View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Queen's University Research Portal

Application-Centric Provisioning of Virtual Security Network Functions

Doriguzzi-Corin, R., Scott-Hayward, S., Siracusa, D., & Salvadori, E. (2017). Application-Centric Provisioning of Virtual Security Network Functions. In 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks, Workshop on Security in NFV-SDN (SN2017) (pp. 276-279). Berlin, Germany: Institute of Electrical and Electronics Engineers (IEEE). DOI: 10.1109/NFV-SDN.2017.8169861

Published in:
2017 IEEE Conference on Network Function Virtualization and Software Defined Networks, Workshop on Security in NFV-SDN (SN2017)

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

Publisher rights
Copyright 2017 IEEE. This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher.

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Download date:09. Sep. 2018

Application-Centric Provisioning of Virtual Security Network Functions

R. Doriguzzi-Corin[α], S. Scott-Hayward[β], D. Siracusa[α], E. Salvadori[α]
[α]CREATE-NET, Fondazione Bruno Kessler - Italy
[β]CSIT, Queen's University Belfast - Northern Ireland

***Abstract*—Network Function Virtualization (NFV) enables flexible implementation and provisioning of network functions as virtual machines running on commodity servers. Due to the availability of multiple hosting servers, such network functions (also called Virtual Network Functions (VNFs)) can be placed where they are actually needed, dynamically migrated, duplicated, or deleted according to the current network requirements. However, the placement of VNFs within the physical network is one of the main challenges in the NFV domain as it has a critical impact on the performance of the network. In this work we focus on efficient placement of Virtual Security Network Functions (VSNFs), i.e. the placement of virtual network functions whose purpose is to prevent or mitigate network security threats. In this regard, we tackle the placement problem not only considering performance optimization aspects, but also trying to find solutions that are consistent from the security viewpoint. Specifically, the main contribution of this paper is the formulation of the placement problem by taking into account both Security and Quality of Service (QoS) requirements of user applications.**

I. Introduction

Network security implemented by Telecommunication Service Providers (TSPs) has traditionally been based on the deployment of specialized, closed, proprietary Hardware Appliances (HAs) for each security network function. Such HAs are fixed in terms of functionality and placement in the network, which means that even slight changes in the security requirements generally necessitate manually intensive and time-consuming re-configuration tasks, the replacement of existing HAs or the deployment of additional HAs. This leads to (i) limited protection of end-users from known security threats, (ii) slow reaction to new security threats or variations of known attacks, and (ii) high Operating Expenditure (OpEx) and Capital Expenditure (CapEx) for TSPs.

The NFV [1] initiative has been proposed as a possible solution to address the operational challenges and high costs of managing proprietary HAs. The main idea behind NFV is to transform network functions (e.g. firewalls, intrusion detection systems etc.) based on proprietary HAs, into software components (called VNFs) that can be deployed and executed in virtual machines on commodity high-performance servers. This approach decouples the software from the hardware, thus it allows any (security) network function to be deployed in any server connected to the network through an automated and centralized management system.

The centralized management system, called NFV Management and Orchestration (NFV MANO), controls the whole life-cycle of each VNF. Specifically, the NFV MANO dynamically creates, destroys, or migrates any VNF, depending on specific network requirements or to balance the load across different servers. Network Service Chaining (NSC) is a technique for selecting subsets of the network traffic and forcing them to traverse various VNFs in sequence. For example, a firewall followed by a Intrusion Prevention System (IPS), then a Network Address Translation (NAT) service and so on. NSC and NFV enable flexible, dynamic service chain modifications to meet the real-time network demands.

In this paper, we investigate how to efficiently provision application-specific security services by using NSC deployment techniques and by ensuring that the user's application requirements in terms of security and QoS are met. The main contribution of this work is a mathematical formulation of the placement problem where the objective function requires minimization of the usage of network, computational and memory resources subject to the following constraints:

- The placement must ensure that the Service Level Agreement (SLA) between the TSP and the user is not violated (namely, maximum end-to-end latency and minimum bandwidth). Thus, the algorithm must place the VSNF chains where links have enough residual capacity and where nodes have enough residual computational and memory resources to execute the functions of the chains.
- The placement must fulfill the security policies and best practices as defined by the TSP. Namely, the order in which the VSNFs are executed, the position of the VSNFs in the network, and the operational mode of VSNFs (either stateless or stateful).

The remainder of this paper is structured as follows: Section II provides the motivation behind this work. Section III details the mathematical formulation of the placement problem. Section IV presents the related work. Finally, the conclusions are provided in Section V.

II. Motivation

In NFV deployments, the placement of VNFs has a significant impact on the performance of the network and on the QoS level the operator can guarantee to users. Several studies in the peer-reviewed literature tackle the problem by proposing architectural and mathematical solutions with the aim of optimizing the utilization of network and computational resources and minimizing the operational costs [2], [3]. As reported in Section IV, recent works go beyond the simple cost and resource optimization by introducing more specific constraints, either to enforce the security of the network [4], [5], or to guarantee QoS parameters such as minimum bandwidth and maximum end-to-end latency [6], [7].

We argue that the placement model for security VNFs cannot neglect the SLA between operator and user, as the SLA defines mandatory QoS policies required by the user. Omitting

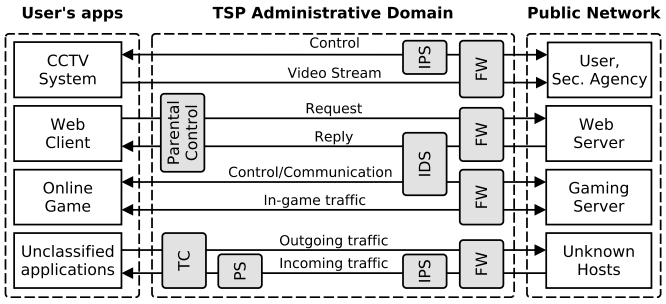

Fig. 1. Motivating example.

the QoS requirements may lead, for instance, to a model that blindly forces all the user traffic to traverse the whole chain of VSNFs. As a result, computationally demanding VSNFs such as Intrusion Detection Systems (IDSs) may cause a noticeable performance degradation to latency-sensitive applications (e.g. online games) or bandwidth sensitive applications (e.g. video streaming). On the other hand, beyond the overall resource consumption and QoS requirements, the model must also take into account the specific security best practices and policies. Omitting such aspects may result in the inappropriate placement of a firewall in the middle of the network, thus allowing unauthorized traffic to reach hosts that should be protected. The approach presented in this paper is illustrated by the sample scenario depicted in Fig. 1. In the example, a TSP exploits the NFV technology to provide security services tailored to specific users' application requirements.

1. For instance, remote access to the user's CCTV system must be guaranteed only to the user and possibly to a security agency. In this case, inspecting the video stream with an IPS would not provide any additional protection but would possibly reduce the frame rate of the video streaming, thus compromising the detection of anomalous events. However, control communication should be inspected.
2. Parental control is applied to Web traffic to block unwanted media and social-media content, while an IDS might be used to intercept malicious software (malware). Stateful VSNFs track the state of network connections (e.g. Layer 4 firewall, NAT). In this case, the same VSNF instance must be traversed by all traffic flows of a network conversation (in Fig. 1, firewall on Web Client-Server interaction). More flexible provisioning schemes can be adopted for stateless VSNFs, where multiple instances of the same VSNF might be deployed on different servers for load balancing.
3. An IDS might also be used to detect possible threats due to the misuse of chatting tools integrated within the gaming software (e.g., phishing [8], social engineering [9], etc.). As the communication between the client and the server relies on timely delivery of packets, IDS operations are not executed on the *in-game* traffic. In this case the security is enforced by a faster VSNF such as a Firewall, which checks the packet headers without any deep-payload analysis. It should be noted that web traffic and chat conversations are often encrypted by TLS/SSL cryptographic protocols. Although encryption preserves the confidentiality of the traffic, it also prevents IDS-based VSNFs such as Parental Control and IDS from inspecting the packets, thus allowing an attacker to obfuscate malicious data in encrypted payloads. However, the TSP could overcome this limitation either using a Transparent Proxy VSNF or by exploiting recent advances in network security [10].
4. As represented in the lowest section of Fig. 1, security best-practices suggest that unwanted traffic should be blocked as soon as it enters the network by placing firewalls and IDS/IPS close to the border of the TSP domain. Another generally accepted practice, is to place firewalls before IDS/IPS (from the point of view of incoming traffic). Firewalls are generally designed to drop non-legitimate traffic very quickly, thus reducing the burden on IDS/IPS, which are more computationally expensive. In this example, unknown traffic is filtered, inspected and possibly classified according to pre-defined categories by a Traffic Classifier (TC) VSNF (e.g. online games, peer-to-peer, media streaming, etc.) to dynamically provision further class-specific VSNF chains. Penetration testing tools, such as Port Scanner (PS) VSNFs, might also be used to proactively find security breaches in the user's configuration.

To summarize, the rationale behind our approach is the following: (i) a user's application should never under-perform because of VSNF operations and (ii) the VSNF placement must obey the operator's security best-practices in terms of application security requirements, position in the network, operational mode (stateless or stateful VSNF), and order with respect to the direction of the traffic. In Section III, we present a mathematical model for the placement of VSNF chains based on these criteria.

III. VSNF PLACEMENT MODEL

In this section, we provide a mathematical model to progressively embed security service requests, formed by one or multiple VSNF chains, onto a physical network substrate by considering the available resources and realistic constraints.

Physical network model. We represent the physical network as an undirected and weighted graph $\mathcal{G} = (N, E)$, i.e. a graph where the edges have no orientation and weights are assigned to nodes and edges (see Fig. 2).

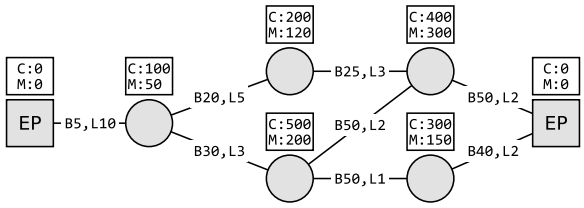

Fig. 2. Representation of the physical network as undirected weighted graph.

A node $i \in N$ is characterized by the computational resources $\gamma_i \in \mathbb{N}$ and memory resources $\mu_i \in \mathbb{N}$ of a server attached to the node, expressed in CPU and memory units respectively (labels `C` and `M` in Fig. 2). Without loss of generality and to simplify the model, μ_i indicates both volatile memory and storage. Endpoints (EP) (e.g., a user's computer and a remote server) are characterized by $\gamma_i = 0$ and $\mu_i = 0$ to prevent the algorithm from placing any VSNF on them.

A link $(k, l) \in E$ is a wired connection between two nodes k and $l \in N$. It is characterized by its capacity $\beta_{k,l} \in \mathbb{N}^+$ and its propagation delay $\lambda_{k,l} \in \mathbb{N}^+$. Both are expressed as

positive integer numbers representing bandwidth and latency units (labels B and L in Fig. 2).

Security service request. We model a security service request as a set of independent weighted directed graphs:

$$\mathcal{G}_s = \{(U^c, U^c_{pairs}) : c \in C_s\}$$

where C_s is the set of unidirectional chains composing the service request. Each graph includes nodes and arcs. Nodes $U^c = A^c \cup V^c$ comprise user and remote applications (A^c, the endpoints of chain c) as well as a subset of all VSNFs (V^c). Each arc in (U^c_{pairs}) delineates the order of traversing the VSNFs $\in V^c$ between endpoints in A^c.

A security service request is represented in Fig. 3. The request in the example comprises three chains, each one identified by the type of traffic and its direction. A security service request is fulfilled if and only if all the chains in the request can be mapped onto the physical network.

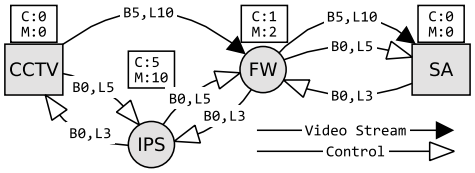

Fig. 3. Example of security service request for the CCTV application.

A node $u \in U^c$ is characterized by its requirements in terms of computational units γ^c_u and memory units μ^c_u for processing the traffic flows in chain $c \in C_s$ (C and M in Fig. 3). Endpoints are characterized by $\gamma^c_u = 0$ and $\mu^c_u = 0$ to allow the algorithm to place them on the physical substrate's endpoints. A VSNF $u \in V^c$ is also characterized by the latency $\lambda^c_{i,u}$ it introduces in the dataplane to process a packet when running on node i. The latency is a function of γ_i and γ^c_u. Each chain $c \in C_s$ is characterized by its requirements in terms of minimum bandwidth β^c and maximum latency λ^c (B and L in Fig. 3).

A. Integer Linear Programming (ILP) formulation

Definitions. Let us first define two binary variables:

- $x^c_{i,u}$ indicates whether node $u \in U^c$ is mapped to $i \in N$.
- $y^c_{k,l,i,j,u,v}$ indicates whether physical link $(k,l) \in E$ belongs to the path between nodes i and j to which $u, v \in U^c$ are mapped.

The residual capacity of a link, $\beta'_{k,l}$, is defined as the total amount of bandwidth available on link $(k,l) \in E$:

$$\beta'_{k,l} = \beta_{k,l} - \sum_{\substack{c \in C,\ i,j \in N \\ (u,v) \in U^c_{pairs}}} \beta^c \cdot y^c_{k,l,i,j,u,v}$$

thus, the nominal capacity of link *(k,l)* minus the bandwidth required by the chains $c \in C$ already mapped on that link.

The residual capacity of a node, is defined as its nominal CPU and memory capacities minus the CPU and memory resources used by the VSNFs v instantiated on the node:

$$\gamma'_i = \gamma_i - \textstyle\sum_{\{c \in C, u \in V^c\}} \gamma^c_u \cdot x^c_{i,u}$$
$$\mu'_i = \mu_i - \textstyle\sum_{\{c \in C, u \in V^c\}} \mu^c_u \cdot x^c_{i,u}$$

Problem formulation. Given a physical network $\mathcal{G}$, for each security service request $\mathcal{G}_s$ find a suitable mapping of all its unidirectional chains on the physical network, which minimizes the physical resources of $\mathcal{G}$ spent to map $\mathcal{G}_s$, also known as the *embedding cost*.
Hence, the solution of the problem is represented by a set of $x^c_{i,u}$ and $y^c_{k,l,i,j,u,v}$ such that the cumulative usage of physical resources for all the virtual networks is minimized:

$$\sum_{\substack{c \in C_s,\ i,j \in N, \\ (k,l) \in E, (u,v) \in U^c_{pairs}}} b_{k,l} \cdot \beta^c \cdot y^c_{k,l,i,j,u,v} + \sum_{c \in C_s, i \in N, u \in V^c} (c_i \cdot \gamma^c_u + m_i \cdot \mu^c_u) \cdot x^c_{i,u}$$

Where $b_{k,l}$, c_i and m_i are the costs for allocating bandwidth, CPU and memory. They penalize the resources with less residual capacity to minimize fragmentation:

$$b_{k,l} = \frac{1}{\beta'_{k,l} + \delta} \quad c_i = \frac{1}{\gamma'_i + \delta} \quad m_i = \frac{1}{\mu'_i + \delta}$$

where $\delta \longrightarrow 0$ is a small positive constant used to avoid dividing by zero in computing the value of the function.

B. Constraints

Routing constraint (1) ensures that each node $u \in U^c$ is mapped to exactly one physical node $i \in N$. Constraint (2) instructs the algorithm to verify that the definition of $y^c_{k,l,i,j,u,v}$ is applied based on the chains in the service request. Constraint (3) ensures the path created for pair (u,v) starts at exactly one edge going out from node i to where VSNF (or start/endpoint) u is mapped. Similarly, (4) ensures the correctness and the uniqueness of the last edges in the path. Constraints (2-4) can be easily linearized. Constraint (5) is the classical *flow conservation constraint*. That is, an outbound flow equals an inbound flow for each intermediate node l (intermediate nodes cannot consume the flow).

$$\textstyle\sum_{\{i \in N\}} x^c_{i,u} = 1 \quad \forall c \in C_s, \forall u \in U^c \tag{1}$$

$$y^c_{k,l,i,j,u,v} \leq x^c_{i,u} \cdot x^c_{j,v} \quad \forall c \in C_s, \forall i,j \in N, \forall (u,v) \in U^c_{pairs}, \forall (k,l) \in E \tag{2}$$

$$\textstyle\sum_{\{(i,k) \in E, j \in N\}} y^c_{i,k,i,j,u,v} \cdot x^c_{i,u} \cdot x^c_{j,v} = 1 \quad \forall c \in C_s, \forall i \in N, \forall (u,v) \in U^c_{pairs} \tag{3}$$

$$\textstyle\sum_{\{(k,j) \in E, i \in N\}} y^c_{k,j,i,j,u,v} \cdot x^c_{i,u} \cdot x^c_{j,v} = 1 \quad \forall c \in C_s, \forall j \in N, \forall (u,v) \in U^c_{pairs} \tag{4}$$

$$\sum_{\substack{k \in N \\ (k,l) \in E}} y^c_{k,l,i,j,u,v} = \sum_{\substack{m \in N \\ (l,m) \in E}} y^c_{l,m,i,j,u,v} \quad \forall c \in C_s, \forall i,j \in N, \forall l \in N, l \neq i, l \neq j, \forall (u,v) \in U^c_{pairs} \tag{5}$$

Resource constraints (6-8) ensure that the resources consumed by a security service do not exceed the available bandwidth, computational and memory capacities.

$$\sum_{\substack{c \in C_s,\ i,j \in N \\ (u,v) \in U^c_{pairs}}} y^c_{k,l,i,j,u,v} \cdot \beta^c \leq \beta'_{k,l} \quad \forall (k,l) \in E \tag{6}$$

$$\textstyle\sum_{\{c \in C_s, u \in V^c\}} x^c_{i,u} \cdot \gamma^c_v \leq \gamma'_i \quad \forall i \in N \tag{7}$$

$$\textstyle\sum_{\{c \in C_s, u \in V^c\}} x^c_{i,u} \cdot \mu^c_v \leq \mu'_i \quad \forall i \in N \tag{8}$$

QoS constraint (9) verifies that the requirements in terms of maximum end-to-end latency are met. It takes into consideration the propagation delay of physical links and the processing delay of VSNFs. Note that the minimum bandwidth requirement is verified against the bandwidth resource constraint (6).

$$\sum_{i \in N, u \in V^c} x^c_{i,u} \cdot \lambda^c_{i,u} + \sum_{\substack{i,j \in N,(k,l) \in E \\ (u,v) \in U^c_{pairs}}} y^c_{k,l,i,j,u,v} \cdot \lambda_{k,l} \leq \lambda^c \quad \forall c \in C_s \qquad (9)$$

Security constraints ensure that the TSP's security policies are applied. Specifically, constraint (10) forces a subset of the chains in the request to share the same VSNF instance in case of stateful flow processing. Constraint (11) forces the algorithm to place the VSNF or start/endpoint $u \in U^c$ in a specific region of the network defined as a subset $M_u \subset N$.

$$x^{c_1}_{u,i} = x^{c_2}_{u,i} \quad \forall c_1, c_2 \in D_s \subset C_s, i \in N, u \in V^c \qquad (10)$$

$$\textstyle\sum_{\{i \in M_u\}} x^c_{i,u} = 1 \quad M_u \subset N, |M_u| \geq 1, u \in V^c \qquad (11)$$

In particular, constraint (11) can be used to place a specific VSNF close to the user or on the border of the TSP network. Similarly, the *veto* constraint (12) can be used to prevent the placement of any VSNFs on a pre-defined subset of nodes $M \subset N$. A TSP may choose to do this to protect specific nodes that host sensitive data or critical functions from potentially malicious user traffic.

$$\textstyle\sum_{\{i \in M, u \in V^c\}} x^c_{i,u} = 0 \quad \forall c \in C_s, M \subset N, |M| \geq 1 \qquad (12)$$

Finally, for each chain $c \in C_s$, the correct order of VSNFs in V^c is ensured by constraints (1-5), plus constraint (11) applied to the endpoints of the chain in A^c (user and remote applications) with $|M_u| = 1$.

IV. Related work

In the context of network security, only a few solutions have been proposed for the problem of the optimal placement of VSNFs. In [4], the authors propose a model for the placement of VSNFs by taking into account security deployment constraints. Such constraints are necessary to avoid incorrect deployment of security functions such as placing an IDS on an encrypted channel. The authors propose an ILP formulation of the problem and validate their model by measuring the execution time in four different scenarios and by comparing the model with other heuristics in terms of placement cost. However, the proposed optimization algorithm is always computed for all flows in the network, therefore it does not scale well. The authors mitigate the problem by partitioning the network into independent blocks. Nevertheless, the partitioning scheme is limited to fat-tree topologies. Furthermore, the end-to-end latency is not considered among the constraints of the proposed model, which limits its application space.

In [5], the authors provide a model to determine the best placement of security VNFs based on the user requirements and the cost for the network operator. However, the proposed approach does not take into account the specific QoS requirements of the user's applications. This may lead to inefficient deployments where resources are over-provisioned to cover as many application classes as possible. Of greater concern, the proposed model could unnecessarily apply computationally demanding VSNFs (e.g., IDS, Deep Packet Inspection (DPI)) to latency-sensitive traffic (e.g., online gaming), resulting in a significant drop in the user's quality of experience.

The method proposed in [11] is based on light-weight, protocol-specific intrusion detection VNFs. The system dynamically invokes a chain of these IDSs according to the traffic characteristics. The placement of the chains is based on a user-defined or common shortest-path algorithm such as Dijkstra, without consideration of the application QoS requirements or available network/computational resources.

In [12], the authors argue that reactive mechanisms used by cloud providers to deploy VSNFs do not ensure an optimal resource allocation. In this regard, the authors propose a novel resource allocation scheme, which estimates the behavior of the traffic load by monitoring the history of the current VSNFs, and pro-actively provisions new instances of those VSNFs as a countermeasure to any incoming resource pressure. The proposed algorithm does not tackle the problem of VSNF chaining. Instead, it focuses on the optimal placement of new instances of VSNFs, which are part of existing chains. Moreover, it assumes infinite network and computational resources.

V. Conclusions and future work

In this position paper, we have tackled the problem of the optimal placement of security VNFs by taking into account security and QoS requirements of user applications. We have also discussed the rationale behind our design decisions and presented an ILP formulation of the placement problem.

As future work, we will develop a heuristic-based approximation solution of the problem and we will validate it on a NFV-enabled network. In this context, we will also investigate the performance of the proposed algorithm when reconfiguring the provisioned security services, for instance, to support mobile users or for optimization purposes.

Acknowledgment

This work has been partially supported by the EU H2020 ACINO project, Grant Number 645127, www.acino.eu.

References

[1] R. Mijumbi et al., "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 236–262, 2016.
[2] F. Bari et al., "Orchestrating virtualized network functions," *IEEE TNSM*, vol. 13, no. 4, pp. 725–739, 2016.
[3] S. Clayman et al., "The dynamic placement of virtual network functions," in *IEEE NOMS*, May 2014.
[4] A. Shameli Sendi et al., "Efficient provisioning of security service function chaining using network security defense patterns," *IEEE Transactions on Services Computing*, 2017.
[5] C. Basile et al., *Towards the Dynamic Provision of Virtualized Security Services*. Cham: Springer International Publishing, 2015, pp. 65–76.
[6] P. Vizarreta et al., "QoS-driven Function Placement Reducing Expenditures in NFV Deployments," in *IEEE ICC*, 2017.
[7] F. Ben Jemaa et al., "QoS-Aware VNF Placement Optimization in Edge-Central Carrier Cloud Architecture," in *IEEE GLOBECOM*, 2016.
[8] S. Gianvecchio et al., "Humans and bots in internet chat: Measurement, analysis, and automated classification," *IEEE/ACM Trans. Netw.*, vol. 19, no. 5, pp. 1557–1571, Oct. 2011.
[9] J. Yan et al., "A Systematic Classification of Cheating in Online Games," in *ACM SIGCOMM NetGames '05*, 2005.
[10] J. Sherry et al., "BlindBox: Deep Packet Inspection over Encrypted Traffic," in *ACM SIGCOMM '15*, 2015.
[11] Y. Park et al., "Dynamic Defense Provision via Network Functions Virtualization," in *ACM SDN-NFV Security*, 2017.
[12] T. V. Phan et al., "Optimizing resource allocation for elastic security VNFs in the SDNFV-enabled cloud computing," in *IEEE ICOIN*, 2017.