Proposing and Evaluating the Performance of a Firewall Implemented as a Virtualized Network Function

Leopoldo A. F. Mauricio*, Marcelo G. Rubinstein†, and Otto C. M. B. Duarte*
*Grupo de Teleinformática e Automação - Universidade Federal do Rio de Janeiro - COPPE/UFRJ, Brazil
†Universidade do Estado do Rio de Janeiro - FEN/DETEL/PEL, Brazil
‡Globo.com, Brazil
Emails: leopoldo@corp.globo.com, rubi@uerj.br, otto@gta.ufrj.br

***Abstract*—The virtualization technology provides many advantages to datacenters such as: reduction of power, cooling, and hardware costs. Moreover, virtualization simplifies administration and maintenance. On the other hand, for assuring security policies, a virtualized datacenter may require a large Access Control List (ACL) that can overload current commercial Top of Rack (ToR) equipment. This paper proposes and evaluates the performance of a firewall implemented as a virtualized network function in the Open source Platform for Network Functions Virtualization (OPNFV) using commercial off-the-shelf servers. Results show that the function provides elastic capacity that can scale up, meeting the current ingress traffic demands.**

I. Introduction

Datacenters are spread worldwide and continue to evolve nowadays. They can benefit from using virtualization to optimize CapEx/OpEx [1] and increase network flexibility [2]. It is possible to reduce expenses related to heat dissipation, electricity consumption, and maintenance by using fewer physical machines. There are also advantages of not being tied down to particular vendors, since Commercial Off-The-Shelf (COTS) hardware can be used.

One of the most used architecture for datacenters is the Fat-tree one [3]. This architecture has a core and pod elements containing aggregation switches, edge switches (ToRs – Top of Racks) and servers. There are trees of routing and switching elements in that network architecture, with progressively more specialized and expensive equipment when moving up the network hierarchy. Thus, the ToR is the minor cost switch/route.

Datacenter servers host different services and many users (tenants) can have several Virtual Machines (VMs) spread over servers and networks that can be located in different racks. Moreover, common services like a MySQL database or a file system can be shared between these networks. Therefore, in order to control the transfer of data between these datacenter networks, security policies need to be applied. To provide a fine-grained security solution, those policies are usually located as ACLs (Access Control Lists) in ToRs, that also act as network gateways to every server located in each of these racks. As these data transfers are not limited to a single rack, the communication requires high link utilization and many ACLs to allow/deny the transfers.

When a cloud solution is implemented to optimize the use of physical resources in the datacenter, it is expected that the amount of Virtual Networks (VNs) and access policies between networks increase. However, it is not possible to increase the processing capacity of ToR access rules without replacing equipment, since this capacity depends on the size of its TCAM. On the other hand, when using routers/virtual firewalls in generic hardware, exploring the Network Functions Virtualization (NFV) paradigm, it is possible to scale up, according to the demand, the processing capacity of access rules in a datacenter. In this scenario, the access rule processing capacity ceases to depend on the size of the TCAM and becomes a function of the amount of memory and CPU allocated to Virtual Network Functions (VNFs). As the processing capability of a group of VMs may be greater than the capacity of a TCAM ToR, such a solution can be employed.

Some researchers work with virtual firewalls. Deng et al. [4] proposed a framework named VNGuard for provisioning and management of virtual firewalls that uses both NFV and SDN concepts. Bi et al. [5] proposed a stateful data plane abstraction system to solve scalability and performance issues of a stateful firewall VNF.

In this context, we propose the use of a distributed FireWall VNF (FW-VNF) to process policies for access between networks. The FW-VNF was implemented in the OPNFV platform (Open Platform for NFV) version 1.0 (https://www.opnfv.org/). Results show that the UDP traffic reception rate can be reduced by 70% when the amount of rules inserted in a FW-VNF is equal to 2000. However, this problem was solved when more FW-VNFs were used. We can infer that, even with 2000 rules, five FW-VNFs are required to support the 900 Mb/s data traffic, higher than the traffic required by each machine currently connected to the ToR.

The rest of the paper is organized as follows. The implementation of the FW-VNF is presented in Section II. The experimental performance evaluation is in Section III. Section IV presents conclusions and future work.

This work was sponsored by CNPq, CAPES, and FAPERJ.

978–1–5090–4671–3/16/$31.00 © 2016 IEEE

II. A Firewall VNF on OPNFV

We propose a firewall as a VNF on the OPNFV platform to process access policies inside virtualized datacenter networks.

OPNFV aims at the integrated implementation of a Virtualized Infrastructure Manager (VIM) and an NFV Infrastructure (NFVI), which are components of the architecture proposed by ETSI [6]. OpenStack (https://www.openstack.org/) is the Virtualized Infrastructure Manager (VIM) of the OPNFV platform that is responsible for the management of NFVI computation resources (memory and CPU) and data storage. A custom Linux, named Fuel (OPNFV manager), was used to manage the installation of the VIM and the NFVI components on physical servers.

The FW-VNFs were implemented as VMs and the network environment was configured to use VLANs. Three network interfaces were configured in an FW-VNF in different VLANs: One to allow communications with other networks of the laboratory in which the OPNFV was installed, the second for allowing access to the Internet, and the third to allow communications with the VMs in the OPNFV cloud. Iptables was configured to act as a "stateless" packet filter. Moreover, no NAT was configured in an FW-VNF. The network default route of VMs involved in the tests was configured to point to a FW-VNF. Thus, all traffic packets that the VMs generated for any destination were inspected by the FW-VNF. The testbed network was also configured to ensure that all traffic returned through the FW-VNF. Therefore, the FW-VNF was configured to guarantee the security perimeter of the cloud networks.

The proposed firewall was created on the OPNFV version 1.0 platform using five machines. The OPNFV manager was installed on a desktop and four other machines were used to implement the OPNFV cloud. Three were deployed as OpenStack controller nodes and one as a compute node (server), where the VNFs were installed. The OPNFV manager and the first two controller nodes of the OpenStack were installed on machines with Intel (R) Core (TM) i7-4770 CPU @ 3.40 GHz with four cores, 8 threads, 32 GB of RAM, and three 1 Gb/s Ethernet interfaces. The third controller and the OpenStack compute node were installed on a machine with two Intel (R) Xeon (R) E5-2650 CPU @ 2.00 GHz with eight cores, 16 threads, 64 GB of RAM, and three 1 Gb/s Ethernet interfaces. Besides, all OpenStack nodes were installed on Ubuntu 14.04, because it is the only OS available on the OPNFV 1.0. Similarly, the KVM virtualization tool, also the only option supported by OPNFV, was used.

III. Experimental Performance Evaluation

In the tests, we use Iperf tool to send an aggregate traffic from one or more generators to a receiver. To ensure a greater control over the experimental scenario and to isolate external factors, the test machines have no Internet access. A traffic generator corresponds to a VM composed by eight virtual CPUs and 16 GB of RAM in an OpenStack server configuration. Each FW-VNF corresponds to a VM with two virtual CPUs and 4 GB of RAM that is also instantiated in the compute node of the OpenStack. We start the tests with a single FW-VNF and a single instance of traffic generator and we create more instances of them when necessary. The number of generators is equal to the number of instantiated FW-VNFs. The receiver in all test scenarios is a physical server that does not belong to the NFVI. The receiver is an Intel (R) Core (TM) i7-4770 CPU @ 3.40 GHz with four cores, 32 GB of RAM, and three 1 Gb/s Ethernet interfaces.

The aggregate transmission rate[1] is measured by the machine(s) that generate traffic. The reception rate is measured on the receiving machine, where the Iperf server is executed. All results are averages with 95%-confidence intervals.

Performance impact evaluation of the packet size and of the number of rules in one FW-VNF - In the first experiment, we configure the firewall with different numbers of rules and we use different packet sizes in a 1 Gb/s UDP flow to traverse the firewall from one generator to the receiver. Regarding the number of rules, only the last rule of a set of them releases the sent traffic. As a standard firewall orderly processes each rule, FW-VNF performance is evaluated in a worst case scenario for a certain number of rules. Figure 1 shows that the transmission rate varies according to the size of the packet, as expected. The transmission rate is reduced due to restricted packet transmission of the Ethernet card and due to packet headers and trailer. Therefore, as the packet size increases, the transmission rate gets closer to the maximum value of 1 Gb/s. Regarding the number of rules, also as expected, there is no significant performance difference of the transmission rate considering a same packet size.

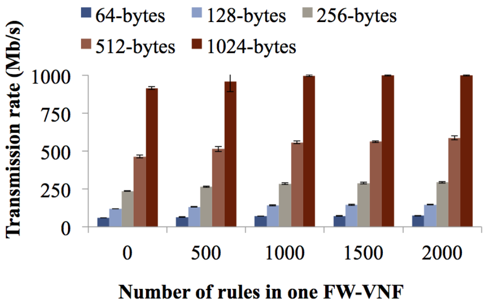

Figure 1. Transmission rate for different UDP packet sizes and different number of rules in one FW-VNF.

Figure 2 shows that, beyond the effect of the limitation of the transmission capacity and of the header overhead, there is a significant variation according to the number of rules. Results show that the firewall performance for more than 500 rules significantly decreases, because the reception rate is much lower than the transmission rate. The performance becomes worse as the number of rules increases. The higher the number of rules, the higher the processing time of each packet in the firewall and, hence, the lower the reception rate.

Performance impact evaluation of the transmission rate and of the number of rules in one or more FW-VNFs - In a second test, we transmit UDP packets at different rates,

[1] The transmission rate is defined as the rate effectively injected into the network.

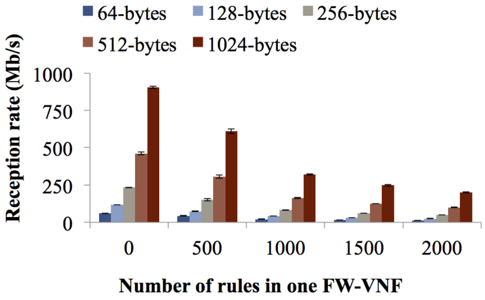

Figure 2. Reception rate for different UDP packet sizes and different number of rules in one FW-VNF.

ranging from 100 to 900 Mb/s. We use 1024-byte packet size, in order to reduce the effect of the limitations of the transmission capacity and of the overhead. The measured transmission is constant and corresponds to the rate offered by Iperf and, then, it was not presented in a figure.

Figure 3 depicts the effect that the number of rules and the transmission rate cause in the reception rate. FW-VNF performs well for every number of rules at a 100 Mb/s transmission rate. For 300 Mb/s rate, however, the performance is good enough up to 1000 rules. For 900 Mb/s, the performance is also sensitive to the number of rules processed by the firewall and worsens when the number of rules increases.

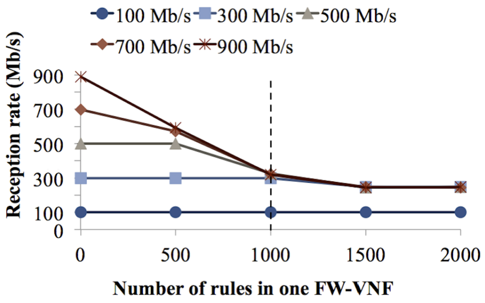

Figure 3. Reception rate for different UDP transmission rates and different number of rules in one FW-VNF.

Figures 2 and 3 show that a firewall function using a unique virtual machine (one FW-VNF) is not sufficient to handle the firewall incoming traffic when the number of firewall access rules is equal to or higher than 500. Thus, we propose to use the elastic property of cloud computing, increasing the number of FW-VNFs to meet the traffic demand.

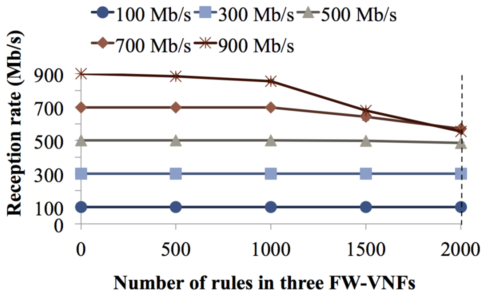

Figure 4. Reception rate for different UDP transmission rates and different number of rules using three FW-VNFs.

In this experiment[2], we instantiate two other FW-VNFs with the same specifications of the previous experiment and we transmit the same UDP packets at different rates, ranging from 100 to 900 Mb/s. However, to each 100 Mb/s transmission rate, 33.3 Mb/s traverse a different FW-VNF instance.

Figure 4 shows that the FW-VNFs could satisfactorily handle a 500 Mb/s traffic when the number of rules is equal or lower than 2000. Observing Figures 3 and 4, we can estimate that each FW-VNF can handle an average rate of about 180 Mb/s when the number of rules is 2000. Therefore, to be able to process a 900 Mb/s traffic, higher than the maximum traffic observed in a ToR of the studied datacenter, it would be necessary to use five FW-VNFs.

IV. Conclusions and Future Work

We implemented a virtual firewall named FW-VNF that can handle a typical number of access policies currently found in the top of rack routers of the studied datacenter.

Performance results showed that one instance of the FW-VNF is not enough to handle 300 Mb/s of incoming traffic or higher, when the number of rules is equal to 2000. We concluded that each FW-VNF was able to handle an average rate of about 180 Mb/s. Therefore, five FW-VNF instances are enough to sufficiently process traffic up to 900 Mb/s in firewalls with 2000 rules. According to the tests, using virtual firewalls to manage top of rack routers access policies in datacenters is viable and presents good performance.

As future work, we aim at extending the analysis to verify the behavior of FW-VNFs in a fully virtualized datacenter.

References

[1] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," *IEEE Transactions on Network and Service Management*, 2016.

[2] I. M. Moraes, D. M. Mattos, L. H. G. Ferraz, M. E. M. Campista, M. G. Rubinstein, L. H. M. Costa, M. D. de Amorim, P. B. Velloso, O. C. M. Duarte, and G. Pujolle, "FITS: A flexible virtual network testbed architecture," *Computer Networks*, vol. 63, pp. 221–237, 2014.

[3] R. S. Couto, M. E. M. Campista, and L. H. M. Costa, "A reliability analysis of datacenter topologies," in *IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 1890–1895.

[4] J. Deng, H. Hu, H. Li, Z. Pan, K.-C. Wang, G.-J. Ahn, J. Bi, and Y. Park, "VNGuard: An NFV/SDN combination framework for provisioning and managing virtual firewalls," in *IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, 2015, pp. 107–114.

[5] J. Bi, S. Zhu, C. Sun, G. Yao, and H. Hu, "Supporting virtualized network functions with stateful data plane abstraction," *IEEE Network*, vol. 30, no. 3, pp. 40–45, May 2016.

[6] M. Chiosi, D. Clarke, P. Willis, A. Reid, J. Feger, M. Bugenhagen, W. Khan, M. Fargano, C. Cui, H. Deng *et al.*, "Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action," in *SDN and OpenFlow World Congress*, 2012, pp. 22–24.

[2] We have also obtained results for two FW-VNFs, but do not present them due to number of pages limitation.