2018 IEEE 32nd International Conference on Advanced Information Networking and Applications

DeMONS: A DDoS Mitigation NFV Solution

Vinícius F. Garcia, Guilherme de F. Gaiardo, Leonardo da C. Marcuzzo,
Raul C. Nunes and Carlos Raniery P. dos Santos
Federal University of Santa Maria - Post-graduate in Computer Science
{vfulber, ggaiardo, lmarcuzzo, ceretta, csantos}@inf.ufsm.br

***Abstract*—Distributed Denial of Service (DDoS) attacks become increasingly sophisticated and massive in traffic volume. These attacks can be mainly classified as IP Spoofing or Real Source IP. In special, Real Source IP attacks are characterized by the use of malware-infected hosts to simulate real network traffic. Those attacks are constantly evolving, new and sophisticated infection methods are always being employed by attackers. To deal with such constant change, the research community is always searching for advanced approaches to mitigate, or even eliminate, those threats. One of these new approaches, is the use of Network Function Virtualization (NFV). This new paradigm supports the creation of more scalable and flexible, thus resilient, network infrastructures. We, therefore, propose a DDoS mitigation system – called DeMONS – that uses NFV concept together both a dynamic allocation and a reputation mechanisms. The results demonstrate that the employed techniques are a feasible solution to reach higher utilization rates.**

I. Introduction

Distributed Denial of Service (DDoS) attacks are large and sophisticated threats that are becoming even more frequent and can lead to significant data losses [1]. Their occurrence is among the top factors of concern for IT companies that rely on the Internet. For those companies, service unavailability due DDoS attacks represent severe financial loss (due broken SLAs or weaken reputation). As an example of how widespread a DDoS can be, an attack performed in 2016 employed more than 100,000 infected machines to overload the Singapore Star Hub Internet Service Provider [2].

Distributed Denial of Service (DDoS) defines attacks that aim to turn a service unavailable by employing a massive number of false requests [3]. DDoS attacks can be classified in two main categories [4]: IP Spoofing and Real Source IP. In the prior, the source IP addresses don't represent real machines, but instead are artificially generated. This kind of attack is the most common form of DDoS and is easily detected by Intrusion Detection Systems (IDS'es). The former, on the other hand, usually employs real infected machines (*i.e.*, a botnet) controlled by a botmaster. Different techniques are also commonly employed by the botmaster to make this kind of attack even more complex to be detected [5].

Due to the potential problems caused by DDoS attacks, the research community is constantly investigating and proposing novel mitigation techniques. Those techniques can be classified in two main groups: capacity and filter. Capacity solutions limit the network traffic using a priority-based approach, while filter methods simple block the malicious traffic. In the occurrence of a large DDoS attack, capacity-based systems are usually overloaded, thus becoming ineffective. In this case, however, due the characteristics of such systems, benign flows are not blocked during an attack even if the detection system errs. On the other side, filter-based solutions rely on a very accurate anomaly detection process, if the system fails, false-positive flows are simple blocked.

Capacity-based DDoS mitigation methods are appropriate when the IDS's accuracy is not determined or complex attacks – such as the Real Source IP – are prevalent. These methods allocate computing resources (*e.g.*, processing, memory and network) only for trusted flows. For unknown or suspicious flows, the system works in a best effort fashion, thus it can become overloaded and eventually not answer some requests. It's important to note that capacity-based methods require significant amount of resources to mitigate DDoS attacks. In this context, technologies that support elastic resource's provisioning, such as virtualization, are specially suitable to be employed.

Network Function Virtualization (NFV) is an emerging paradigm that uses existing virtualization technologies to decouple the network functions from their associated hardware (*i.e.*, middleboxes) [6]. The NFV paradigm envisages a flexible and scalable environment, thus encouraging innovation, reducing time to market and capital/operational costs (CAPEX and OPEX), and enabling commodity hardware (less expensive) to be employed [7].

Due its advantages, several efforts available in the literature [8] [9] [10] [4] are using NFV as an interesting approach to mitigate DDoS attacks. In special, [4] proposed a solution – called VGuard – to mitigate real source IP DDoS attacks with a capacity-based mechanism. VGuard employs tunnels with different priorities (high and low) to separate traffic by reputation. The high priority tunnel is never overloaded, thus guaranteeing that packets reach their destination. The second tunnel, on the other hand, receives flows with low reputation and operates in a best effort mode (*i.e.*, when overloaded packets are discarded).

Even VGuard being a feasible solution to mitigate DDoS attacks and ensuring a certain Quality of Service (QoS), it is not efficient avoiding DDoS attacks. For example, once a specific flow is allocated to a certain tunnel, the system doesn't reassess it. This approach is not adequate in all the scenarios, such when some DDoS flows reach the high priority tunnel or higher priority flows are allocated in low priority tunnel.

In this context, this work presents DeMONS, a new hybrid solution (based on both capacity and filter approaches) to mitigate DDoS attacks by using the NFV paradigm. The

1550-445X/18/$31.00 ©2018 IEEE
DOI 10.1109/AINA.2018.00115

proposed architecture was inspired by VGuard architecture, with an IDS providing priorities, a firewall used to block the recognized malicious traffic, and an allocation system assigning flows in tunnels with different priorities. DeMONS, however, introduces a new module – the Classifier – that constantly analyzes the network flows to decide which tunnel they should be allocated. Further, the low priority tunnel employs a reputation mechanism (implemented as a VNF) responsible for limiting the network traffic according to the priorities defined by the IDS. With such approach, packets with higher priorities has lower drop rates when the low priority tunnel is overloaded.

This paper is structured as follows: Section 2 presents the main concepts related to the proposed solution. In Section 3 NFV-based proposals for DDoS mitigation are presented and discussed. Section 4 examines VGuard architecture and concepts. Next, DeMONS is introduced in Section 5, with a comparison between VGuard and DeMONS discussed in Section 6. Finally, Section 7 concludes the paper.

II. Background

This section presents the main concepts related to NFV and Security Service Chains (SSC).

A. Network Function Virtualization

Traditional network infrastructures rely on dedicated, and specialized, hardware – called Middleboxes – to execute important functions, such as routing and proxying. Such choice is mainly due to strict performance requirements of network maintainers. However, the use of such middleboxes introduce limitations in the daily operations of computer network. For example, high capital and operational costs (CAPEX and OPEX), lack of flexibility, and complexity to manage and scale the infrastructure [11].

To overcome those issues, a new paradigm called Network Functions Virtualization (NFV) was recently introduced [6]. The NFV paradigm aims to decouple network functions from its associated hardware by using existing virtualization technologies, such as Virtual Machines (VMs) and Linux Containers. There are several advantages in using NFV, such as fast prototyping, CAPEX/OPEX reduction, and dynamic support to changing user requirements.

However, the introduction of a virtualization layer to implement the network functions leads to performance degradation (*e.g.*, higher delay, lower throughput). Fortunately, disparate NFV enablers (*e.g.*, OpenNetVM [12] and Click-On-OSv [13]) are employing modern technologies (*e.g.*, DPDK [14], NetMap [15]) to prevent such problem. All those efforts enable NFV to be used in production environments with little to no negative effects.

B. Security Service Chain

The European Telecommunications Standards Institute (ETSI) introduced the concept of Service Function Chain (SFC), which can be defined as a group of virtualized functions that process network flows [16]. By adopting a structural point of view, a SFC can be represented as a forwarding graph [17], with nodes representing the virtualized functions, linked by edges representing the logical connections.

Ongoing efforts introduced the concept of Security Service Chain (SSC) [18]. A SSC is essentially a SFC with all the components (*e.g.*, Intrusion Detection Systems, Intrusion Prevention Systems, and firewalls) devoted to perform security related tasks, such as detecting network anomalies and preventing malicious activities. Finally, the NFV paradigm usage to provide system security makes the security topology administration easy and efficient.

III. Related Works

Recent efforts of the research community are employing NFV as an approach to improve the efficacy of security architectures. Bellow, the most significant of those works are presented and discussed.

In [8] the authors demonstrated the advantages and problems in using traditional DDoS mitigation techniques in virtualized environments. The authors, for example, highlight as positive aspects of NFV its flexibility, elasticity, and reduced CAPEX/OPEX. However, despite those characteristics, NFV still presents several challenges, such as the lack of support for fast and efficient deployment of defense topologies.

In [8] the authors proposed a DDoS mitigation architecture with support to analyze both network and application data. The central component of the architecture – *i.e.*, the traffic screener – was designed to analyze all the data traffic in the search for potential anomalies. This component classifies the traffic and routes it according to its type (*i.e.*, attack or service). Service traffic is forwarded to the final system, while attack traffic is forwarded to be processed by a VNF. One shortcoming of the proposed approach, however, is that how to efficiently execute an efficiently resource provisioning process according to traffic screener demand.

VFence [9] is a system developed to mitigate SYN flood attacks. The system is based on scalable VNFs responsible for filtering malicious traffic: the dispatcher and the agents. The dispatcher is responsible for load balancing the traffic between the available agents. The agent, in turn, is responsible for verifying if a flow is malicious or benign by coordinating the three-way handshake process. Authenticated flows are stored in a white list, while any other traffic is blocked.

CoFence [10] was created to support the collaboration of multiple administrative domains in order to reduce SYN flood attacks. Each domain provides resources (*e.g.*, a company or a government agency) that could be used in the DDoS detection process. It's important to note that, despite the advantages in using a collaborative-based approach, this method leads to privacy issues since the traffic can be analyzed by a different domain.

IV. VGuard Solution

VGuard [4] is a Real Source IP DDoS mitigation solution based on NFV technology. The architecture consists of: a classification module (based on an IDS) that assigns priorities

for each incoming flow and marks its packets; a firewall to block recognized malicious traffic; and an allocation module that assigns traffic with different priorities to distinct tunnels. The high priority tunnel is designed to support every flow allocated to it, while the low priority tunnel works in a best effort fashion.

The classification module (*i.e.*, the IDS) provides a priority table accessed by the other modules of the proposed solution. If an specific flow is identified as benign, it will have priority to access the desired service. If a flow is considered malicious (*i.e.* zero priority) the firewall simple discards the network packets before reaching the target system. Finally, if a flow has an unknown state, it will receive a priority between zero and one.

The evaluated scenario consists of multiple users accessing an application server (*e.g.*, ProFTPD, Apache Server) while a DDoS attack is initiated. During the attack, new users are also included to the system in order to simulate the behavior of real network service scenarios.

It's important to notice that DDoS attacks have specific characteristics that distinct them from benign traffic. For example, the amount of malicious traffic is higher, suddenly initiates, and are not usual to the security system [19]. These characteristics can be used to determine the priorities for each network flow. However, the simple flow analysis may not be sufficient to determine if an attack is occurring. In this case, additional systems (*e.g.*, sandbox and anti-virus on-line tools) may be used to update the flow priority.

A. Performance

To analyze the VGuard performance it is first necessary to determine the drop rate (d) for each tunnel. The drop rate lies in the interval [0;1], with 0 representing the absence of packet dropping and 1 when all the traffic is dropped. The drop rate is calculated using the tunnel capacity (C) and the amount of network traffic crossing the tunnel (r), as shown in Equation (1).

$$d = 1 - min(1, C/r) \tag{1}$$

Service satisfaction is a method to quantify and capture the quality of service. The satisfaction (s) assumes a parabola form $h(x) = x^2$. Considering that all traffic is properly processed by the server, the calculated satisfaction is 1, in other case the satisfaction will drop quadratically with the tunnel drop rate (d). The tunnel satisfaction is presented in Equation (2).

$$s = (1 - d)^2 \tag{2}$$

VGuard efficacy is evaluated by the aggregate satisfaction (S_V) of the low and high priority tunnels (S_L and S_H respectively), weighed by the flow's priorities (p). The S_V corresponds to a dimensionless number from zero up to sum of the tunnels capacity. Such satisfaction can be expressed as follows in Equation (3).

$$S_V = S_L + S_H = \sum_{i=0}^{n} r_i p_i s_L + \sum_{j=0}^{m} r_j p_j s_L \tag{3}$$

B. VGuard Flows Allocation

The VGuard solution provides two flow allocation types: static and dynamic. The dynamic flow allocation reaches better results than the static. This allocation method uses the priority information and the tunnels state to decide where an incoming flow should be allocated. The dynamic allocation, demonstrated in Algorithm 1, presents the following characteristics:

- New flows are allocated on the least used tunnel if all the tunnels are underloaded;
- High priority tunnel goes to selective mode when it is near the maximum load, an algorithm starts to decide where new flows are allocated;
- Summary flows allocation on low priority tunnel occur when a small signal of overload or full utilization is noted in the high priority tunnel.

Algorithm 1: VGuard's Dynamic Flows Allocation

```
t_H, t_L : high and low priotity tunnels
τ_max : maximum utilization of high priority tunnel
τ_norm : threshold for slectivity mode entrance
U_H, U_L : high and low tunnels utilization
begin
    //Initializing
    t_H = t_L = 0
    Event e trigered when a new flows f arrives
    if U_L < U_H then
        t_L ← t_L ∪ f
    end
    else
        if U_L < τ_norm then
            t_H ← t_H ∪ f
        end
        else
            if U_H > τ_max then
                t_L ← t_L ∪ f
            end
            else
                //Selective mode working
                if Prioridade(f) > averagePriority(t_H) then
                    t_H ← t_H ∪ f
                end
                else
                    t_L ← t_L ∪ f
                end
            end
        end
    end
end
```

V. DeMONS Solution

DeMONS is a hybrid solution (capacity and filter based) to mitigate DDoS attacks that uses NFV paradigm to guarantee an elastic resource management. The DeMONS architecture, depicted in Figure 1, consists in five main modules (implemented as VNFs): Priority Classifier, Firewall, Flow Allocation, Traffic Policing, and Manager.

First, the incoming traffic crosses the Priority Classifier, which analyzes the flow and defines a priority value in the range [0;1], with 0 representing a recognized malicious flow, thus being blocked by the next module (*i.e.*, the Firewall). Other flows are allowed by the Firewall to be processed by the Flow Allocation module.

The Flow Allocation module, in turn, forwards the flows according to its priorities to distinct tunnels (*i.e.*, a high and a low priority tunnel). The prior handles high priority flows when the system is overloaded. It's important to notice that

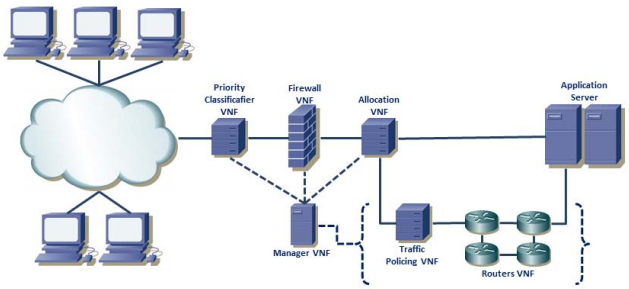

Fig. 1. DeMONS Architecture

this tunnel is designed to never operate above its capacity. The former receives flows with low priorities and can operate overloaded. When this tunnel is overloaded, the Traffic Policing allocates the bandwidth according to the flow priority.

The Manager is the responsible for provisioning the DeMONS system. It monitors the infrastructure load and decides when the VNFs must scale in, out, up or down. Also, in an underutilization scenario, it can turn off all the low priority segment, thus avoiding resource and energy wasting.

The Flow Allocation methods used in DeMONS is presented in Algorithm 2. When both tunnels are underloaded, the Flow Allocation balances the traffic between the tunnels for a better resource usage (line 10 to 21). When the high priority tunnel reaches a predefined capacity utilization, it goes to selective mode (line 22) and a flow balancing method is executed to reallocate the higher priority flows to the high priority tunnel. The flow balancing consists in execute a conditional allocation for every flow crossing the low priority tunnel with priority greater than the minor flow priority crossing the high priority tunnel.

Once the flows balance ensures that flows of higher priorities are crossing the high priority tunnel, next allocations decisions for new flows takes in consideration only the smallest flow priority in there (Line 31). Conditional allocation (Lines 24 to 36) can be performed to add new flows with higher priorities in the high priority tunnel by offloading previous allocated flows with lower priorities, the Algorithm 3 presents the routine.

To benefit flows with higher priorities allocated in the low priority tunnel, the DeMONS solution includes a Traffic Policing VNF. This VNF determines the maximum traffic that will be accepted for each flow considering their priority. This reputation system aims to reduce the overload in a controlled way. The routine check the flows in crescent priority order, for flows with same priority the decreasing traffic amount is considered as secondary order. The applied policing is represented by Equation (4).

The traffic policing uses a minimum packet dropping rate of 10% independently of the analyzed flow priority. The solution aims to completely compute the largest possible number of benign incoming flows, so even flows with 1 of priority can be policed avoiding that only a single flow uses a large parcel

Algorithm 2: DeMONS Dynamic Flows Allocation

```
1  t_H, t_L : high and low priotity tunnels
2  τ_Hmax, τ_Lmax :
     maximum utilization of high and low respectively priority tunnel
3  τ_Hnorm : threshold para entrada em modo seletivo
4  U_H, U_L : high and low tunnels utilization
5  LU_H : last utilization of high priority tunnel
6  begin
7      //Initializing
8      t_H = t_L = 0
9      Event e trigered when a new flows f arrives
10     if U_L < U_H then
11         t_L ← t_L ∪ f
12     end
13     else
14         if U_L < τ_Hnorm then
15             if U_H + traffic(f) <= τ_Hmax then
16                 t_H ← t_H ∪ f
17             end
18             else
19                 t_L ← t_L ∪ f
20             end
21         end
22         else
23             if U_H > τ_Hmax then
24                 condicionalAllocation(f)
25             end
26             else
27                 //Selective mode working
28                 if LU_H < τ_Hnorm then
29                     balanceFlows()
30                 end
31                 if priority(f) > minorPriority(t_H) then
32                     if U_H + traffic(f) <= τ_Hmax then
33                         t_H ← t_H ∪ f
34                     end
35                     else
36                         condicionalAllocation(f)
37                     end
38                 end
39                 else
40                     t_L ← t_L ∪ f
41                 end
42             end
43         end
44     end
45 end
```

Algorithm 3: Routine condicionalAllocation(f)

```
1  begin
2      g_D : flows set to be allocated in t_L
3      U_D : traffic generated by g_D flows
4      g_D = 0
5      while priority(f) > minorPriority(t_H) do
6          g_D ← g_D ∪ minorPriority(t_H)
7          t_H ← t_H − minorPriority(t_H)
8          if U_D >= traffic(f) then
9              t_L ← t_L ∪ g_D
10             t_H ← t_H ∪ f
11             return
12         end
13     end
14     t_H ← t_H ∪ g_D
15 end
```

of tunnel capacity. The policing stops when the tunnel limit usage is reached or when the last flow crossing the low priority tunnel is verified, indicating that, even with the controlled packet dropping, some random packet discarding may still occur. The Algorithm 4 shows how the reputation system was implemented.

$$((1-p)+(p*0.1))*traffic(flow) \tag{4}$$

VI. Validation and Tests

We performed a series of simulations in order to validate our proposal [1].The methodology employed in the simulations was based on the employed in [4], with both low and high priority tunnels set at 50 Mbps each, and the high priority tunnel goes into selective mode when the traffic reaches 97% of the capacity. Benign flows are generated at a rate of 100 Kbps, with a period of 10 seconds limited at 10 Kbps. Malicious

[1]GitHub: https://github.com/ViniGarcia/DeMONS-PoC

Algorithm 4: TRAFFIC POLICING USING A REPUTATION SYSTEM

```
1  t_Lexc : over traffic in low priority tunnel
2  f_drop : drop rate for a flow
3  l_drop : set of droping rates by flow
4  begin
5      l_drop = 0
6      if U_L > τ_Lmax then
7          t_Lexc = U_L − τ_Lmax
8          sort(t_L)
9          for fluxo em t_L do
10             f_drop = ((1 − prioridade(fluxo)) + (prioridade(fluxo) *
               0.1)) * traffic(fluxo)
11             if f_drop < t_Lexc then
12                 l_drop = l_drop ∪ f_drop
13                 t_Lexc = t_Lexc − f_drop
14             end
15             else
16                 l_drop = l_drop ∪ t_Ldrop
17                 return
18             end
19         end
20     end
21 end
```

flows also starts with 100 Kbps, however the transmission rate does not degrades during the attack.

The amount of generated traffic is represented by Equation (5). Each simulation lasts 30 seconds, with tunnel and flows information (*e.g.*, traffic amount, priorities) being updated each second. Priorities are randomly attributed for each flow, benign flows present priorities between 0.4 and 1, while malicious flows have priorities between 0.1 and 0.4. Priority 0.4 is an intermediate zone, where some flows are malicious and some are benign. Zero priority flows (*i.e.*, malicious) were not generated since they are blocked by the Firewall.

In order to compare VGuard and DeMONS, the following three scenarios were defined:

- Benign flows and normal traffic;
- Benign flows and traffic overload;
- DDoS attack.

We also analyzed the performance of the traffic policing module. It was configured with multiple reputation systems and applied in both benign traffic overload and DDoS attack scenarios. Finally, we identified the consequences of using a restrictive system (*i.e.*, resulting in high controlled drop rates) or a softer one.

A. Benign Flows and Normal Traffic

In this simulation, all the generated flows are benign and present a peak of 99.1 Mbps.

$$f(x) = \frac{66x}{5} - \frac{11x^2}{25} \tag{5}$$

Since the traffic amount does not exceed the capacity of the tunnels at any moment, all the network traffic reaches the server in VGuard and DeMONS. The satisfaction of both low and high priority tunnels are presented respectively in Figures 2 and 3.

Figure 3 depicts the moment when the high priority tunnel goes into selective mode (from $t = 13s$ to $t = 16s$) and the low priority tunnel is overloaded in both solutions. In this case, the low priority tunnel of VGuard and DeMONS operates with a satisfaction level of, respectively, 1.1% and 0.6% below the maximum reachable.

When the high priority tunnel goes to selective mode, the flow balance realized in DeMONS ($t = 13s$) results in an abrupt changes in the satisfaction level measurements. A sudden fall occurs in the low priority tunnel (once it receives flows with lower priorities) and a sudden increase is noted in the high priority tunnel (where flows with higher priorities are passing).

B. Benign Flows and Traffic Overload

In this simulation, all the generated flows were benign and the peak traffic was 506 Mbps. The function that defines the traffic distribution in time is presented in 6.

$$f(x) = \frac{135x}{2} - \frac{9x^2}{4} \tag{6}$$

With traffic five times greater than the aggregated tunnels capacity, a significant overload is imposed to the solution. Since the high priority tunnel do not accepts more traffic than its capacity, much more traffic is forced to pass through the low priority tunnel. The satisfaction of the low priority tunnel – represented in Figure 4 – decreases as the number of benign flows also decreases. Figure 5 shows a zoomed version of traffic peak moments in the low priority tunnel, while the high priority tunnel's satisfaction is demonstrated in Figure 6.

The high priority tunnel goes into selective mode in $t = 2s$, and continues until $t = 27s$ as depicted in Figure 6. The low priority tunnel's satisfaction has an abrupt and premature drop and keep this level until the simulation finishes. The use of a reputation system in DeMONS leads to a satisfaction increase of 21.3% in comparison to VGuard.

Due to the rapid balancing of DeMONS high priority tunnel, the satisfaction increases as soon as the simulation begins. The satisfaction on VGuard's high priority tunnel takes longer to increase because it depends on lower priority flows ending and higher priority flows entering. As the time pass and priority maturation happens, VGuard achieves the highest satisfaction rates, as seen in moments 16 to 22 in Figure 6.

C. DDoS Attack

The DDoS attack scenario was simulated generating benign flows (with a maximum rate of 99.1 Mbps and represented in 5) and an continuous malicious traffic of 500 Mbps starting at $t = 10s$ and ending at $t = 20s$. Figure 7 depicts the benign and malicious flows, besides the traffic distribution that are submitted to the system.

Due to the malicious traffic overload and the low priority presented by the malicious flows, a massive allocation in the low priority tunnel occurs. It generates an abrupt and continuous low satisfaction rate. The high priority tunnel keeps delivering its flows normally, keeping the traffic under the maximum utilization. The low and high priority tunnel satisfactions are presented respectively in Figures 8 and 10, while 9 depicts a zoom of low priority tunnel satisfaction when the DDoS attack is occurring.

The high priority tunnel goes into selective mode at the beginning of the DDoS attack and keeps in this state until the

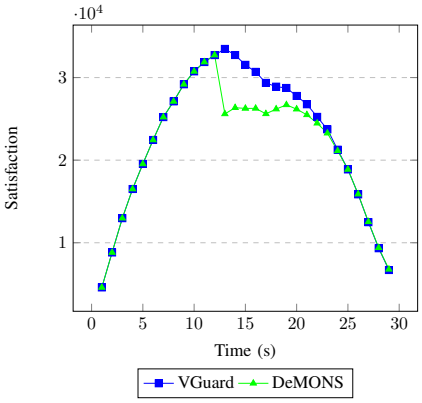

Fig. 2. Low Priority Tunnel (Equation (5))

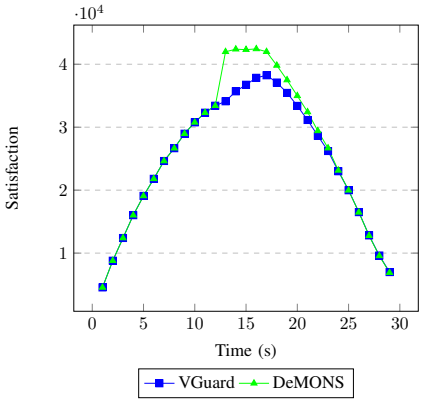

Fig. 3. High Priority Tunnel (Equation (5))

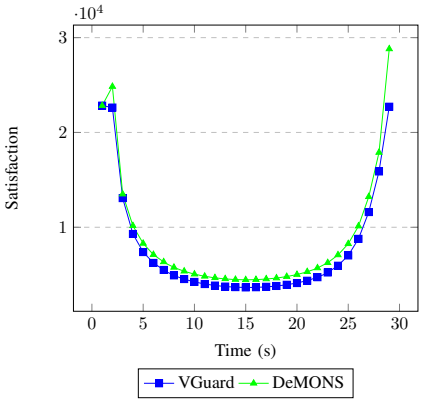

Fig. 4. Low Priority Tunnel (Equation (6))

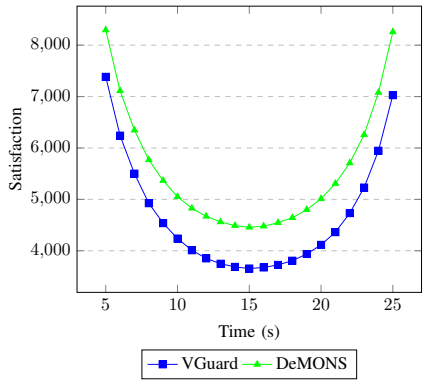

Fig. 5. Low Priority Tunnel Zoom (Equation (6))

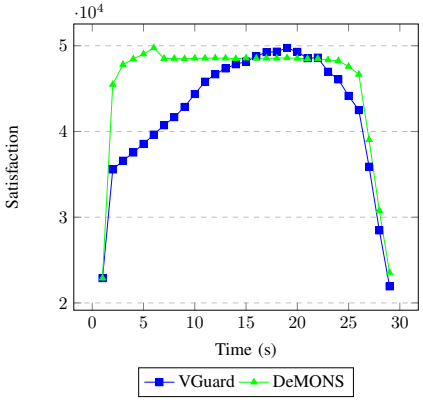

Fig. 6. High Priority Tunnel (Equation (6))

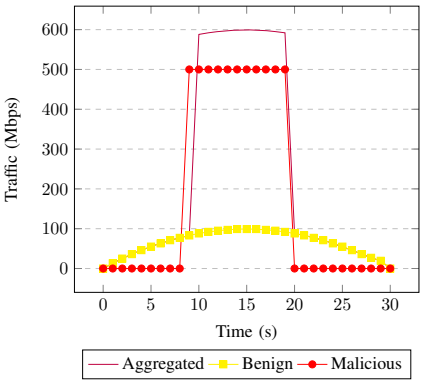

Fig. 7. DDoS Scenario Traffic

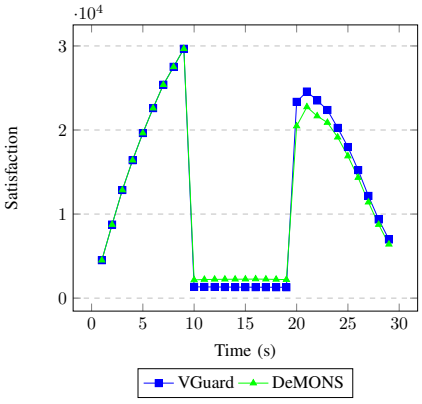

Fig. 8. Low Priority Tunnel (DDoS)

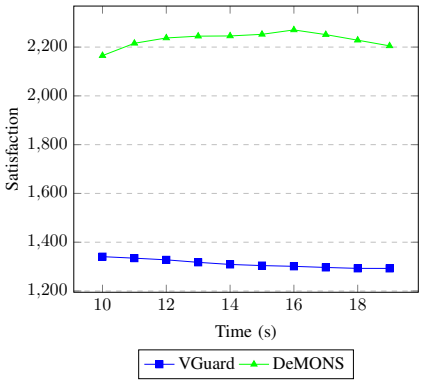

Fig. 9. Low Priority Tunnel Zoom (DDoS)

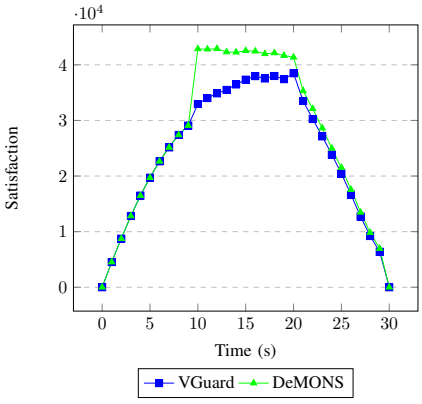

Fig. 10. High Priority Tunnel (DDoS)

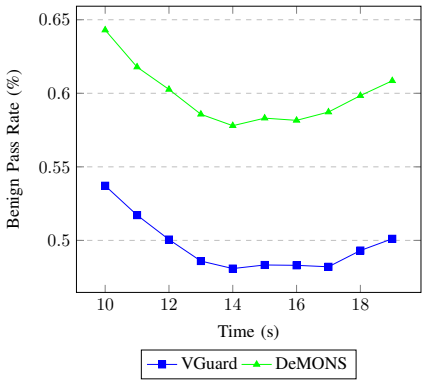

Fig. 11. Benign Flows Pass Rate (DDoS)

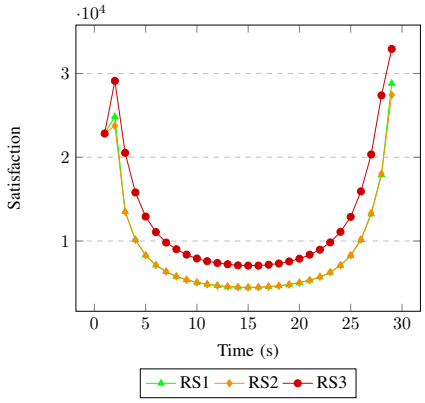

Fig. 12. Reputation Systems Satisfaction (Eq. 6)

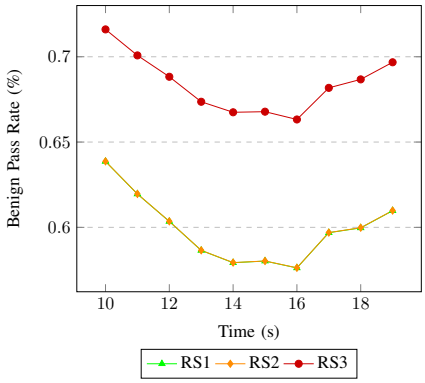

Fig. 13. Reputation System Benign Pass Rate (DDoS)

malicious traffic ends. The malicious flows are allocated in the low priority tunnel due their low priorities. Therefore, the low priority tunnel presents low satisfaction levels and since it is overloaded high drop rates are verified. However, the reputation system employed in DeMONS is capable of increasing the low priority tunnel satisfaction level by dropping most of the malicious traffic.

The load balancing executed by DeMONS (when the high priority tunnel goes into selective mode), besides hosting the highest priority flows, also removes all malicious flows crossing it before entering into selective mode. These malicious flows generated 6.9 Mbps of traffic in the high priority tunnel. Since these flows do not decrease and ends only when the attack stops, the presence of them in the high priority tunnel represents a resources wasting.

Figure 11 presents the percentage of benign flows that reached the server for each tested solution. VGuard presented the worst results due to the malicious flows crossing the high priority tunnel and a higher number of benign flows in the low priority tunnel, where a high drop rate is noted.

The DeMONS solution achieved better results for all the evaluated scenarios. The reputation system in the low priority tunnel was responsible for dropping most of the malicious traffic due to the corresponding low priorities. The efficient utilization of the high priority tunnel due to balancing of network flows and the use of a reputation system in the low priority tunnel results in higher rates of benign traffic.

D. Traffic Policing Analysis

The DeMONs traffic policing module can use different reputation systems, leading to a greater or minor traffic amount crossing the low priority tunnel. Extremely restrictive reputation systems may lead to discard of all low priority flows, thus avoiding traffic of new unknown flows (but benign ones) to reach the server. On the other hand, a too soft system enables most of the traffic to be delivered, including very low priority flows (and possibly malicious). This leads to low satisfaction levels and benign flows pass rate (when a DDoS occurs) in the low priority tunnel.

In addition to the reputation system already integrated to DeMONS (RS1 - Equation (4)), two other possible implementations were considered to be analyzed in the scenarios of benign traffic overload and DDoS attack (*i.e.*, when the low priority tunnel is truly overloaded): a less restrictive one, with low drop rates for the low priority flows and never totally discarding any network flow (RS2 - Equation (7)); and a more restrictive one, with drop rates corresponding to the amount of excess traffic in the low priority tunnel and capable to discard all the traffic of a specific flow (RS3 - Equation (8)).

$$(1 - priority(f)) * traffic(f) \tag{7}$$

$$((1 - priority(f)) + (1 - priority(f) + priority(f) * 0.1) * excess(t_L)) * traffic(f) \tag{8}$$

In the scenario of benign traffic overload (Equation (6)), the low priority tunnel satisfaction, depicted in Figure 12, is almost the same (moments $t = 3$ and $t = 28$) when the reputation systems 1 and 2 are employed. This represents a maximum variation of 0.8% (moments $t = 2$ and $t = 29$) and a satisfaction variation of, respectively, 4.3% and 4.8% in favor of RS1 is noted due the greater drop of low priority flows in a

little overloaded moments (*i.e.*, most of the flows with higher priorities in the low priority tunnel is served by the server). The RS3 reached better satisfaction levels in all moments of the evaluation, however it drops all the traffic generated by very low priority flows, that never reaches the server.

In the scenario of DDoS attack (Figure 7), the benign pass rate between the reputation systems, presented in Figure 13, shows that, in the same way that satisfaction in the stress test of the low priority tunnel, the difference between RS1 and RS2 is minimal, with slightly higher rates (between 0.005% and 0.015% greater) for RS2. The RS3 reaches a higher pass rate of benign traffic since it is capable to drop all low priority malicious traffic, being between 10.80% and 13.21% higher when compared to RS1 and RS2 results.

VII. Conclusion

DDoS attacks are sophisticated threats that generate a massive amount of network traffic. The rising of new technologies (*e.g.*, IoT, Fog Computing) lead to the increasing of connected equipments to the Internet. These equipments, however, can be infected and used to perform significant attacks. At the same time, a new paradigm – called NFV – presents interesting features (*e.g.*, flexibility and scalability) that support the development of new DDoS mitigation solutions.

This paper proposes a new hybrid solution to mitigate DDoS called DeMONS. The solution is composed of five main modules (implemented as VNFs): a priority classifier, a firewall, an allocation, a traffic policing, and a manager. In DeMONS, network flows are analyzed by the priority classifier and marked with a reputation between 0 and 1. Zero priority flows are blocked in the firewall, while the remaining flows are assigned to distinct tunnels (high and low priority) by the allocation module. When the low priority tunnel is overloaded, the Traffic Policing module executes an algorithm to limit the traffic of each flow based on their priorities. The manager module is responsible for provisioning and controlling all modules life cycle.

The obtained results demonstrates the feasibility of DeMONS to efficiently mitigate DDoS attacks. The use of a reputation system, with the policy shaping module, and an effective allocation method, enabled us to achieve better results compared to VGuard. It's important to notice that, further the improvements obtained in DDoS mitigation, DeMONS also provides a better Quality of Service from end-user's perspective.

As future work, we will analyze the impact of disabling the selective mode of the high priority tunnel. Further, we also aim to evaluate different scenarios with disparate traffic and priority distributions for a full understanding of the solution's capabilities.

References

[1] L. Garber, "Denial-of-service attacks rip the internet," *Computer*, vol. 33, no. 4, pp. 12–17, 2000.

[2] StarHub, "Starhub confirms cause of home broadband incidents on 22 october and 24 october 2016," http://www.starhub.com/about-us/newsroom/2016/october/media-statement--2---starhub-confirms-cause-of-home-broadband-in.html, accessed: 2017-05-26.

[3] C. Douligeris and A. Mitrokotsa, "Ddos attacks and defense mechanisms: classification and state-of-the-art," *Computer Networks*, vol. 44, no. 5, pp. 643–666, 2004.

[4] C. J. Fung and B. McCormick, "Vguard: A distributed denial of service attack mitigation method using network function virtualization," in *2015 11th International Conference on Network and Service Management (CNSM)*, Nov 2015, pp. 64–70.

[5] E. Alomari, S. Manickam, B. Gupta, S. Karuppayah, and R. Alfaris, "Botnet-based distributed denial of service (ddos) attacks on web servers: classification and art," *arXiv preprint arXiv:1208.0403*, 2012.

[6] N. ETSI, "Network functions virtualization, white paper," 2012.

[7] W. John, K. Pentikousis, G. Agapiou, E. Jacob, M. Kind, A. Manzalini, F. Risso, D. Staessens, R. Steinert, and C. Meirosu, "Research directions in network service chaining," in *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, Nov 2013, pp. 1–7.

[8] T. Alharbi, A. Aljuhani, and H. Liu, "Holistic ddos mitigation using nfv," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan 2017, pp. 1–4.

[9] A. H. M. Jakaria, W. Yang, B. Rashidi, C. Fung, and M. A. Rahman, "Vfence: A defense against distributed denial of service attacks using network function virtualization," in *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, June 2016, pp. 431–436.

[10] B. Rashidi, C. Fung, and E. Bertino, "A collaborative ddos defence framework using network function virtualization," *IEEE Transactions on Information Forensics and Security*, vol. PP, no. 99, pp. 1–1, 2017.

[11] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar, "Making middleboxes someone else's problem: Network processing as a cloud service," *SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 4, pp. 13–24, Aug. 2012. [Online]. Available: http://doi.acm.org/10.1145/2377677.2377680

[12] W. Zhang, G. Liu, W. Zhang, N. Shah, P. Lopreiato, G. Todeschi, K. Ramakrishnan, and T. Wood, "Opennetvm: A platform for high performance network service chains," in *Proceedings of the 2016 Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, ser. HotMIddlebox '16. New York, NY, USA: ACM, 2016, pp. 26–31. [Online]. Available: http://doi.acm.org/2940147.2940155

[13] L. d. C. Marcuzzo, V. F. Garcia, V. Cunha, D. Corujo, J. P. Barraca, R. L. Aguiar, A. E. Schaeffer-Filho, L. Z. Granville, and C. R. P. d. Santos, "Click-on-osv: A platform for running click-based middleboxes," in *2017 IFIP/IEEE International Symposium on Integrated Network Management*. IEEE, 2017.

[14] Intel, "Intel data plane development kit," http://http://dpdk.org, accessed: 2017-05-27.

[15] L. Rizzo, "Netmap: a novel framework for fast packet i/o," in *21st USENIX Security Symposium (USENIX Security 12)*, 2012, pp. 101–112.

[16] P. Quinn and T. Nadeau, "Problem statement for service function chaining," 2015.

[17] G. ETSI, "Network functions virtualisation (nfv); use cases," *V1*, vol. 1, pp. 2013–10, 2013.

[18] W. Lee, Y.-H. Choi, and N. Kim, "Study on virtual service chain for secure software-defined networking," *Advanced Science and Technology Letters*, vol. 29, no. 13, pp. 177–180, 2013.

[19] X. Yang, D. Wetherall, and T. Anderson, "Tva: a dos-limiting network architecture," *IEEE/ACM Transactions on Networking (ToN)*, vol. 16, no. 6, pp. 1267–1280, 2008.