2016 IEEE 40th Annual Computer Software and Applications Conference

VFence: A Defense against Distributed Denial of Service Attacks using Network Function Virtualization

A H M Jakaria†, Wei Yang*, Bahman Rashidi*, Carol Fung*, and M. Ashiqur Rahman†
†Department of Computer Science, Tennessee Tech University, Cookeville, USA
Emails: ajakaria42@students.tntech.edu, marahman@tntech.edu
*Department of Computer Science, Virginia Commonwealth University, Richmond, USA
Emails: yangw3@mymail.vcu.edu, rashidib@vcu.edu, cfung@vcu.edu

***Abstract*—With the exponential growth of the Internet use, cyber-threats are emerging rapidly. Distributed Denial of Service (DDoS) attack is one of the most common but damaging kinds of cyberattacks. A DDoS attack to a server typically prevents clients from receiving service by making the server overwhelmed with many invalid service requests. It is always a challenging problem to protect a system from DDoS attacks as it is not trivial to distinguish between an attack packet and a legitimate one. In this work, we have proposed VFence – a defense mechanism against DDoS attack that leverages the capability of the Network Function Virtualization (NFV) architecture. NFV is the technology of virtualizing network functions in virtual machines on commodity servers and it allows a flexible and dynamic implementation of the network functions. Our proposed mechanism uses network agents to intercept packets when the system is potentially under attack, to verify their authenticity, and to keep the server safe by dropping illegitimate packets. Since the attack intensity often varies, our NFV-based defense framework deploys agents dynamically to balance the attack load. Our simulation results demonstrate that the mechanism can successfully defeat the DDoS attacks by having all legitimate requests served, and the increase in the server's response time is insignificant compared to that of a successful DDoS attack.**

***Index Terms*—Distributed Denial of Service; Network Function Virtualization; Dynamic Defense Mechanism.**

I. Introduction

Distributed Denial of Service (DDoS) attacks can cause severe damage to ISPs and online services, especially for those small or medium-sized organizations who lack the resources to withstand a high volume of DDoS traffic. Some recent incidents show that DDoS attacks are becoming stronger and more frequent. For example, the Spamhaus attack in 2013 [1] has generated 300 Gbps attack traffic. This number has been increased to 600 Gbps in January 2016 [2].

There are two major types of DDoS techniques in the attack traffic: IP spoofing and real source IP attack. The DDoS attack, based on real source IP addresses, commonly utilizes compromised nodes in the Internet to launch the attack. IP spoofing DDoS is the type of attack where the source addresses are not the real IP address of the attacker. An example of this type of attack is SYN Floods [3]. Reports show that SYN Floods take the vast majority of the attack volume in recent major DDoS attacks [4]. Existing solutions on SYN Floods, including firewall and cloud service [5], either bring high cost by purchasing dedicated hardware or trigger privacy concerns by directing traffic to a third party. In this paper, we introduce a new approach to defend against SYN Floods leveraging the Network Function Virtualization (NFV) architecture.

NFV is an emerging technology where network functions are implemented and provided in software, which runs on the commodity hardware [6]. The network functions are implemented as software and deployed as virtual machines. The virtual machines run on general purpose commodity hardware systems so that NFV not only provides the benefit of elasticity, but also reduces the cost by running on commodity platforms like x86- or ARM-based servers instead of specialized hardware, resulting in a much easier deployment and lower cost. At the same time, NFV also introduces an opportunity for DDoS detection and mitigation. The traditional methods of DDoS detection are limited by the computation capacity and flexibility of involved network functions, such as firewall and routers. Upgrading or adding new hardware introduces high cost. The use of NFV opens a new opportunity for enterprises to find low cost solutions for defending DDoS attacks.

In this work, we introduce a novel solution that involves the use of NFV to mitigate DDoS attacks. We utilize the flexibility of network functions to create DDoS traffic filtering agents to handle external traffic to the targeted online service. External traffic is directed to one of the agents for filtering. The clean traffic is forwarded to the destination. A dynamic load-balancing method is used to select an agent to handle an incoming flow. Our experimental results demonstrate that our proposed solution can effectively reduce the attack flow to the target server, and therefore our solution mitigates DDoS attacks. The contributions of this paper include:

1) To the best of our knowledge, this is the first proposed solution using NFV to mitigate SYN Flood attacks.
2) We propose dynamic NFV DDoS filtering architecture and protocols.
3) We evaluate our proposed solutions using simulation and demonstrate the effectiveness of our solutions.

The rest of this paper is organized as follows: Section II

0730-3157/16 $31.00 © 2016 IEEE
DOI 10.1109/COMPSAC.2016.219

discusses the attack model, corresponding problems, research motivation and challenges, and contributions. We present the proposed framework in Section III. The evaluation results of our model are presented in Section IV. Then, we conclude the paper in Section V.

II. Background and Related Work

This section briefly overviews the NFV technology, existing DDoS attack techniques, and DDoS defense strategies.

A. Network Function Virtualization

NFV is gaining growing attention from both academia and industry due to its potential for cost reduction and operational efficiency. The main idea of NFV is to replace dedicated network appliances, such as hardware-based routers and firewalls, with software that runs on commercial off-the-shelf servers. A network based on NFV can achieve much lower cost and much higher flexibility compared to a traditional computer network. Based on ETSI NFV ISG [7], the NFV architecture is composed of three key elements (Fig. 1) [6]:

1) Network Function Virtualization Infrastructure (NFVI): NFVI is composed of the commercial-off-the-shelf hardware and the abstractions of the computing, storage and network resources.The abstraction is achieved through a virtualization layer based on hyperviser, which decouples the virtual resources from the underlying physical resources.

2) Virtual Network Functions (VNF): A VNF is a virtualized functional block within a network infrastructure that has well defined external interfaces and functional behavior. Examples of VNFs include virtualized residential gateway, virtualized firewall, and virtualized load balancer. VNFs can be realized through virtual machines.

3) NFV Management and Orchestration (NFV MANO): NFV MANO performs the orchestration and lifecycle management of NFVI resources and VNFs. It is in charge of the operations of the VNFs such as the configuration of the VNFs and the infrastructure these functions run on. It covers three functional blocks: NFV Orchestrator, VNF Managers, and Virtualized Infrastructure Manager. NFV MANO also interacts with the (NFV external) business support systems (OSS/BSS) landscape, which allows NFV to be integrated into an already existing network-wide management landscape.

B. DDoS Attack Techniques

A typical type of DDoS attack based on true source IPs utilizes botnets [8]. The bot masters can easily orchestrate a large scale of bot-nodes on the Internet to perform attacks. However, the botnet-based attacks induce a high cost for compromising and controlling attacks nodes. The true identity of attack nodes can also be easily detected and blacklisted so that the overall cost of using bot-nodes is much higher than the spoofing-based attacks. Therefore the bot-node attack usually only takes a small percentage of the entire attack volume.

On the other hand, the spoofed DDoS attack techniques such as SYN Floods can effectively hide the true identity of attacking nodes and also require much less resources to launch

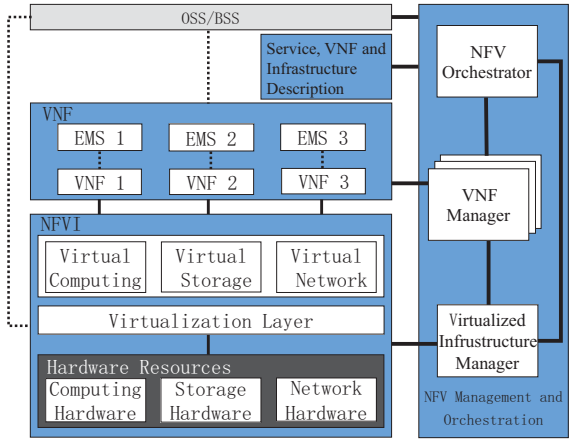


Fig. 1. A typical architecture of the NFV framework (adapted from [6]).

the attacks. In SYN Floods, the attacker fabricates a large number of TCP SYN packets using spoofed source IPs to initiate hand-shakings with the victim, which overflows the backlog on the victim and therefore prevents legitimate SYN requests from being processed. In this paper, we will focus on how to defend against SYN Floods using NFV.

C. DDoS Defense Strategies

There are many DDoS solutions in the literature. They can be roughly divided into three categories: (1) solutions on the source only; (2) solutions on the target only; and (3) solutions involving intermediate routers. Solutions on the source only block attack traffic from the source domain, so they cannot get into the Internet. For example, the BCP 38 standard [9] enforces ingress routers to verify that the packets from its administered network are using legitimate source IP addresses and filters traffic that uses IP addresses outside its subscribed range. This type of solution has proven to be effective and can mitigate IP-spoofing attacks completely when all ISPs adopt BCP38. However, it is not practical due to the lack of incentive and the difficulty of enforcement. The solutions involving immediate routers include IP traceback mechanisms [10], and packet marking and filtering mechanisms [11], [12]. This type of solution can only be effective if there are a sufficient number of participating routers along the routing paths. [13] talks about the use of NFV and SDN to implement a DDoS defense system, Bohatei. It relies on vendor provided directed acyclic graphs (DAG) for different attack types. The third type of defense involves only the destination infrastructure, for example a firewall or a proxy, to perform filtering of an attack flow. However, dedicated high capacity hardware for DDoS may not be practical for small/medium enterprises due to the high cost. On the other hand, using the cloud based third party proxy (e.g. Prolexic [5]) also brings privacy concern issue. The emerging of NFV technology introduces new opportunities for enterprises to find effective DDoS defense [14]. In this paper we focus on how to utilize NFV for SYN Floods attacks.

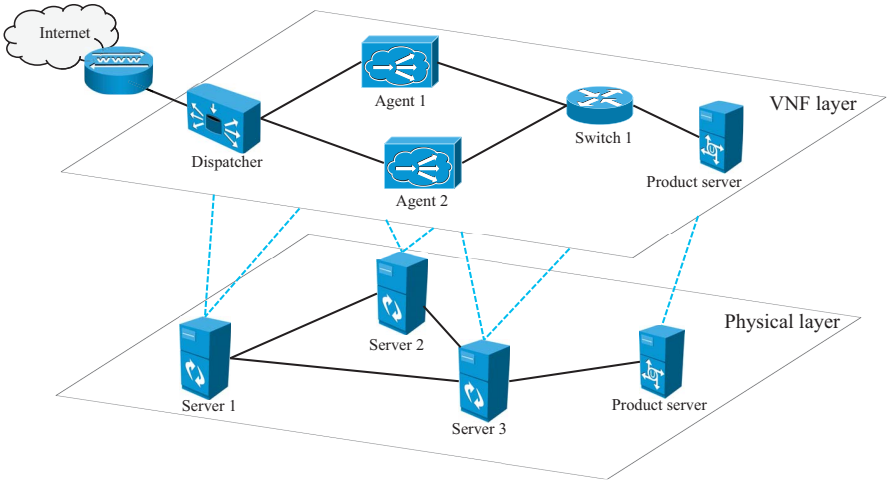


Fig. 2. A NFV network topology that defends DDoS attacks on the target using multiple agents.

III. Dynamic DDoS Defense Architecture

In this section, we first discuss the proposed framework of for DDoS defense, then we present the DDoS detection and prevention algorithm and the purpose of NFV in this design.

A. Architecture

Fig. 2 shows an overview of the network topology that we propose to defend DDoS attacks. The defense network is an NFV network consisting of dynamically allocated virtualized network functions (VNFs). In the physical layer, we have one product server which provides online service to customers from the Internet, and three commodity servers which are connected to each other. Each commodity server hosts virtual machines to realize different virtual network functions, such as dispatcher, switches, and agents. The dispatcher is the gateway for packets to the virtual filtering system. This agents act as filters for attack traffic. The VNFs are organized in a way so that attack flows will be handled by filtering agents (Fig. 2). The overall packet handling work flow is illustrated in Fig. 3.

Dispatcher: The dispatcher is a simple forwarding device. It receives packets from the Internet and distributes them to one or multiple handling agents. It also forwards packets from the product server back to the Internet. One of the major functions of the dispatcher is load balancing, where packets are delivered to various agents without overloading them. The packet distribution is based on the packet source and destination addresses and it keeps tracking which connection is handled by which agent. It maintains a table which contains the source and destination addresses with ports, as well as the corresponding handling agent ID. When the dispatcher gets a packet that is not in its map, it will randomly select another agent with a lower workload. This way, the dispatcher does a load balancing for delivering packets to agents. Note that this table is updated when a flow ends or expires. Overall, the number of agents can be updated depending on the number of

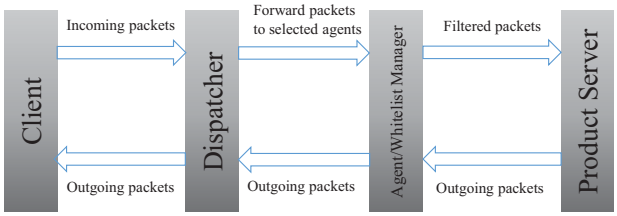


Fig. 3. A schematic diagram of components and their interactions.

packets the dispatcher receives and the load balancing function keeps each agent with a moderate load.

Agents: Each agent acts like a proxy or a firewall. It maintains a whitelist of legitimate source addresses (white-list) and some filtering rules to filter known attack traffic. The agents also verify the source IP addresses of all SYN packets through a spoofed handshaking process. Once a source IP from a client is verified, it initiates another spoofed SYN packet to complete the handshaking process for the verified source IP with the product server, and it updates the white-list correspondingly to add the legitimate source IP. Hence, a valid TCP connection transfer is completed.

B. DDoS Detection and Prevention Mechanism

Verification of valid connection: The flowchart presented in Fig. 4 shows the proposed algorithm. The figure shows flow of operation for both the dispatcher and an agent.

A client sends a SYN packet towards the product server to initialize a TCP connection; i.e., it starts a three-way handshaking. This handshaking is shown in Fig. 5(a). The dispatcher intercepts and gets the SYN packet first. It runs a load balancing process and decides which agent to use. To do this, the dispatcher has a table which maps each source/destination address pairs to an agent ID. This helps assign packets from the same flows to the same handling agents. If a matching pair is not found, the dispatcher will

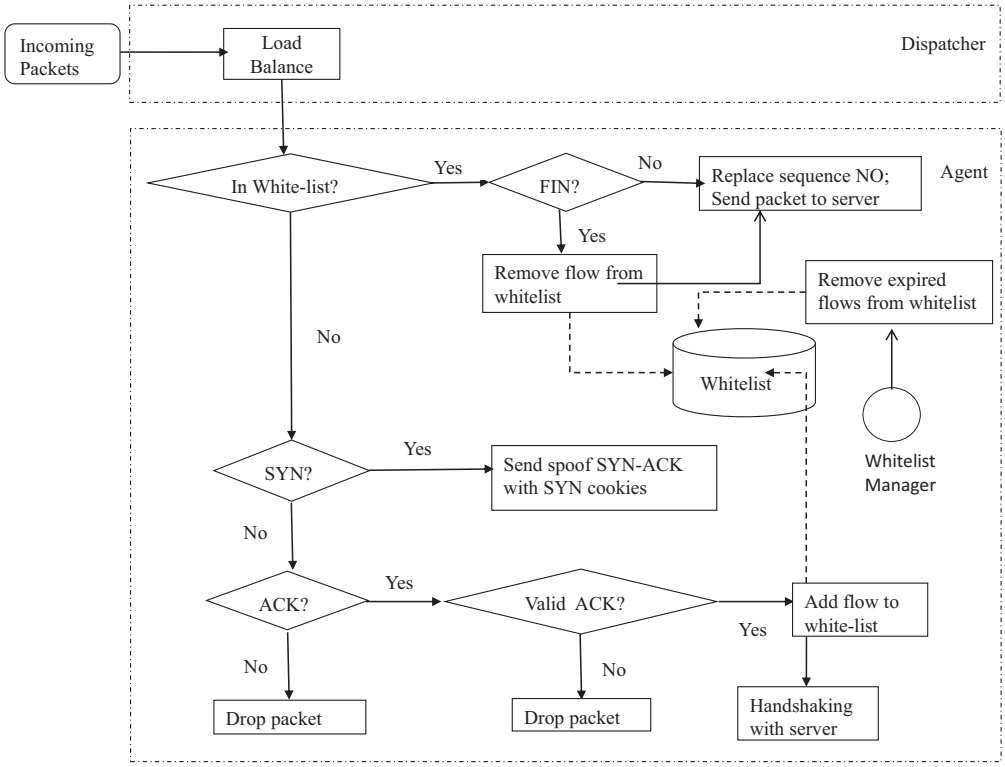


Fig. 4. Flowchart showing the operations of the dispatcher and an agent.

find an agent with a lower workload and assign the packet to the agent.

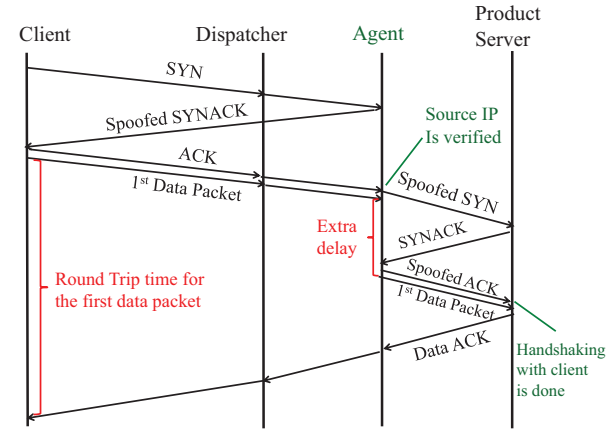
The agent receives the received packet, and it checks if the packet is on the whitelist. If the packet is not in the whitelist and it is a SYN packet, the agent attempts to verify if the SYN packet is from a spoofed source IP or not. In order to do so, the agent sends back a spoofed SYN-ACK with SYN cookies [15]. When SYN cookie is used, the sequence number from the server will be generated from the current time and from the IPs of the source and destination. This way the server does not have to store the sequence number for each SYN packet and therefore is efficient when most SYN packets are spoofed under a DDOS attack. The purpose of using SYN cookies is to turn the handshaking process to a stateless process which can greatly enhance the capacity of the agent to handle SYN packets. The client which is communicating from the Internet, gets the SYN-ACK and returns its ACK. The dispatcher should then forward this ACK packet to the corresponding handling agent, by looking up its mapping table. The client should not have any idea that it is actually communicating with an agent, instead of the server.

The agent checks if it is a valid ACK or not. This can be done through verifying whether the acknowledge number from the client matches the sequence number generated through the SYN cookies function. If it is a valid ACK, it adds the source address to the whitelist and starts a TCP spoofed handshaking with the product server. That is, it sends a spoofed SYN packet to the product server (Fig. 5(a)).

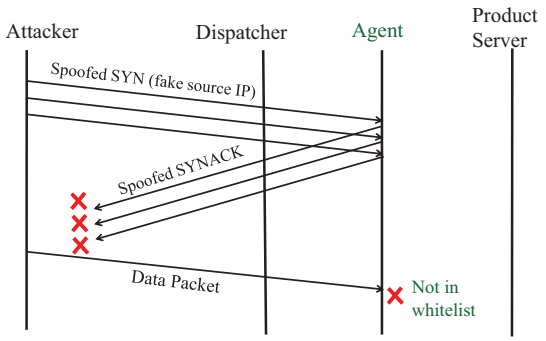
Upon receiving a spoofed SYN initiated by the agent, the product server processes it by returning a SYN-ACK containing a server sequence number. Upon receiving the SYN-ACK from the product server, the agent sends the spoofed ACK to the server. At this point, a connection between the verified client and the product server is established. All subsequent data packets in between the verified client and product server will go through the same agent to keep the connection alive. The verified source address will be added to the whitelist.

Special Cases: It may happen that the data packet from a client reaches the agent before the whitelist is updated. That is, before the agent could add the source address from a SYN or an ACK packet of the sender to the whitelist, a data packet arrives at the agent. The agent should hold the packet until it has finished a successful handshake with the client and the product server. After that, it will forward the data packet to the product server.

Another exceptional case is that sometimes a connection can get terminated without any FIN packet exchange. Thus, the agent needs some way to know when to remove a source address from the whitelist. The agent will do this by maintaining a timer. This works as follows: the agent starts a timer whenever it adds a source address to the whitelist. The timer is specific to that source address entry in the whitelist. When the agent gets a packet from a valid source, i.e., from a source that is in the whitelist, it checks whether it is a FIN packet or not; if not, it resets the timer. The agent gets to remove a source from the whitelist, if a timer expires.



(a)



(b)

Fig. 5. These figures show the spoofed TCP handshaking with a real client and an attack (a) exchange of SYN, SYN-ACK and ACK packets, (b) attack scenario where a DoS attack is prevented.

Attack Scenario: The handshake procedure in the scenario of an attack is shown in 5(b). When the dispatcher gets a SYN packet from an attacker, it forwards the packet to a corresponding agent. The agent sends a SYN-ACK with SYN cookies. If the source address in the SYN packet is spoofed, the SYN-ACK will be sent to an invalid IP address or an incorrect source IP, and the real attacker would never receive this SYN-ACK, which contains critical information to generate a correct ACK packet. Then, the agent would not receive a correct ACK back from the attacker, and SYN Flood attackers would not get on the whitelist. As a result, the attacker's traffic would not be forwarded to the product server. This process protects the product server.

On the other hand, if the attacker sends a fabricated ACK to attempt to finish the handshaking, the agent verifies it is not a valid ACK by checking its ACK number, then it drops the packet. Any other type of packet whose source address is not on the whitelist will be dropped by agent.

C. Deployment of Dynamic Agents

The purpose of using NFV in this idea of DDos mitigation is to use dynamic virtual machines to implement all network functions including agents. That is, the agents will be deployed dynamically according to the requirements. The number of agents will change over time. An automated mechanism of synthesizing this deployment plan will be solved in a future research work. In the rest of this section, we briefly discuss how the dispatcher deals with the increased or decreased number of agents.

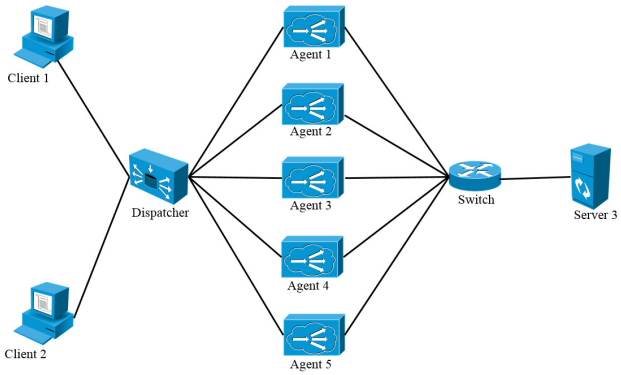


Fig. 6. The simulation topology.

When new agents are added: The dispatcher executes a dynamic load balancing to distribute the incoming packets to the agents for inspection. The system will increase the number of agents when there is a huge flow of packets. That is, whenever there is an increased number of incoming packets, the system will deploy new agents by creating more virtual machines that run the functionality of an agent. The dispatcher will then add new entries to include the new agents. For example, it will add the new agent ID into the agent list for the agent selection function to choose from. This way, the dispatcher can send new SYN packets to the new agent.

When agents are removed: When there is a decrease of inflow traffic, the system may decide to turn off some agents. Before putting an agent to sleep, the system should send a caveat to the dispatcher to notify the dispatcher that the agent will be off list after a period of time. That is, it has to make sure that the dispatcher does not allocate new flows to the retiring agent. The exiting flows on the retiring agent will be migrated to another agent. When the migration is done, the agent is put into sleep mode until it is needed again.

IV. Evaluation

Methodology: We develop a Java program to simulate the proposed defense architecture. Fig. 6 shows the simulation topology. For the proof of concept, we construct a network consisting of 2 clients, 1 dispatcher, 5 agents, 1 switch, and 1 server. Client 1 generates normal (good) traffic only, and client 2 is responsible for sending SYN Floods attack (bad) traffic. In our simulation, the normal traffic rate is 100 packets/second. The maximum packet processing rate at the server is 200 packets/second. The packet handling rate at each agent is 1000 packets/second. The buffer size at both the sever and agents are 200 packets. As described in Fig. 7(a), during the

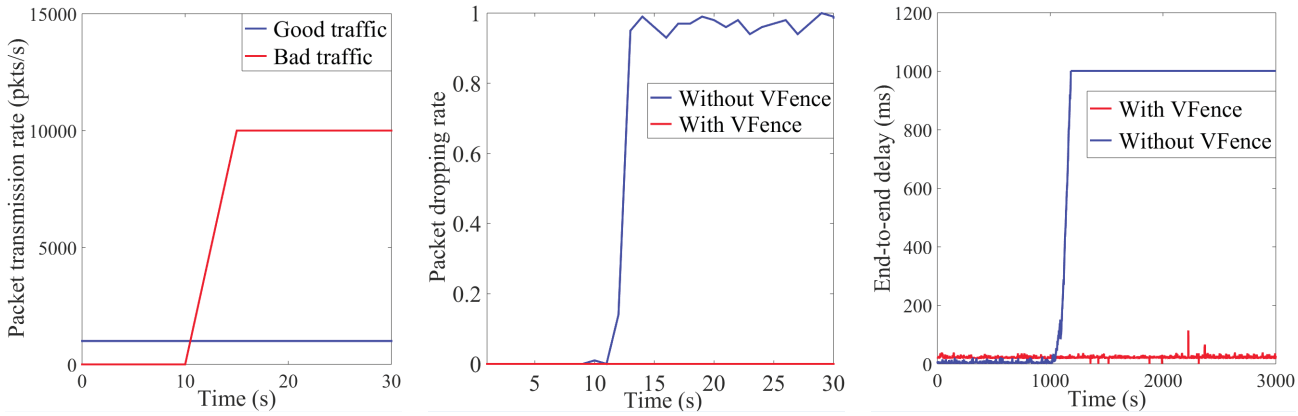


Fig. 7. (a) Attack plan of the experiment: SYN floods attack (10 times of good traffic volume) starts at time 10 and reaches max strength at time 15, (b) The packet dropping rate for good traffic under SYN floods attack, and (c) The packet delay for good traffic under SYN floods attack.

simulation, client 1 keeps sending normal traffic at 100 pkts/s, and client 2 starts sending SYN Floods attack traffic at time 10 second and increases the attack volume constantly until it reaches 1000 pkts/s at time 15s. After that, the attack volume maintains at 1000 pkts/s until the end of the simulation at time 30s. We choose two scenarios to compare with: the scenario with our proposed VFence system and the scenario without the proposed VFence system. We use the packets dropping rate and the processing delay in the network to be the metrics to evaluate the performance of the proposed VFence system.

Evaluation Results: Fig. 7(b) depicts the packets dropping rate across the time of the entire simulation. We can see that the attack traffic drastically increases the dropping rate (to nearly 100%) of the normal traffic when there is no defense. However, the dropping rate of normal traffic maintains at a very low level even under attack when VFence is in place. This is because VFence uses five agents (processing rate 5000 pkts/s) to block all the SYN floods so that only normal packets eventually arrive on the product server.

Fig. 7(c) presents the delay of data packets in normal traffic. The results show that the delay of normal traffic increases drastically upon the arrival of attack traffic when no defense is in place. We speculated the reason and found that the product server buffer is filled quickly as soon as attack traffic increases. However, when VFence is in place, packets delay is maintained at low level. Both results demonstrate that VFence is able to successfully defend the SYN Floods attack and effectively protect the targeted server.

V. Conclusion

This paper introduces a novel solution based on the NFV architecture to mitigate DDoS attacks. The defense mechanism leverages the flexible feature of NFV that allows the implementation of network functions dynamically according to the requirements. Our mechanism creates DDoS traffic filtering agents to handle external traffic to the targeted server. External traffic is directed to one of the agents according to a load-balancing idea and the agent forwards clean traffic to the destination. Our evaluation results on SYN Floods attacks demonstrate that the proposed solution can effectively reduce the DDoS attack flow to the target server. In future, we would like to develop an automated mechanism for planning dynamic deployment of traffic filtering agents with the attack intensity.

References

[1] Biggest internet attack in history threatens critical systems. http://www.ibtimes.co.uk/biggest-internet-attack-history-threatens-critical-infrastructure-450969.

[2] Swati Khandelwal. 602 gbps! this may have been the largest ddos attack in history. http://thehackernews.com/2016/01/biggest-ddos-attack.html.

[3] Haining Wang, Danlu Zhang, and Kang G Shin. Detecting syn flooding attacks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1530–1539. IEEE, 2002.

[4] Atlas q2 2015 update. http://www.slideshare.net/Arbor_Networks/atlas-q2-2015final.

[5] Prolexic routed. https://www.akamai.com/us/en/solutions/products/cloud-security/prolexic-routed.jsp.

[6] ETSI NFV ISG. Network Functions Virtualization White Paper 3: Network Operator Perspectives on Industry Progress. In *SDN and OpenFlow World Congress*, Oct. 2014.

[7] The etsi nfv isg homepage. http://www.etsi.org/technologies-clusters/technologies/nfv.

[8] Ryan Vogt, John Aycock, and Michael J Jacobson Jr. Army of botnets. In *NDSS*, 2007.

[9] P. Ferguson and D. Senie. Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing (bcp 38). http://tools.ietf.org/html/rfc2827.

[10] A John and T Sivakumar. Ddos: Survey of traceback methods. *International Journal of Recent Trends in Engineering*, 1(2):241–245, 2009.

[11] Haining Wang, Cheng Jin, and Kang G Shin. Defense against spoofed ip traffic using hop-count filtering. *IEEE/ACM Transactions on Networking (ToN)*, 15(1):40–53, 2007.

[12] Abraham Yaar, Adrian Perrig, and Dawn Song. Pi: A path identification mechanism to defend against ddos attacks. In *Security and Privacy, 2003. Proceedings. 2003 Symposium on*, pages 93–107. IEEE, 2003.

[13] Seyed K. Fayaz, Yoshiaki Tobioka, Vyas Sekar, and Michael Bailey. Bohatei: Flexible and elastic ddos defense. In *24th USENIX Security Symposium*, pages 817–832. USENIX, 2015.

[14] Carol J Fung and Bill McCormick. Vguard: A distributed denial of service attack mitigation method using network function virtualization. In *Network and Service Management (CNSM), 2015 11th International Conference on*, pages 64–70. IEEE, 2015.

[15] Syn cookies. http://cr.yp.to/syncookies.html.