IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 25, 2020, accepted February 11, 2020, date of publication February 17, 2020, date of current version March 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2974569

SDN/NFV-Based Security Service Function Tree for Cloud

JING-LUN LUO, SHUN-ZHENG YU, AND SI-JIE PENG
School of Data and Computer Science, East Campus, Sun Yat-sen University, Guangzhou 510006, China

Corresponding author: Shun-Zheng Yu (syu@mail.sysu.edu.cn)

This work was supported in part by the NSFC under Grant U1636118 and Grant 61972431, and in part by The High Technology Research and Development Program of Guangdong under Grant 2017B030306015 and Grant 2017B010125003.

ABSTRACT Network security for cloud computing is very important. Service function chain (SFC) that integrates software defined network (SDN) and network function virtualization (NFV) can provide a new approach for solving the network security issues for cloud computing. In this paper, we combine multiple SFCs into a security service function tree (or SecSFT, for short) to reduce requirement for resources in allocating virtual security functions. According to the idea of decision tree used for classification, we assign decision rules and detection rules to the nodes of the SecSFT so that they can identify and split suspicious flows from the mixed traffic and detect/prevent intrusions in the suspicious ones. The nodes of the SecSFT implement various virtualized functions including security-related network functions (e.g., load balancing, and traffic shaping), network security functions (e.g., intrusion detection, firewall), and virtualized network security hardware. Finally, we build a SecSFT in an experiment cloud and test and validate its security services in detection and mitigation of network attacks.

INDEX TERMS Cloud, decision tree, intrusion detection, network function virtualization, security service function tree, software defined network.

I. INTRODUCTION

The widespread use of cloud computing services has brought a series of security problems. Due to abstracting and decoupling computing, storage and network resources from dedicated hardware devices, the traditional network security boundaries deployed around the target devices have vanished. The security issues faced by cloud computing come from both internal and external. The internal security threats come from the lack of isolation within the cloud hosts, and external ones come from all directions in the cloud environment. Therefore, security of cloud computing has become a general concern of government, enterprise and academia.

Software defined network (SDN) is an emerging architecture that is manageable, cost-effective and adoptable. It has separate control plane and data plane so that an external controller can manage network traffic in a unified manner [1]. Virtual network function (VNF) [2] was first proposed in the NFV (network function virtualization) white paper published by ETSI (European Telecom Standards Institute) in October 2010. The essence of NFV is to decouple the network functions from dedicated network devices through virtualization. Traditional dedicated network devices like firewall, deep packet inspection, intrusion detection and network address translation, are virtualized as VNFs, which can be deployed throughout a cloud to implement corresponding network security services.

SDN and NFV integration gives power of virtualization and improves network services. Service function chain (SFC) is a mechanism that provides abilities to define an ordered list of service functions and dynamically lead network traffic through various service function paths [3]. Therefore, an SFC can be established to sequentially implement the virtual security functions to provide security services for tenants of the cloud. However, to satisfy a variety of tenants and to prevent all sorts of attacks, a lot of security service function chains are required. Existing works that use SFCs for providing security services over the cloud are still limited in the manner of one SFC per security service [4]–[15]. In this paper, we propose a novel security service function tree architecture (or SecSFT, for short) to overcome this limitation. In the novel architecture, VNF nodes in multiple SFCs are merged to one VNF node if they implement the same network security function.

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

There are a few of related works that focus on the detection and prevention of few types of attacks using an SFC [4]–[7]. Since a full-functioned virtual security function may require a high capacity of computation and the traffic led to it may consume a large amount of network resources, the resource allocation optimization and performance enhancement of an SFC is the main concern [8]–[15]. In contrast, we reduce the requirement for the resources by merging similar VNF nodes of multiple SFCs and allocating the virtual security functions close to targets. The merged SFCs form a distributed decision tree, and the performance of the security services is enhanced due to the distributed processing manner. Since the security situation in cloud environment is complex and network attacks can aim at multiple targets, the distributed decision tree is thus more suitable than a chain.

The main contributions of our paper can be summarized as follows:

(1) In considering that a lot of security service function chains are required to satisfy a variety of tenants and to prevent all sorts of attacks on a cloud, and most of VNF nodes among the chains implement the same functions with very low reuse rates, we propose a novel SecSFT architecture. The same functioned VNF nodes are merged as many as possible, and the tree/trees are optimally deployed in the sense of resource consumption and efficient delivering and filtering of network traffic.

(2) We propose a distributed model for the SecSFT according to the decision tree classification algorithm, which associates decision rules to the corresponding VNF nodes of the SecSFT. Network flow attribute values are collected and analyzed in each of the VNF nodes. The network flows are identified and divided by matching the decision rules, and suspicious network flows are detected and filtered at the current node or forwarded to the next nodes for more fine-grained division and detection.

II. THE RELATED WORK

There are a few of the related works using SFC to provide security services for the networks. For example, Xing *et al.* [4] proposed an intrusion detection system framework, SnortFlow, using a combination of OpenFlow and Snort in a cloud environment. Snort is used to identify intrusion behaviors in network messages. SDN controller is responsible for distributing flow tables to reconfigure the network. Phan and Park [5] proposed a solution to tackle DDoS attacks in the SDN-based cloud environment. The traffic classification is made based on the support vector machine and self-organizing map algorithms, and the attack detection is performed by an enhanced history-based IP filtering scheme. An SFC was formed in an SDN-based cloud to defend different level DDoS attack. Nguyen *et al.* [6] used DES, AES and other encryption algorithms to encrypt the message headers that contains the forwarding path information in the SFC messages, thereby preventing information eavesdropping or threat of man-in-the-middle attack. Li *et al.* [7] proposed an automatic selection scheme for security service function chain that uses Q-learning reinforcement learning algorithm to analyze and weight the network states, so as to achieve a diversified network defense solution. In contrast to the existing works that use one SFC to provide a security service, we use a tree that combines multiple SFCs to provide a collection of security services for the cloud.

Different from the fewer works focusing on security issues in security service chaining, the major works in this area concern the optimization of resource allocation for the security service function chains, and their performance enhancement. For example, Ye [8] proposed a scheme that lets every two VNFs share one server to minimize bandwidth resource consumption. Lin *et al.* [9] constructed a game model to coordinate the deployment of network service. Liu *et al.* [10] proposed a system architecture solution that uses SDN and NFV to control virtualized security resources for security service function chain construction. Dwiardhika and Tachibana [11] proposed an optimal placement of security virtual network functions for the security service function chains based on the security level. Shameli-Sendi *et al.* [12] proposed a security approach for cloud infrastructure that incorporates the best practice, know-how of the security experts, and various security constraints into a network security pattern. The optimal placement of compliant security functions in the cloud is believed to be a NP-Hard problem, and so a scalable networking and computing resources aware optimization framework was proposed in the work. A heuristic solution based on the breadth first search algorithm was proposed by Liu *et al.* [13] to optimize the resource allocation with the constraints that do not violate security and resource requirements. Pei *et al.* [14] studied the dynamic VNF placement in geo-distributed cloud system. It is formulated as a Binary Integer Programming model to minimize the embedding cost in the embed SFC requests, and optimize the number of placed VNF instances. Li *et al.* [15] realized context-based and dynamic SFC over multi-domain networks by allocating metadata to share context information of packets among those networks. In contrast to the existing works, we use the tree architecture to reduce the resource consumption in duplicated deploying of same-functioned virtual security functions.

III. ARCHITECTURE OF SECURITY SERVICE FUNCTION TREE

In this section, we propose a security service function tree (SecSFT) architecture that combines multiple SFCs to provide a variety of security services and to reuse security resources. It allows to be flexibly deployed close to multiple targets to detect and prevent network attacks coming from multiple directions.

A. DECISION TREE

We design the service function tree based on the decision tree to classify the network attack traffic. Correspondingly assign the rules of the decision tree to VNF nodes of the SecSFT. Specifically, C4.5 decision tree algorithm [16] is

used to construct the SecSFT. C4.5 algorithm determines the attribute value of each internal node according to Information Gain Ratio. To avoid over-fitting, we adopt Pessimistic Error Pruning (PEP) algorithm [17]. PEP doesn't need extra test data set, pruning from top to bottom. For a leaf node numbered i with n_i samples and e_i errors, the error rate was $(e_i+0.5)/n_i$, where 0.5 is the penalty factor. Then for a subtree with L leaf nodes, its misjudgment rate is:

$$ErrorRatio = \left(\sum_{i=1}^{L} e_i + 0.5L\right) \Big/ \sum_{i=1}^{L} n_i$$

When *ErrorRatio* of a sample equals 1, it means that the subtree has misclassified the sample; when the value equals 0, the sample is properly classified. The number of misjudgments of the subtree can be expressed by Bernoulli distribution. The mean and standard deviation of the misjudgment for the subtree are:

$$ErrorMean = ErrorRatio \times \sum_{i=1}^{L} n_i$$

$$ErrorSTD = \sqrt{ErrorRatio \times \sum_{i=1}^{L} n_i \times (1 - ErrorRatio)}$$

Replace the subtree with a leaf node, then the error rate of the leaf node is:

$$ErrorRatio' = \left(e' + 0.5\right)/n', \tag{1}$$

where $e' = \sum_{i=1}^{L} e_i$, $n' = \sum_{i=1}^{L} n_i$, and the mean of the number of misjudgments of the leaf node is:

$$ErrorMean' = ErrorRatio' \times n'.$$

Therefore, when the original subtree meets the following conditions, the subtree will be pruned.

$$ErrorMean + ErrorSTD \geq ErrorMean'.$$

B. CONSTRUCTION OF SecSFT

The SecSFT architecture has a control and management plane which consists of orchestration and global monitoring modules. The global monitoring module mainly provides a global perspective for control and management, collects the entire network topology, finds network-wide available computing resources, and obtains the entire network security status. The orchestration module mainly formulates orchestration schemes of SecSFTs according to the current security situation, performs VNF resource mapping and scheduling according to the orchestration schemes and available resources, and defines flow tables to construct the tree topologies so that data flows are transmitted through the SecSFTs.

The SDN controller plays an important role in the control and management plane. The SDN controller uses OpenFlow [18] network protocol to communicate with networking devices. There are several popular SDN controllers, such as POX, Floodlight, OpenDayLight, ONOS, etc. By taking account of the performance among those controllers, ONOS is selected in this paper. ONOS has the advantages of open source ecosystem, supporting controller cluster environment, and full open programmability.

The SDN controller can send different flow tables to OpenFlow switches according to the types of security services and the tree topologies. When network traffic arrives at the entry of a SecSFT, it will be led to different branches of the tree according to the decision rules.

The SecSFT architecture uses JSON (JavaScript Object Notation) as the standard format for data storage and exchange. Its text record mode has the advantages of clear structure and concise layers. It can be used for network-wide topological structure records, updates of infrastructure resources, and templates of OpenFlow flow tables, etc. Restful API is utilized to achieve real-time data communication between modules.

We construct a virtual network for each SFC to ensure that network traffic passing through different chains can be isolated by identifying the virtual logic network IDs. To detect and prevent network attacks, security resources of the cloud are deployed near the potential attack targets and integrated into the SecSFT. As Figure 1 shows, two SFCs are deployed in the same cloud across multiple network domains. The first two VNFs in the two SFCs shown in Figure 1 have the same network security functions, and so they are merged into the same VNF.

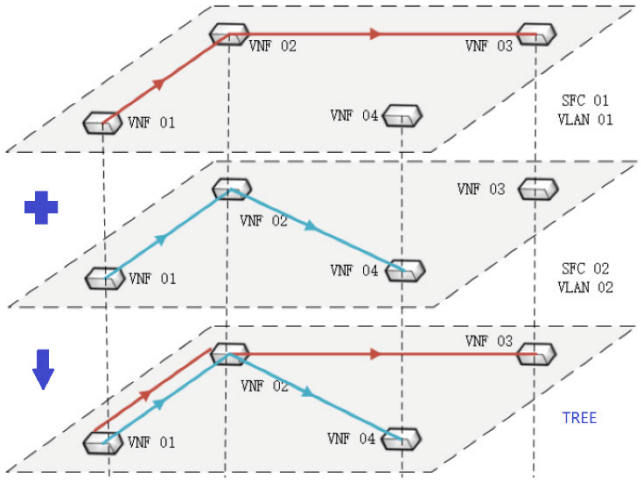


FIGURE 1. Two chains are merged to a tree.

The merging approach is illustrated in Figure 2. We create several virtual network cards (vNICs) for the merged VNF. Virtual network ports (vPorts) corresponding to these vNICs are mounted on the virtual bridge of the virtual switch. VLAN IDs are set for the virtual logic subnets that are connected to the vPorts.

When a frame arrives from the inbound port to a VNF, VLAN ID in its header is stripped at vPort. After the frame is processed in the VNF, one vPort is selected as the forwarding

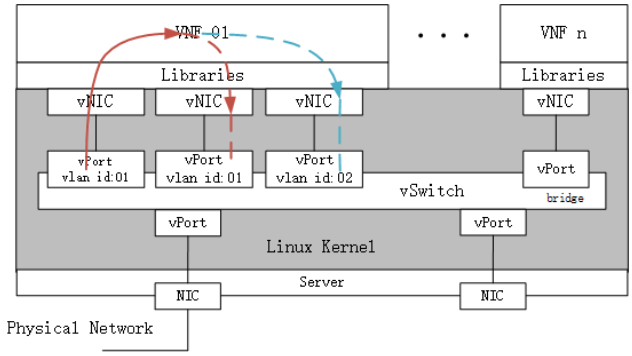


FIGURE 2. Different virtual logic networks are bridged in the same VNF.

port, and a new VLAN ID belonging to the next virtual network is added to the header of the frame before it is forwarded.

C. COLLECTION AND CLASSIFICATION OF TRAFFIC

Typical network attack types include DoS (Denial of Service), port scanning, unauthorized remote access and unauthorized super user privilege access, etc. In this paper, we choose four specific network attacks, SYN Flood, UDP Flood, IPsweep and Portscan, to test the SecSFT architecture.

When a new network attack arrives, the network traffic attribute values will be significantly different from normal network traffic at some network nodes. Referring to the network intrusions provided by KDDCUP 99 [19], there are up to 41 attributes of network intrusion behaviors. For the four specific network attacks we choose, eleven of the 41 attributes are most suitable for the SecSFT to construct the decision tree, as listed in Table 1.

The attributes have varying values over time. By using time-based statistical attribute values, the abnormal behavior of network attacks can be dynamically captured. In this paper, a 1-second sliding window is used for collecting the flow attribute values.

Network flow attribute values are collected in the VNF nodes corresponding to those of the decision tree. By collecting the attribute values of the network traffic passing through a VNF node in the time window and matching them with the decision rules of the VNF node, a decision can be made that determines whether a flow is forwarded to the next hop or stopped to process locally. For the flow to be processed locally, intrusion detection/prevention rules corresponding to the virtual security functions of the current VNF are applied. Though we can combine the decision rules and the intrusion detection rules to determine forwarding or dropping a flow, the matching time against a lot of rules may cause additional delay for the traffic.

We use open source libpcap (a network packet capture function library) to sniff the network traffic delivering to specific VNF nodes, statistic the relative attribute values, and forward the traffic to the virtual networks. The virtualized network functions in this paper use Docker as the container.

TABLE 1. Attributes selected for secsft to detect the attacks.

	Attribute	Attribute description	Attack
1.	pkt_all_count	Total number of packets arriving at the VNF in the time window	Flooding DDoS
2.	pkt_all_bytes	Total bytes of packets arriving at the VNF in the time window	
3.	src_port_diff_rate	Relative rates of packets from different source ports arriving at the VNF in the time window	
4.	src_host_diff_rate	Relative rates of packets from different source hosts arriving at the VNF in the time window	
5.	dst_port_diff_rate	Relative rates of packets to different destination ports arriving at the VNF in the time window	Flooding DDoS, Probing
6.	dst_host_diff_rate	Relative rates of packets from different source hosts arriving at the VNF in the time window	Probing
7.	tcp_rate	Rate of TCP packets arriving at the VNF in the time window	SYN Flooding
8.	syn_rate	Rate of SYN bits in TCP packets arriving at the VNF in time window	
9.	udp_rate	Rate of UDP packets arriving at the VNF in the time window	UDP Flooding
10.	icmp_rate	Rate of ICMP packets arriving at the VNF in the time window	Probing, ICMP Flooding
11.	avg_pkt_rate	Average rate of incoming packets arriving at the VNF in the time window	Flooding

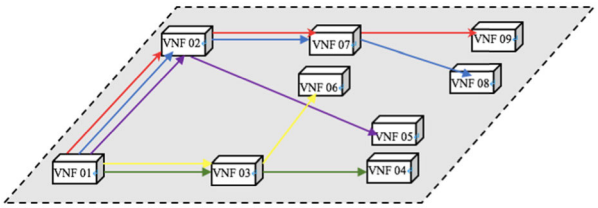


FIGURE 3. Topology of service function tree.

An example of the SecSFT is shown in Figure 3. Different color lines represent different SFCs, and each chain provides a set of specific security services. VNF 01 is the root of the tree. When network traffic arrives at the entry of the tree, it is subdivided into smaller and smaller streams while they pass through the tree.

D. COMMUNICATION BETWEEN VNFs

For communication between VNFs in the virtual environment, we use OVS as multi-layer virtual switch. To ensure high availability and scalability, an OVS double bridge architecture scheme is designed. We construct two OVS virtual bridges on a host. The br-int virtual bridge mainly completes the adding and stripping of the VLAN IDs of the virtual networks and the normal forwarding operation of data packets. However, in the br-tun virtual bridge, OpenFlow multi-level flow table [20] is adopted to group and process data packets from different sources. The header of each data packet is matched with the matching field of the flow table of OpenFlow protocol. The corresponding operation such

as forwarding, discarding and modifying, are executed if the match is successful. OVS establishes a pair of patch ports between br-int and br-tun to realize the delivery of the packets.

To meet the requirement of communication among VNFs distributed in multiple data centers, we use overlay network to maintain multiple data centers in a broadcast domain. The broadcast packets from the VNFs can reach all the data centers, realizing a large layer-2 extended network. It allows all servers, containers, virtual machines, etc. to communicate within the scope of the large layer-2 extended network.

VxLAN (Virtual Extensible LAN) [21] is used to abstract the underlay physical network, build up virtual tunnel and a large layer-2 virtual network, and to realize layer-2 message transmission across layer-3 networks.

IV. EXPERIMENT OF SecSFT

We used six servers and several switching devices. One of the servers is used as the control center of the SecSFT. We chose Docker container to implement VNF and to manage virtualized security resources. The multi-layer virtual switch scheme of OpenvSwitch is adopted to construct virtual network of containers. The open source SDN operating system ONOS was deployed in the control center. The experiment cloud and the SecSFT are shown in Figure 4.

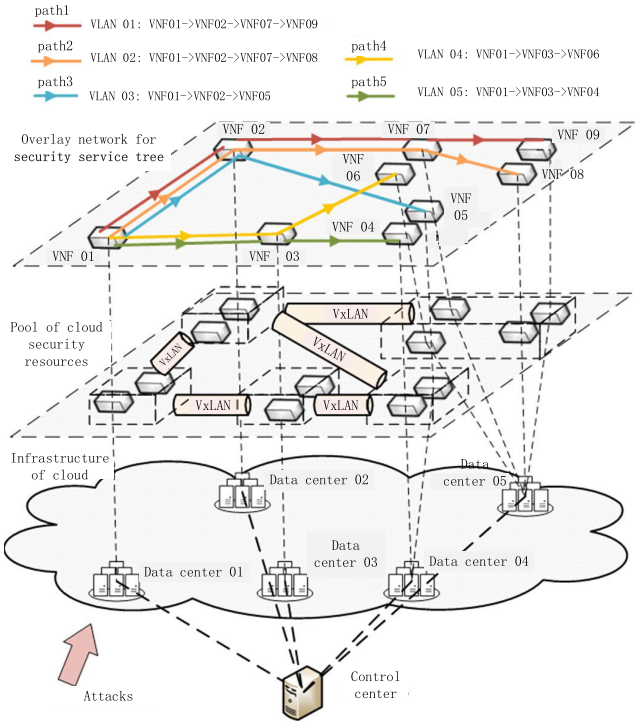


FIGURE 4. The experiment cloud and the SecSFT.

The experiment protype uses VxLAN tunnels to abstract the underlay physical network and construct a large layer-2 virtual network. Isolation among different types of network traffic is achieved by dividing VLANs. The containers communicate with each other through the virtual network.

The construction and training of the SecSFT needs sample sets of corresponding network attacks traffic. We collected a variety of flow attribute values of the network attack flows and stored them in a database. In the experiment, we selected 500 training samples for each type of attack.

Four types of attacks were used to verify the SecSFT. SYN Flood and UDP Flood are selected from the DDoS attack types, and IPsweep and Portscan are selected from the scanning attack types.

We generated SYN Flood, UDP Flood, IPsweep and Portscan attacks through the raw socket of Linux. Normal traffic is obtained by collecting the network traffic in the real network and re-launching it to the experiment cloud. The network traffic attribute values are extracted and analyzed in the VNF nodes of the SecSFT.

The network traffic attribute values versus the four network attacks were collected in the controller, as shown in Figure 5 and Figure 6.

```
[root@controller FlowsAttrSample]# ./flow_sample_loop
success: device: enol
loop: 1
-----------------------------------
avg_bandwidth:1.3896MB/s
icmp_rate:0.000077
udp_rate:0.004373
tcp_rate:0.995395
syn_rate:0.997628
src_port_diff_rate:0.824265
dst_port_diff_rate:0.001161
src_host_diff_rate:0.991950
dst_host_diff_rate:0.000426
```

```
[root@controller FlowsAttrSamp
success: device: enol
loop: 1
-----------------------------------
avg_bandwidth:5.7629MB/s
icmp_rate:0.000027
udp_rate:0.999043
tcp_rate:0.000670
syn_rate:0.142857
src_port_diff_rate:0.602920
dst_port_diff_rate:0.000314
src_host_diff_rate:0.995598
dst_host_diff_rate:0.000178
```

FIGURE 5. The traffic attribute values vs. SYN Flood and UDP Flood.

```
[root@controller FlowsAttrSample]# ./flow_sample_loop
success: device: enol
loop: 1
-----------------------------------
avg_bandwidth:0.0326MB/s
icmp_rate:0.685714
udp_rate:0.302857
tcp_rate:0.011429
syn_rate:0.000000
src_port_diff_rate:0.148571
dst_port_diff_rate:0.062857
src_host_diff_rate:0.160000
dst_host_diff_rate:0.720000
```

```
[root@controller FlowsAttrSample
success: device: enol
loop: 1
-----------------------------------
avg_bandwidth:0.0582MB/s
icmp_rate:0.003565
udp_rate:0.133690
tcp_rate:0.862745
syn_rate:1.000000
src_port_diff_rate:0.055258
dst_port_diff_rate:0.882353
src_host_diff_rate:0.058824
dst_host_diff_rate:0.017825
```

FIGURE 6. The traffic attribute values vs. IPsweep and Portscan.

To verify whether the SecSFT can correctly identify and process the network attack traffic, we respectively send the Normal network traffic, SYN Flood attack traffic, UDP Flood attack traffic, IPsweep attack traffic, and Portscan attack traffic to the root node of the SecSFT. In the virtual resource global monitoring module of the control center, we check the packet rate delivering into each VNF node in the SecSFT. The rates are used to show different network traffic paths, and to verify whether the network traffic is forwarded to more fine-grained branches based on the decision rules.

As an example of the results, the flow of the SYN Flood attack is shown in Figures 7 and 8. We can find that a chain of VNFs in the service function tree is VNF01 $->$ VNF03 $->$ VNF06. The intrusion detection alarm for the SYN Flood attack is issued by VNF06, as shown in Figure 8.

Similarly, as shown in Figures 9-14, different attacks are led to different chains of VNFs of the SecSFT and alarms for

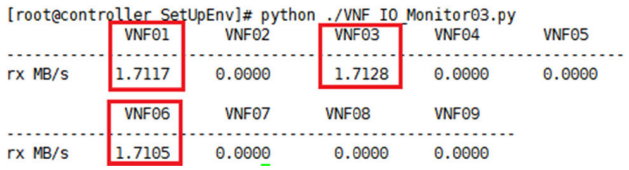


FIGURE 7. The SYN flood led to a chain of VNFs.

```
04/18-19:27:34.404129 [**] [1:29292:0] "SYN FLOOD Attack!!!" [**] [Priority: 0]
{TCP} 124.209.175.82:25515 -> 192.168.5.10:23333
04/18-19:27:34.692761 [**] [1:29292:0] "SYN FLOOD Attack!!!" [**] [Priority: 0]
{TCP} 121.104.242.187:2428 -> 192.168.5.10:23333
04/18-19:27:34.738554 [**] [1:29292:0] "SYN FLOOD Attack!!!" [**] [Priority: 0]
{TCP} 123.255.111.92:33949 -> 192.168.5.10:23333
04/18-19:27:34.867645 [**] [1:29292:0] "SYN FLOOD Attack!!!" [**] [Priority: 0]
{TCP} 57.222.7.76:62432 -> 192.168.5.10:23333
04/18-19:27:35.054186 [**] [1:29292:0] "SYN FLOOD Attack!!!" [**] [Priority: 0]
{TCP} 72.13.172.216:31715 -> 192.168.5.10:23333
04/18-19:27:35.190925 [**] [1:29292:0] "SYN FLOOD Attack!!!" [**] [Priority: 0]
{TCP} 126.126.48.41:36816 -> 192.168.5.10:23333
04/18-19:27:35.261180 [**] [1:29292:0] "SYN FLOOD Attack!!!" [**] [Priority: 0]
{TCP} 73.133.93.28:12405 -> 192.168.5.10:23333
04/18-19:27:35.662744 [**] [1:29292:0] "SYN FLOOD Attack!!!" [**] [Priority: 0]
{TCP} 89.225.37.164:50350 -> 192.168.5.10:23333
```

FIGURE 8. The alarm for SYN flood issued in VNF06.

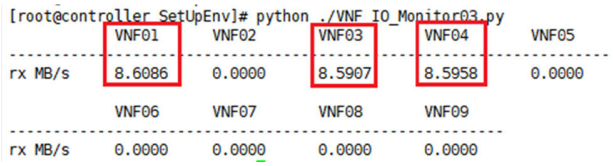


FIGURE 9. The UDP flood led to a chain of VNFs.

```
04/18-18:58:55.559510 [**] [1:24323:0] "UDP FLOOD Attack!!!" [**] [Priority: 0]
{UDP} 236.139.235.8:9290 -> 192.168.5.10:23333
04/18-18:58:55.559515 [**] [1:24323:0] "UDP FLOOD Attack!!!" [**] [Priority: 0]
{UDP} 20.90.165.60:57184 -> 192.168.5.10:23333
04/18-18:58:55.559562 [**] [1:24323:0] "UDP FLOOD Attack!!!" [**] [Priority: 0]
{UDP} 146.159.66.92:53309 -> 192.168.5.10:23333
04/18-18:58:55.559784 [**] [1:24323:0] "UDP FLOOD Attack!!!" [**] [Priority: 0]
{UDP} 113.44.133.8:58979 -> 192.168.5.10:23333
04/18-18:58:55.559833 [**] [1:24323:0] "UDP FLOOD Attack!!!" [**] [Priority: 0]
{UDP} 174.96.161.49:52487 -> 192.168.5.10:23333
04/18-18:58:55.559833 [**] [1:24323:0] "UDP FLOOD Attack!!!" [**] [Priority: 0]
{UDP} 112.217.189.20:28702 -> 192.168.5.10:23333
04/18-18:58:55.559937 [**] [1:24323:0] "UDP FLOOD Attack!!!" [**] [Priority: 0]
{UDP} 75.37.11.7:31669 -> 192.168.5.10:23333
04/18-18:58:55.559977 [**] [1:24323:0] "UDP FLOOD Attack!!!" [**] [Priority: 0]
{UDP} 154.144.51.75:1676 -> 192.168.5.10:23333
```

FIGURE 10. The alarm for UDP flood issued in VNF04.

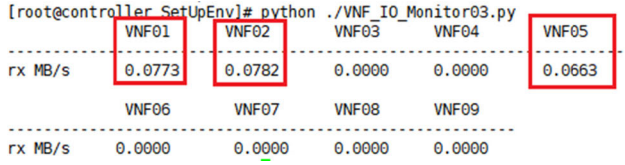


FIGURE 11. The IPsweep led to a chain of VNFs.

the attacks are issued, respectively, by corresponding security VNFs. In contrast, the normal traffic passes through the tree without any alarm, as shown in Figure 15.

The traffic mixed with the normal network traffic, SYN Flood attack traffic, UDP Flood attack traffic, IPsweep attack traffic, and Portscan attack traffic are similarly drawn to five different types of network flows after passing through the SecSFT. During the processing of the SecSFT, the attribute

```
04/18-18:57:08.111107 [**] [1:234321:0] "IPsweep Attack!!!" [**] [Priority: 0] {
ICMP} 172.18.218.234 -> 192.168.1.55
04/18-18:57:08.119990 [**] [1:234321:0] "IPsweep Attack!!!" [**] [Priority: 0] {
ICMP} 172.18.218.234 -> 192.168.1.56
04/18-18:57:08.129107 [**] [1:234321:0] "IPsweep Attack!!!" [**] [Priority: 0] {
ICMP} 172.18.218.234 -> 192.168.1.58
04/18-18:57:08.130520 [**] [1:234321:0] "IPsweep Attack!!!" [**] [Priority: 0] {
ICMP} 172.18.218.234 -> 192.168.1.59
04/18-18:57:08.137419 [**] [1:234321:0] "IPsweep Attack!!!" [**] [Priority: 0] {
ICMP} 172.18.218.234 -> 192.168.1.61
04/18-18:57:08.145577 [**] [1:234321:0] "IPsweep Attack!!!" [**] [Priority: 0] {
ICMP} 172.18.218.234 -> 192.168.1.62
04/18-18:57:08.163271 [**] [1:234321:0] "IPsweep Attack!!!" [**] [Priority: 0] {
ICMP} 172.18.218.234 -> 192.168.1.57
04/18-18:57:08.171879 [**] [1:234321:0] "IPsweep Attack!!!" [**] [Priority: 0] {
ICMP} 172.18.218.234 -> 192.168.1.60
```

FIGURE 12. The alarm for IPsweep issued in VNF05.

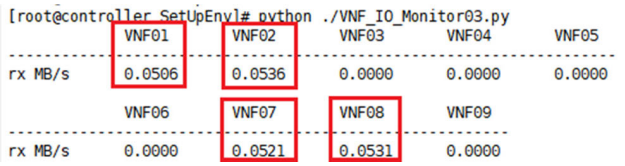


FIGURE 13. The Portscan led to a chain of VNFs.

```
04/18-18:53:58.122201 [**] [1:77622:0] "Portscan Attack!!!" [**] [Priority: 0] {
} 172.18.218.234:20 -> 192.168.5.10:243
04/18-18:53:58.124321 [**] [1:77622:0] "Portscan Attack!!!" [**] [Priority: 0] {
} 172.18.218.234:20 -> 192.168.5.10:244
04/18-18:53:58.126415 [**] [1:77622:0] "Portscan Attack!!!" [**] [Priority: 0] {
} 172.18.218.234:20 -> 192.168.5.10:245
04/18-18:53:58.128519 [**] [1:77622:0] "Portscan Attack!!!" [**] [Priority: 0] {
} 172.18.218.234:20 -> 192.168.5.10:246
04/18-18:53:58.130477 [**] [1:77622:0] "Portscan Attack!!!" [**] [Priority: 0] {
} 172.18.218.234:20 -> 192.168.5.10:247
04/18-18:53:58.132604 [**] [1:77622:0] "Portscan Attack!!!" [**] [Priority: 0] {
} 172.18.218.234:20 -> 192.168.5.10:248
04/18-18:53:58.134545 [**] [1:77622:0] "Portscan Attack!!!" [**] [Priority: 0] {
} 172.18.218.234:20 -> 192.168.5.10:249
04/18-18:53:58.136799 [**] [1:77622:0] "Portscan Attack!!!" [**] [Priority: 0] {
} 172.18.218.234:20 -> 192.168.5.10:250
```

FIGURE 14. The alarm for Portscan issued in VNF08.

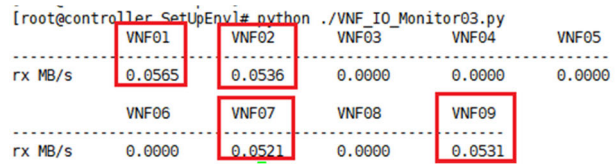


FIGURE 15. The Normal traffic led to a chain of VNFs.

values of the network flows passing through the VNF nodes are collected and matched with the decision rules, and undetermined ones are forwarded to the next hop for finer-grained check. Each VNF chain in the SecSFT perform a sort of intrusion detection and prevention.

Finally, we use *accuracy*, *false alarm*, and *miss alarm* to evaluate the detection results of the SecSFT. The *accuracy* reflects the proportion in the samples of the mixed flows that correctly identifies the attack and the normal traffic, the *false alarm* reflects the proportion in the samples of normal traffic that is falsely identified as the attack, and the *miss alarm* reflects the proportion in the samples of the attack that is falsely identified as the normal traffic.

The confusion matrix for evaluating the detection results of the SecSFT against attacks is shown in Table 2, and

TABLE 2. Confusion matrix of detection results for each attack type.

		output attack types				
		Normal	SYN Flood	UDP Flood	IPsweep	Portscan
actual attack type	Normal	500	0	0	0	0
	SYN Flood	9	491	0	0	0
	UDP Flood	14	0	486	0	0
	IPsweep	0	0	0	493	7
	Portscan	49	0	0	0	441

TABLE 3. The accuracy, false alarm rate and miss alarm rate of the tree.

Network Traffic Type	Accuracy	False Alarm	Miss Alarm
Normal	1.000	0.000	0.000
SYN Flood	0.982	0.151	0.018
UDP Flood	0.972	0.000	0.028
IPsweep	0.986	0.000	0.014
Portscan	0.882	0.030	0.118

the *accuracy*, *false alarm*, and *miss alarm* are shown in Table 3.

The evaluation results show that the SecSFT designed in this paper has a high identification accuracy for the four types of network attacks, and the system has a good intrusion detection capability.

V. CONCLUSION

In this paper, we designed an architecture of distributed decision tree based on SDN/NFV and SFC for security services in a cloud. We deployed the SecSFT in an experimental cloud and successfully implemented it in classification, detection and filtering of four types of network attacks. Finally, we evaluated it with three performance indicators.

For the future work, we will evaluate SecSFT with more types of attacks and compare it with the existing attack detection schemes.

REFERENCES

[1] A. Prajapati, A. Sakadasariya, and J. Patel, "Software defined network: Future of networking," in *Proc. 2nd Int. Conf. Inventive Syst. Control (ICISC)*, Coimbatore, India, Jan. 2018, pp. 1351–1354.

[2] (Jan. 2020). *ETSI Network Functions Virtualization (NFV)*. [Online]. Available: https://www.etsi.org/technologies/689-network-functions-virtualisation

[3] (Jan. 2020). *IETF Service Function Chaining*. [Online]. Available: https://tools.ietf.org/wg/sfc/

[4] T. Xing, D. Huang, L. Xu, C.-J. Chung, and P. Khatkar, "SnortFlow: A OpenFlow-based intrusion prevention system in cloud environment," in *Proc. 2nd GENI Res. Educ. Exp. Workshop*, Mar. 2013, pp. 89–92.

[5] T. V. Phan and M. Park, "Efficient distributed denial-of-service attack defense in SDN-based cloud," *IEEE Access*, vol. 7, pp. 18701–18714, 2019.

[6] V.-C. Nguyen, A.-V. Vu, K. Sun, and Y. Kim, "An experimental study of security for service function chaining," in *Proc. 9th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2017, pp. 797–799.

[7] G. Li, H. Zhou, B. Feng, G. Li, and S. Yu, "Automatic selection of security service function chaining using reinforcement learning," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–6.

[8] Z. Ye, "Efficient, scalable and reliable network (function) virtualization in software-defined optical networks," Ph.D. dissertation, California State Univ., Los Angeles, Los Angeles, CA, USA, Jun. 2015. [Online]. Available: https://ubir.buffalo.edu/xmlui/handle/10477/51473

[9] T. Lin, Z. Zhou, M. Tornatore, and B. Mukherjee, "Demand-aware network function placement," *J. Lightw. Technol.*, vol. 34, no. 11, pp. 2590–2600, Jun. 1, 2016.

[10] Y. Liu, Z. Guo, G. Shou, and Y. Hu, "To achieve a security service chain by integration of NFV and SDN," in *Proc. 6th Int. Conf. Instrum. Meas., Comput., Commun. Control (IMCCC)*, Harbin, China, Jul. 2016, pp. 21–23.

[11] D. Dwiardhika and T. Tachibana, "Optimal construction of service function chains based on security level for improving network security," *IEEE Access*, vol. 7, pp. 145807–145815, 2019.

[12] A. Shameli-Sendi, Y. Jarraya, M. Pourzandi, and M. Cheriet, "Efficient provisioning of security service function chaining using network security defense patterns," *IEEE Trans. Services Comput.*, vol. 12, no. 4, pp. 534–549, Jul. 2019.

[13] Y. Liu, Y. Lu, W. Qiao, and X. Chen, "A dynamic composition mechanism of security service chaining oriented to SDN/NFV-enabled networks," *IEEE Access*, vol. 6, pp. 53918–53929, 2018.

[14] J. Pei, P. Hong, K. Xue, and D. Li, "Efficiently embedding service function chains with dynamic virtual network function placement in geo-distributed cloud system," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 10, pp. 2179–2192, Oct. 2019.

[15] G. Li, H. Zhou, B. Feng, and G. Li, "Context-aware service function chaining and its cost-effective orchestration in multi-domain networks," *IEEE Access*, vol. 6, pp. 34976–34991, 2018.

[16] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 2014.

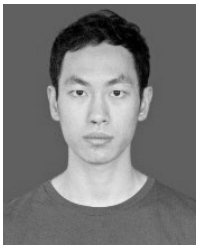
[17] P. Kapoor and R. Rani, "Efficient decision tree algorithm using J48 and reduced error pruning," *Int. J. Eng. Res. Gen. Sci.*, vol. 3, no. 3, pp. 1613–1621, 2015.

[18] Y. Li, D. Zhang, J. Taheri, and K. Li, "SDN components and OpenFlow," in *Big Data and Software Defined Networks*, J. Taheri, Ed. Edison, NJ, USA: IET, Mar. 2018, ch. 3, pp. 49–67.

[19] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, "Selecting features for intrusion detection: A feature relevance analysis on KDD 99 benchmark," in *Proc. 3rd Annu. Conf. Privacy, Secur. Trust*, Oct. 2005, pp. 12–14.

[20] L. Dong, L. Chen, B. He, and W. Wang, "The research on designs of multiple flow tables in the Openflow protocol," in *Proc. 27th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Hangzhou, China, Jul. 2018, pp. 1–2.

[21] *Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks Over Layer 3 Networks*, document RFC 7348, 2014. [Online]. Available: https://tools.ietf.org/html/rfc7348



JING-LUN LUO received the B.S. degree in electronics and information system from South China Normal University, Guangzhou, China, in 2016, and the M.S. degree in radio physics from Sun Yat-sen University, Guangzhou, in 2019.

From 2016 to 2019, he was a Research Assistant with the School of Data and Computer Science, Sun Yat-sen University. His research interests include security service for cloud, software-defined networks, network function virtualization, security service function tree, and intrusion detection.

SHUN-ZHENG YU received the B.S. degree in radio physics from Peking University, China, in 1982, and the M.S. and Ph.D. degrees in communication and information system from the Beijing University of Posts and Telecommunications, China, in 1988.

Since 1989, he has been a Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. From 1999 to 2000, he was a Visiting Scholar with Princeton University. From 2000 to 2002, he was with the IBM Thomas J. Watson Research Center. He is the author of one book, more than 200 articles, and more than 40 inventions. His research interests include anomaly detection, software-defined network-based intrusion detection/prevention, traffic analysis and control for green networking of cloud/data center, protocol reverse engineering, and machine learning.

SI-JIE PENG is currently pursuing the B.S. degree in information security with Sun Yat-sen University, Guangzhou, China, where she will pursue the M.S. degree in computer engineering.