal VNF/VM placement problem to mitigate the security threats through standard optimization algorithm. Furthermore, we conducted simulations whose results showed that our solution could significantly reduce the level security threats in NFV. Our future work will focus on following directions. First, we will make our solution incremental and more efficient in order to handle more dynamics in terms of VM placement and immigration. Second, we will consider weighing different exploits and asset values to optimally choose among those different options for a given NFV application. Finally, we will study the integration of our solution with existing deployment policies based on an NFV testbed.

Acknowledgements. The authors thank the anonymous reviewers for their valuable comments. This work was partially supported by the National Science Foundation under grant number IIP-1266147, by the Army Research Office grant W911NF-13-1-042, and by the Natural Sciences and Engineering Research Council of Canada under Discovery Grant N01035.

References

1. Alhebaishi, N., Wang, L., Jajodia, S., Singhal, A.: Mitigating the insider threat of remote administrators in clouds through maintenance task assignments. J. Comput. Secur. **27**(4), 427–458 (2019)
2. Alhebaishi, N., Wang, L., Singhal, A.: Threat modeling for cloud infrastructures. ICST Trans. Secur. Saf. **5**(17), e5 (2019)
3. Alnaim, A.K., Alwakeel, A.M., Fernandez, E.B.: Threats against the virtual machine environment of NFV. In: 2019 2nd International Conference on Computer Applications Information Security (ICCAIS), pp. 1–5, May 2019
4. Atya, A.O.F., Qian, Z., Krishnamurthy, S.V., Porta, T.F.L., McDaniel, P.D., Marvel, L.M.: Catch me if you can: a closer look at malicious co-residency on the cloud. IEEE/ACM Trans. Netw. **27**(2), 560–576 (2019)

5. Basile, C., Lioy, A., Pitscheider, C., Valenza, F., Vallini, M.: A novel approach for integrating security policy enforcement with dynamic network virtualization. In: NetSoft 2015, pp. 1–5 (2015)
6. Bates, A., Mood, B., Pletcher, J., Pruse, H., Valafar, M., Butler, K.: Detecting co-residency with active traffic analysis techniques. In: Proceedings of the 2012 ACM Workshop on Cloud Computing Security Workshop, CCSW 2012, pp. 1–12, Association for Computing Machinery, New York (2012)
7. Borbor, D., Wang, L., Jajodia, S., Singhal, A.: Diversifying network services under cost constraints for better resilience against unknown attacks. In: Proceedings of the 30th Annual IFIP WG 11.3 Conference on Data and Applications Security and Privacy XXX, DBSec 2016, Trento, Italy, 18–20 July 2016, pp. 295–312 (2016). https://doi.org/10.1007/978-3-319-41483-6_21
8. Borbor, D., Wang, L., Jajodia, S., Singhal, A.: Securing networks against unpatchable and unknown vulnerabilities using heterogeneous hardening options. In: Livraga, G., Zhu, S. (eds.) DBSec 2017. LNCS, vol. 10359, pp. 509–528. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61176-1_28
9. Calheiros, R.N., Ranjan, R., Beloglazov, A., Rose, C.A.F.D., Buyya, R.: CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw. Pract. Exp. **41**(1), 23–50 (2011)
10. Chinchani, R., Iyer, A., Ngo, H.Q., Upadhyaya, S.: Towards a theory of insider threat assessment. In: 2005 International Conference on Dependable Systems and Networks (DSN 2005), pp. 108–117, June 2005
11. CloudSim. CloudSim: a framework for modeling and simulation of cloud computing infrastructures and services (2020). http://www.cloudbus.org/cloudsim/. Accessed 27 Jan 2020
12. Coughlin, M., Keller, E., Wustrow, E.; Trusted click: overcoming security issues of NFV in the cloud. In: SDN-NFV@CODASPY 2017, pp. 31–36 (2017)
13. Dewri, R., Poolsappasit, N., Ray, I., Whitley, L.D.: Optimal security hardening using multi-objective optimization on attack tree models of networks. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (eds.) Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, 28–31 October 2007, pp. 204–213. ACM (2007)
14. ETSI: ETSI-Welcome to the World of Standards. https://www.etsi.org
15. Fayazbakhsh, S.K., Reiter, M.K., Sekar, V.: Verifiable network function outsourcing: requirements, challenges, and roadmap. In: Workshop on Hot Topics in Middleboxes and Network Function Virtualization (HotMiddlebox 2013), pp. 25–30 (2013)
16. Firoozjaei, M.D., Jeong, J.P., Ko, H., Kim, H.: Security challenges with network functions virtualization. Future Gener. Comput. Syst. **67**, 315–324 (2017)
17. Flittner, M., Scheuermann, J.M., Bauer, R.: ChainGuard: controller-independent verification of service function chaining in cloud computing. In: NFV-SDN 2017, pp. 1–7 (2017)
18. Golberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addion wesley, Boston (1989)
19. Gruschka, N., Jensen, M.: Attack surfaces: a taxonomy for attacks on cloud services. In: 2010 IEEE 3rd International Conference on Cloud Computing, pp. 276–279. IEEE (2010)
20. Han, Y., Chan, J., Alpcan, T., Leckie, C.: Virtual machine allocation policies against co-resident attacks in cloud computing. In: IEEE International Conference on Communications, ICC 2014, Sydney, Australia, 10–14 June 2014, pp. 786–792 (2014)

21. Huang, W., Zhu, H., Qian, Z.: AutoVNF: an automatic resource sharing schema for VNF requests. J. Internet Serv. Inf. Secur. **7**(3), 34–47 (2017)
22. Intel. Realising the benefits of network functions virtualisation in telecoms network. https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/benefits-network-functions-virtualization-telecoms-paper.pdf./
23. Ixia, A.K.B.: Network Function Virtualization (NFV): 5 Major Risks. https://www.ixiacom.com/resources/network-function-virtualization-nfv-5-major-risks/
24. Lakshmanan, S., et al.: Modeling NFV deployment to identify the cross-level inconsistency vulnerabilities. In: 2019 11th IEEE International Conference on Cloud Computing Technology and Science (CloudCom), pp. 167–174, December 2019
25. Lal, S., Taleb, T., Dutta, A.: NFV: security threats and best practices. IEEE Commun. Mag. **55**(8), 211–217 (2017)
26. Luna, J., Ghani, H., Germanus, D., Suri, N.: A security metrics framework for the cloud. In: 2011 Proceedings of the International Conference on Security and Cryptography (SECRYPT), pp. 245–250, July 2011
27. Madi, T., et al.: QuantiC: distance metrics for evaluating multi-tenancy threats in public cloud. In: 2018 IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2018, Nicosia, Cyprus, 10–13 December 2018, pp. 163–170 (2018)
28. Marchetto, G., Sisto, R., Yusupov, J., Ksentini, A.: Virtual network embedding with formal reachability assurance. In: CNSM 2018, pp. 368–372 (2018)
29. National Institute of Standards and Technology. Cloud Computing Service Metrics Description. http://www.nist.gov/itl/cloud/upload/RATAX-CloudServiceMetricsDescription-DRAFT-20141111.pdf
30. Pattaranantakul, M., He, R., Meddahi, A., Zhang, Z.: SecMANO: towards network functions virtualization (NFV) based security management and orchestration. In: 2016 IEEE Trustcom/BigDataSE/ISPA, pp. 598–605, August 2016
31. Pattaranantakul, M., He, R., Song, Q., Zhang, Z., Meddahi, A.: NFV security survey: from use case driven threat analysis to state-of-the-art countermeasures. IEEE Commun. Surv. Tutor. **20**(4), 3330–3368 (2018)
32. Rates Crippa, M., et al.: Resource sharing for a 5g multi-tenant and multi-service architecture. In: 23th European Wireless Conference on European Wireless 2017, pp. 1–6, May 2017
33. Saripalli, P., Walters, B.: QUIRC: a quantitative impact and risk assessment framework for cloud security. In: 2010 IEEE 3rd International Conference on Cloud Computing, pp. 280–288, July 2010
34. Sheyner, O., Haines, J., Jha, S., Lippmann, R., Wing, J.M.: Automated generation and analysis of attack graphs. In: Proceedings of the 2002 IEEE Symposium on Security and Privacy 2002, pp. 273–284 (2002)
35. Sun, X., Dai, J., Singhal, A., Liu, P.: Inferring the stealthy bridges between enterprise network islands in cloud using cross-layer bayesian networks. In: Tian, J., Jing, J., Srivatsa, M. (eds.) SecureComm 2014. LNICSSITE, vol. 152, pp. 3–23. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23829-6_1
36. Sun, X., Singhal, A., Liu, P.: Towards actionable mission impact assessment in the context of cloud computing. In: Proceedings of the 31st Annual IFIP WG 11.3 Conference on Data and Applications Security and Privacy XXXI, DBSec 2017, Philadelphia, PA, USA, 19–21 July 2017, pp. 259–274 (2017)
37. Tian, Z., Sun, Y., Su, S., Li, M., Du, X., Guizani, M.: Automated attack and defense framework for 5G security on physical and logical layers. CoRR, abs/1902.04009 (2019)

38. Tschaen, B., Zhang, Y., Benson, T., Banerjee, S., Lee, J., Kang, J.-M.: SFC-checker: checking the correct forwarding behavior of service function chaining. In: NFV-SDN 2016, pp. 134–140 (2016)
39. Wang, L., Jajodia, S., Singhal, A., Cheng, P., Noel, S.: k-zero day safety: a network security metric for measuring the risk of unknown vulnerabilities. IEEE Trans. Dependable Secure Comput. **11**(1), 30–44 (2014)
40. Wang, L., Jajodia, S., Singhal, A., Noel, S.: k-zero day safety: measuring the security risk of networks against unknown attacks. In: Gritzalis, D., Preneel, B., Theoharidou, M. (eds.) ESORICS 2010. LNCS, vol. 6345, pp. 573–587. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15497-3_35
41. Wang, Y., Li, Z., Xie, G., Salamatian, K.: Enabling automatic composition and verification of service function chain. In: IWQoS 2017, pp. 1–5 (2017)
42. Yang, W., Fung, C.: A survey on security in network functions virtualization. In: NetSoft 2016, pp. 15–19 (2016)
43. Zhang, X., Li, Q., Wu, J., Yang, J.: Generic and agile service function chain verification on cloud. In: IWQoS 2017, pp. 1–10 (2017)
44. Zhang, Y., Juels, A., Oprea, A., Reiter, M.K.: HomeAlone: co-residency detection in the cloud via side-channel analysis. In: 2011 IEEE Symposium on Security and Privacy, pp. 313–328, May 2011
45. Zhang, Y., Wu, W., Banerjee, S., Kang, J.-M., Sanchez, M.A.: SLA-verifier: stateful and quantitative verification for service chaining. In: INFOCOM 2017, pp. 1–9 (2017)