Volume 66, Issue 2, 2022

Journal of Scientific Research

of

The Banaras Hindu University

MAC based model to Differentiate Flash crowd and Malicious traffic in SDN

Jitendra Patil[1], Vrinda Tokekar[2], Alpana Rajan[3], Anil Rawat[4]

[1,3,4] Raja Ramanna Centre For Advanced Technology, Indore-13, India
[2] Institute of Engineering & Technology Devi Ahilya Vishwavidyala Indore-17, India
jpatil@rrcat.gov.in

Abstract. The nature of computer network flash crowd traffic, which is generated by legitimate users accessing servers or other network resources are similar to the traffic generated by Distributed Denial of Service (DDoS) like attacks. With advancement in spoof packet generation tools, attacker may generate Multi-source Multi-destination Multi-protocol (MMM) traffic; characteristics of such traffic are very similar to on-going genuine/ flash crowd traffic in the network. In the case of Software Defined Network (SDN), attacker's target is controller plane. Controller plane in SDN is a centralized processing unit of the underlying network, which manages several data planes. Controller plane frames the policies and pushes forwarding rules to the data planes. Data planes just maintain the forwarding rules. Thus by overloading the SDN controller, functionality of complete computer network will be hampered. In this paper, we have proposed Media Access Control (MAC) address based Model to Differentiate Flash crowd and Malicious traffic in SDN (MDFMS). Novelty of the proposed model is to detect, locate and mitigate the source of Traditional DDoS (T-DDoS) and MMM-DDoS traffic. MDFSM has been implemented on separate machine to avoid any additional computing load on SDN controller. It also preserves the original design of the SDN architecture. Proposed model has been evaluated under various scenarios and encouraging results have been obtained to differentiate T-DDoS and MMM-DDoS from benign flash crowd traffic.

Keywords: SDN, Multi Destination DDoS, MAC

1 Introduction

To introduce the centralized control and capability of dynamic programming to all the underlying network devices, Software Defined Network (SDN) concept has been involved. SDN is new networking model in which control plane and data plane decouples and operate separately. Control plane is a centralized process unit which can control several data planes. Data planes are functioned like Ethernet switches without any local controller in the box. All the decisions to forward or drop the packet is the decision of centralized controller plane (more number of control plane may be there for large network). Data planes just maintain the forwarding rules received from controller. This makes the control plane an award winning target for the attackers.

Traditional Distributed Denial of Service (T-DDoS) is a prevalent attack, normally targeted services are web servers, file servers etc. T-DDoS exploit the vulnerability associated with Transmission Control protocol (TCP) three-way-connection [1,2]. To launch this attack, attacker sends crafted malicious TCP-SYN packet to the targeted server, in reply to this targeted server sends ACK-SYN to the source (Internet Protocol) IP. At this stage, attacker host goes silent and targeted host goes into waiting stage till time out of the TCP session. Such type of large number of malicious connections does not left targeted server with enough resources to respond to other genuine requests in queue.

In a SDN context, Malicious Traffic (MT) generated by T-DDoS, contributes significantly to saturate the SDN controller [3,4]. There are many detection mechanisms have already been proposed and provide solution to detect T-DDoS traffic i.e. attack on targeted services may be a file server, email server or web server. When we talk about targeted service, means majority of traffic associated with same destination IP. Frequency of destination IP is the basic key adopted by entropy based proposed solutions. Some solutions were based on SYN-proxy implemented at controller or at data plane to verify the source host. But in this case only TCP traffic are handled. Some models in the literature are based on Machine Learning (ML) uses parameters like rate of source IP (rSIP), rate of Destination IP (rDIP), rate of source port, rate of destination port, rate of change in bytes received and their deviation etc. ML and Entropy based models are associated with thresholds and accuracies are dependent on training datasets.

In SDN based networking, aim of attacker is to saturate SDN controller processing capabilities to collapse the complete network. Multi-source Multi-destination Multi-protocol Malicious Traffic (MMM-MT) are similar to genuine traffic and are difficult to detect, locate and mitigate source of MT. To discover the source of MT, pretending like genuine traffic and block them at data plane, motivated us to design and implement "Media Access Control address (MAC) based model to Differentiate Flash crowd and Malicious traffic in SDN" (MDFMS).

Novelty of the proposed model is to identify the basic nature of Flash Crowd Genuine Traffic (FCGT) and malicious (T-DDoS + MMM-DDoS) traffic, so that MT may be detected, located and blocked at the source. Proposed model MDFMS has been designed without any modification in original design of the SDN architecture. MDFSM has been evaluated under various scenarios and encouraged results have been obtained to differentiate T-DDoS/MMM-DDoS flows from FCGT.

DOI: 10.37398/JSR.2022.660217

This paper has been organized in five sections. Section II provides related work in the domain. Section III gives back ground on SDN, DDoS attack and its impact. Section IV depicts functionality of proposed model MDFMS and performance evaluation. Discussion and comparison of MDFMS has been given in section V. Section VI concludes the strength and weakness of the proposed model and proposed possible future enhancement in MDFMS.

2 Related work

MMM is a unique type of DDoS attack discovered and discussed in this paper for SDN based networking infrastructure. Models proposed by researchers are helped us to enrich our understanding about presence of DDoS in SDN. Broadly detection mechanisms discussed in this section can be categorized under analogy based and pattern matching detection systems. Analogy based detection based systems are normally employed with machine learning, entropy, neural network algorithms to detect presence of attacks. These systems are associated with false positive returns but capable to detect wide range of attacks. However, pattern matching proposed models are able to detect accurately the attacks for which it is designed.

The researchers have proposed several useful methods for distinguishing DDoS attack traffic from the flash crowd traffic. Thapngam et al. [5] proposed a model which is based on behavior of net flows using Pearson's correlation coefficient to distinguish DDoS attack traffic from flash crowd traffic generated by benign users. However, a constant observing data may introduce more processing time.

Distribution of packet size approach adopted by Zhou et al. [6] and proposed a model to differentiate slow DDoS attack in presence of legitimate traffic. However packet size can be customized to pretend to legitimate traffic. Xiang et al. [7] used generalized entropy and information distance metric to detect low-rate DDoS traffic and trackback the Source Internet Protocol (SIP) address. However the source of spoof IP cannot be tracked. Hoque et al. [8] proposed a model based on statistical properties of net-flows to measure Feature Feature Score (FFSc) to detect DDoS traffic. Threshold score of FFSc are calculated on entropy of SIPs, variation of SIP and packet rate. Calculated threshold value is static in nature which make this approach rigid in nature. Zhang et al. [9] explored to use of TCP congestion control mechanism and designed Congestion Participation Rate (CPR) metric to achieve per-flow level detection. This method achieved detection rate 100% and false positive rate 1.625% with a threshold value of 0.63.

Fichera et al. [10], proposed OPERETTA, an openflow-based approach to mitigate TCP-SYN flooding attacks targeting towards against web servers. This mechanism can be configured for centralized and delocalized controllers. It is an OpenFlow-based approach implemented in the controller and inspects TCP-three-way connection requests for detecting and rejecting malicious requests. OPERETTA module acts as a proxy to the client machine and upon successful verification, drop the connection by sending RST and install flow entry in the flow table. So that when client machine tries again to establish the connection, it does not need to negotiate with controller. Mitigation process is based on MAC address instead of IP to avoid covert attack. This model rejects the first attempt to establish TCP-three-way and force client machine to initiate the connection again. This functionality of OPERETTA introduces delay in establishing TCP connection. OPERETTA is fully deployed at controller and act as end server to verify each new TCP connection, overload the controller CPU under no attack condition.

To overcome the limitations in OPERETTA, SLICOTS, a model which is based on watching the TCP-SYN traffic in controller and decides whether the connection is valid or not is proposed by Mohammadi et al. [11]. Unlike OPERETTA, SLICOTS does not send SYN-ACK packet to client machine, rather it maintains pending_list_table with status field along with other information like Source MAC (SMA), Destination MAC (DMA), Source TCP port (STP), and Destination TCP port (DTP). When a new TCP packet reaches to controller, SLICOTS first verifies the type of incoming packet. If the packet is SYN, SLICOTS stores SMA, DMA, STP, and DTP values along with status field as "SYN". And install temporary forwarding rules for SYN packet on the OF switches. This temporary forwarding rule has hard timeout of three seconds considering the timeout value for waiting SYN-ACK packet (Paxson et al., [12]). Upon receiving SYN-ACK from server end, SLICOTS change the value of status field to SYN-ACK and then finally wait for ACK packet. Upon ACK packet received by SLICOTS, entry is removed from the pending_list table and install forwarding rules with normal hard timeout. Why ACK packet reaches to controller though forwarding rule with hard timeout of three seconds already in the flow table, is not clear. Mitigation process is based on MAC address instead of IP but blocking on MAC, would also block legitimate traffic as well. Performance of SLICOTS depends on threshold value of K. K is a maximum size of the pending_list and in the worst case it is n x K where n is the number of hosts..

A model named SAFETY has been proposed to detection and mitigate TCP-SYN Flooding attack by Kumar et al., [13]. This model claims for early detection and mitigation of TCP-SYN flooding attack using entropy. This model proposes two major components: a detection unit and a mitigation unit. Functionality of detection unit is based on values in the counter. Upon TCP-SYN packet comes to the controller, counter increment by one, upon SYN-ACK, counter decrement by one and upon RST, decrement by one. After Δt time interval, entropy of the windows is calculated. If it is more than threshold value for K times, a detection of attack alarms are raised which is followed by mitigation process. Mitigation process identifies the victim and attacker IP and blocks the traffic at edge switch. SAFETY assumed the attacker traffic always towards single destination IP address using TCP-SYN only. Spoof traffic from UDP and ICMP has not been considered. In practical scenarios, multiple destination attacks are possible, particularity in SDN environment to saturate SDN controller.

David et al. [14], proposed a model to differentiate flash crowd traffic from DDoS attack using efficient algorithm. This approach is based on dynamic entropy calculation on two network parameters destination IP address and packet size. However this approach does not support for MMM-DDoS attack traffic and also it does not locate and mitigate the source of MT.

There are few review papers [14,15,16,17] related to discrimination of flash crowd from malicious traffic. Solutions discussed in this section are considered T-DDoS traffic. T-DDoS attacks are designed to target end servers and in a SDN environment it impacts SDN controller as well. Generally solutions are designed to detect TCP-SYN flooding attack but to attack on SDN controller any type of IP traffics are sufficient to consume its resources. Thus we focused on MMM-DDoS attack pattern and proposed a light weight, scalable and easy to implement MDFMS model in this paper.

3 Background in designing MDFMS

This section discusses SDN functionality and nature of FCGT and malicious (T-DDoS + MMM-DDoS) traffic to understand how SDN functionality can be the award winning target for attackers.

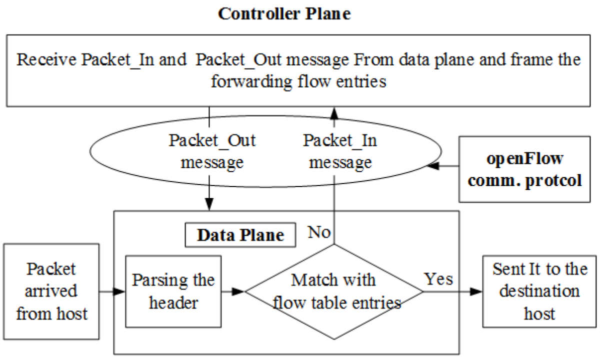
3.1 Working of SDN

SDN is a new networking model in which control and data plane are decoupled and operate separately [18]. Control plane is the central brain of the complete networking infrastructure. Control plane formulates the forwarding rules and pushes to the data planes. There are several data planes in the network, all are controlled by controller plane. Data planes maintain the forwarding rules received form controller plane.

When a packet received by the data plane, data plane match it with the existing forwarding rules, if there is a hit, packet gets forwarded or drops according to the forwarding rule and if there is miss, packet get encapsulated in packet_in message and send it to the controller plane. Controller plane, as per the networking logic, generates forwarding rule and push it to data plane using packet_out message. All the communication between data and controller plane takes planes over OpenFlow (OF) communication protocol [19]. OF provides a set of Application Programming Interfaces (API) to controller plane to install flow table entries at data plane [20, 21]. This way, for any packet get missed at data plane, controller and data plane come into the action. As long as traffic is normal, control plane and data plane works fine and can handle organization level local area network traffic efficiently. Problem starts when there is a MT hits the data plane at a high Packet Per Second (PPS). Attacker take use of SDN functionality and crafts the spoof packet in such a way that it get miss at data plane all the time which consequently consume SDN controller CPU usage as well as the memory available with data plane to store forwarding rules. Under such situation SDN controller get saturated and could not process any genuine or malicious request. This way complete network gets paralyzed and becomes inaccessible. Functionality of SDN and sample flow table entry is shown in Fig. 1. When a packet reaches to the data plane, data plane matches it to the available flow table entries at that instance. If there is a match, packet get forwarded to the destination or drop otherwise inform the controller using packet_in message encapsulating the received packet.

3.2 Impact of Malicious Traffic on SDN controller

Functionality of SDN is described in section 3.1. Using the virtual experimental setup depicted in section 3.2 over mininet, impact of MT on SDN controller is evaluated. Mininet is network emulator which supports SDN based networking, OF protocol, OF virtual switch and open source various SDN controller [22]. Hosts <h1-h10> were used to generate MT at various packets per seconds (PPS) targeting web server at port 80 (please refer section 4.2). Here aim of the attacker is not web server rather generate more flow table entries to consume SDN controller resources. Every time MT generated by compromised hosts are unique in nature using spoof SIP and Source Port Number (SPN), so that each packet get miss at Data Plane. It can be seen in the Fig. 2. that CPU usage of SDN controller starts saturated at 1000 PPS. Under this situation SDN controller becomes extremely slow to process genuine request received from user hosts.


Controller Plane
Receive Packet_In and Packet_Out message From data plane and frame the forwarding flow entries
Packet_Out message
Packet_In message
openFlow comm. protcol
Data Plane
No
Packet arrived from host
Parsing the header
Match with flow table entries
Yes
Sent It to the destination host

(a)

```
cookie=0x0,     duration=31.231s,     table=0,     n_packets=0,     n_bytes=0,
idle_timeout=30,  priority=65535,  tcp,  in_port  ="s1  eth12",  vlan_tci=0x0000,
dl_src=2c:33:7a:55:df:d9,    dl_dst=3a:d9:ab:4c:7c:5a,    nw_src    =10.0.0.12,
nw_dst=10.0.0.11,nw_tos=0, tp_src=44838, tp_dst = 80 actions=output:"s1-eth11"
```

(b)

Fig 1. (a) SDN functionality flow (b) net-flow at Data Plane

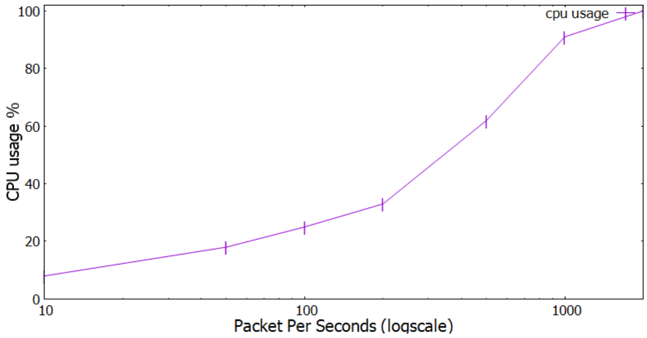
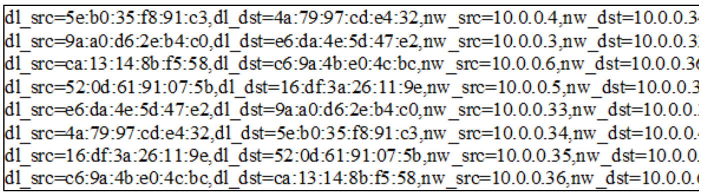


Fig. 2. Impact of unique PPS on SDN controller

3.3 Flash Crowd and MMM Malicious Traffic

In this section nature of flash crowd and malicious traffic has been identified through series of experiments. Details on test experimental setup and parameters users are given in section 4.2.

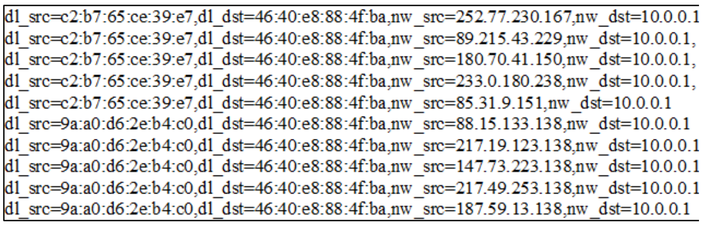
Experiment 1- Generation of normal traffic: Normal traffic has been generated using hping3, wget and scapy from hosts <h2> to <h32> destination to <h33> to <h64>. Snap shot of forwarding flow table entries are shown in Fig. 3. It may be noted that flow table entries recorded during normal traffic maintains single SMA-SIP relationship.



```
dl_src=5e:b0:35:f8:91:c3,dl_dst=4a:79:97:cd:e4:32,nw_src=10.0.0.4,nw_dst=10.0.0.3
dl_src=9a:a0:d6:2e:b4:c0,dl_dst=e6:da:4e:5d:47:e2,nw_src=10.0.0.3,nw_dst=10.0.0.3
dl_src=ca:13:14:8b:f5:58,dl_dst=c6:9a:4b:e0:4c:bc,nw_src=10.0.0.6,nw_dst=10.0.0.3
dl_src=52:0d:61:91:07:5b,dl_dst=16:df:3a:26:11:9e,nw_src=10.0.0.5,nw_dst=10.0.0.3
dl_src=e6:da:4e:5d:47:e2,dl_dst=9a:a0:d6:2e:b4:c0,nw_src=10.0.0.33,nw_dst=10.0.0.
dl_src=4a:79:97:cd:e4:32,dl_dst=5e:b0:35:f8:91:c3,nw_src=10.0.0.34,nw_dst=10.0.0.
dl_src=16:df:3a:26:11:9e,dl_dst=52:0d:61:91:07:5b,nw_src=10.0.0.35,nw_dst=10.0.0.
dl_src=c6:9a:4b:e0:4c:bc,dl_dst=ca:13:14:8b:f5:58,nw_src=10.0.0.36,nw_dst=10.0.0.
```

Fig. 3. Flow table entries generated by Normal Traffic

Experiment 2- Generation of T-DDoS malicious traffic: Hosts <h2-h10> are compromised hosts generating T-DDoS attack traffic targeting web server at TCP port 80. Recorded flow table entries in Fig. 4., shows that source MAC is same with different SIPs targeting destination web server IP 10.0.0.1. Here SMA "c2:b7:65:ce:39:e7" is associated with <h2> and "9a:a0:d6:2e:b4:c0" with <h3>.

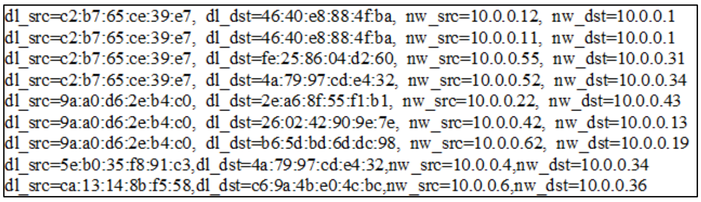


```
dl_src=c2:b7:65:ce:39:e7,dl_dst=46:40:e8:88:4f:ba,nw_src=252.77.230.167,nw_dst=10.0.0.1
dl_src=c2:b7:65:ce:39:e7,dl_dst=46:40:e8:88:4f:ba,nw_src=89.215.43.229,nw_dst=10.0.0.1,
dl_src=c2:b7:65:ce:39:e7,dl_dst=46:40:e8:88:4f:ba,nw_src=180.70.41.150,nw_dst=10.0.0.1,
dl_src=c2:b7:65:ce:39:e7,dl_dst=46:40:e8:88:4f:ba,nw_src=233.0.180.238,nw_dst=10.0.0.1,
dl_src=c2:b7:65:ce:39:e7,dl_dst=46:40:e8:88:4f:ba,nw_src=85.31.9.151,nw_dst=10.0.0.1
dl_src=9a:a0:d6:2e:b4:c0,dl_dst=46:40:e8:88:4f:ba,nw_src=88.15.133.138,nw_dst=10.0.0.1
dl_src=9a:a0:d6:2e:b4:c0,dl_dst=46:40:e8:88:4f:ba,nw_src=217.19.123.138,nw_dst=10.0.0.1
dl_src=9a:a0:d6:2e:b4:c0,dl_dst=46:40:e8:88:4f:ba,nw_src=147.73.223.138,nw_dst=10.0.0.1
dl_src=9a:a0:d6:2e:b4:c0,dl_dst=46:40:e8:88:4f:ba,nw_src=217.49.253.138,nw_dst=10.0.0.1
dl_src=9a:a0:d6:2e:b4:c0,dl_dst=46:40:e8:88:4f:ba,nw_src=187.59.13.138,nw_dst=10.0.0.1
```

Fig. 4. Flow table entries generated by DDoS

Experiment 3- Generation of MMM malicious traffic: In this experiment MMM traffic has been generated by <h2-h3> and hosts <h4-h6> were configured to generate normal traffic. Closely monitoring the recorded flow table entries shown in Fig. 5. reveals that there are many SIPs associated with single SMA. But in case of <h4> and <h6>, there is single association of SMA with SIP.

Above three experiments reveals that association of SMA-SIP on each inPORT is a unique identity to differentiate flash crowd from malicious traffic. Normal/Flash crowd hosts maintain only one SMA-SIP association irrespective of Destination Internet Protocol (DIP) address, destination port number, source port number or any other parameter in the net flows. This association is unique in nature to differentiate flash crowd from malicious traffic and thus base to design proposed model MDFMS.



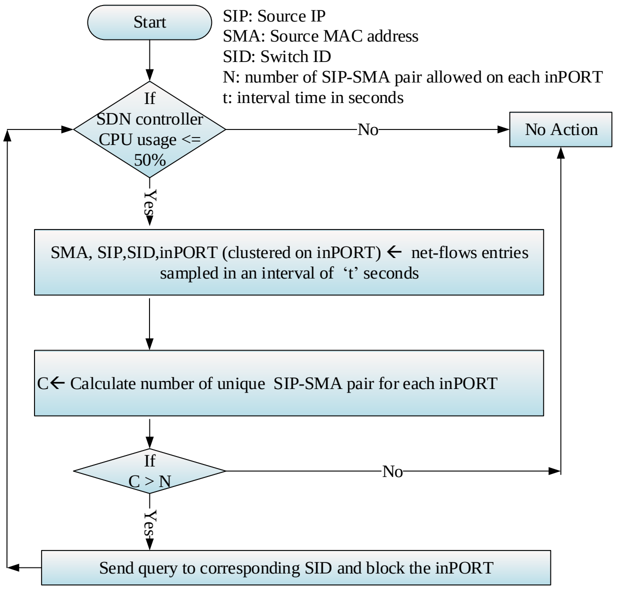
```
dl_src=c2:b7:65:ce:39:e7,  dl_dst=46:40:e8:88:4f:ba,  nw_src=10.0.0.12,  nw_dst=10.0.0.1
dl_src=c2:b7:65:ce:39:e7,  dl_dst=46:40:e8:88:4f:ba,  nw_src=10.0.0.11,  nw_dst=10.0.0.1
dl_src=c2:b7:65:ce:39:e7,  dl_dst=fe:25:86:04:d2:60,  nw_src=10.0.0.55,  nw_dst=10.0.0.31
dl_src=c2:b7:65:ce:39:e7,  dl_dst=4a:79:97:cd:e4:32,  nw_src=10.0.0.52,  nw_dst=10.0.0.34
dl_src=9a:a0:d6:2e:b4:c0,  dl_dst=2e:a6:8f:55:f1:b1,  nw_src=10.0.0.22,  nw_dst=10.0.0.43
dl_src=9a:a0:d6:2e:b4:c0,  dl_dst=26:02:42:90:9e:7e,  nw_src=10.0.0.42,  nw_dst=10.0.0.13
dl_src=9a:a0:d6:2e:b4:c0,  dl_dst=b6:5d:bd:6d:dc:98,  nw_src=10.0.0.62,  nw_dst=10.0.0.19
dl_src=5e:b0:35:f8:91:c3,dl_dst=4a:79:97:cd:e4:32,nw_src=10.0.0.4,nw_dst=10.0.0.34
dl_src=ca:13:14:8b:f5:58,dl_dst=c6:9a:4b:e0:4c:bc,nw_src=10.0.0.6,nw_dst=10.0.0.36
```

Fig. 5. Flow table entries generated by MMM-DDoS

4 Proposed Model MDFMS

4.1 MDFMS Functionality

MDFMS has been designed to operate on separate machine. It collects the net flows at an interval of 't' seconds from data plane and clustered SMA-SIP pairs for each inPORTs of each switch in the network. In this implementation 't' is taken as 3 seconds, assumed attacker traffic at 10000 PPS would generated maximum 60000 entries (maximum 30000 pairs) in the table. MDFMS has been tested to process 60000 entries and clustered the SMA-SIP for each inPORT in an average time of 0.831 seconds. Flow sequence of proposed model MDFMS is depicted in Fig. 6.



Start
SIP: Source IP
SMA: Source MAC address
SID: Switch ID
N: number of SIP-SMA pair allowed on each inPORT
t: interval time in seconds
If SDN controller CPU usage <= 50%
No
No Action
Yes
SMA, SIP,SID,inPORT (clustered on inPORT) ← net-flows entries sampled in an interval of 't' seconds
C← Calculate number of unique SIP-SMA pair for each inPORT
If C > N
No
Yes
Send query to corresponding SID and block the inPORT

Fig. 6. Flow sequence of proposed model MDFMS

MDFMS gets activated only when CPU usage of controller goes above 50%. Under normal scenario, it has been observed that CPU usage remains in the range of 5-10% only. During peak hours (flash crowd) it may hit upto 30-50%. If CPU usage goes above 50%, MDFMS gets activated, calculate unique SIP and Source MAC Address (SMA) count for each inPORT. If multiple association of SIP-SMA are found at inPORT, MDFMS initiates locate and mitigate process to remove such flow table entries and block corresponding inPORT at data plane. Location of source of malicious traffic is a combination of Switch-ID (SID) and inPORT number. MDFMS may be executed in multiple instances for each switch or some group of switches in the network which makes it scalable and responsive for large network.

4.2 Experimental setup

To evaluate the performance of MDFMS, a virtual experimental setup has been built using Mininet. This setup compromise of 128 hosts connected over 1 Gbps network link. Details of various components and parameters used for this setup are given in Table 1. Layout of experimental setup is shown in Fig. 7.

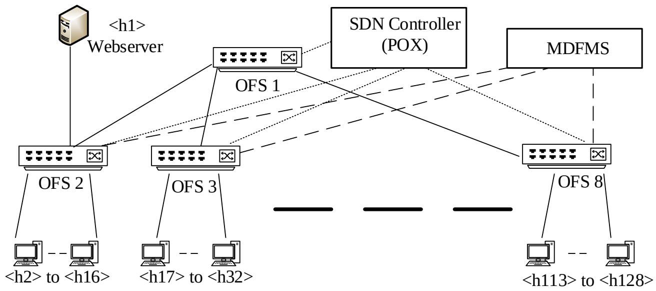


Fig. 7. Layout of experimental setup to analyze the performance of MDFMS under various scenarios.

Table 1. Parameters used to setup experimental setup

Network components/tools	Specifications
Base machine hardware	i5 30 GHz, RAM 16 GB, Windows 8.1
Virtual machine for Mininet	Cores 2 nos., RAM 4 GB, OS Ubuntu 18.04
Victim server	Webserver (apache)
Emulator	Mininet (2.2.26)
OF vSwitch	v.2.9.2
MDFMS VM	Ubuntu 18.04 loaded with OF libraries
Hosts <h1-h128>	Ubuntu 14.04 connected over 1 Gbps link
SDN Controller	POX (version)
Tool used to generate DDoS traffic	scapy, wget, mgen, hping3
SDN controller CPU usage threshold	50%
Normal traffic hosts	Hosts <h2-h10>@5-10 PPS (One connection per host)
Flash crowd traffic hosts	Hosts <h61-h110>@50-100 PPS (One connection per host)
Traditional DDoS traffic hosts (single destination using TCP)	Hosts <h11-h60> @50-100 PPS (Destination host- webserver)
MMM-DDoS traffic hosts (multiple destination using TCP, UDP, ICMP)	Hosts <h11-h60> @50-100 PPS (Destination hosts- <h61-h128>)

4.3 Performance Evaluation

In this section MDFMS has been evaluated under three scenarios. Number of attacker hosts and flash crowd hosts are kept at same PPS to mix these traffics more uniformly.

4.3.1: Estimation of CPU usage: To evaluate the usage of computing resources of SDN controller, hosts <h2-h10> were configured for normal traffic and 50 hosts <h61-h110> were configured for flash crowd traffic. And there are no hosts generating malicious traffic. CPU usage of SDN controller without MDFMS is shown in Fig. 8(a). During establishment of net-flow entries, picks (28-30%) are observed and rest of the time CPU operated at 4-5% load. In presence of MDFMS, same scenario has been repeated. It is noticed in Fig. 8(b). that CPU usage of SDN controller is almost same as without MDFMS. Advantage of operating MDFMS on separate machine is clearly visible. To analyze MDFMS computing load on SDN controller, we set the CPU usage threshold to 2%, so that MDFMS gets activated. Observations recorded during the experiment reveals that MDFMS does not add any load on SDN controller. Load on MDFMS machine has been analyzed and found it in the range of 2 to 3%.

4.3.2 T-DDoS attack detection in presence of flash crowd traffic: Traditional DDoS attack is normally targets end servers. In this evaluation, 50 hosts <h11-h60> were configured to attack webserver in presence of flash crowd traffic generated by 50 hosts <h61-h110>. It can be seen in the graph depicted in Fig. 9. that MDFMS took about 3 seconds to detect and block all malicious sources (about 15000 net flows) in an average time of 0.212 seconds. Eventually CPU usage of SDN controller returns to normal situation.

4.3.3 MMM-DDoS attack detection in presence of flash crowd traffic: MMM-DDoS attack traffic is almost similar to generic traffic generated by genuine hosts during day to day activity in an organization. MMM-DDoS attack traffic cannot be determined by entropy on destination address. In this exercise, MMM-DDoS traffic has been generated in presence of flash crowd traffic. Obtained results are shown in Fig. 10. These results are similar to results shown in Fig. 9. Reason could be the operating mode of MDFMS, which is based on the SIP/SMA pair and is independent of DIP, Destination port number or protocols used. All the attacker hosts were detected in a cycle of 3 seconds and blocked in about 0.213 seconds. After 4th second only 100 net-flows are left which established by 50 flash crowd hosts.

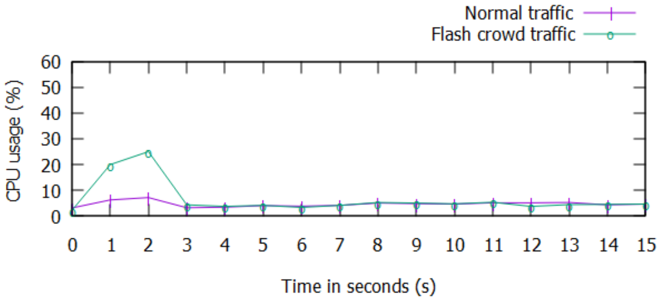


Fig. 8. a) Load on SDN controller without MDFMS

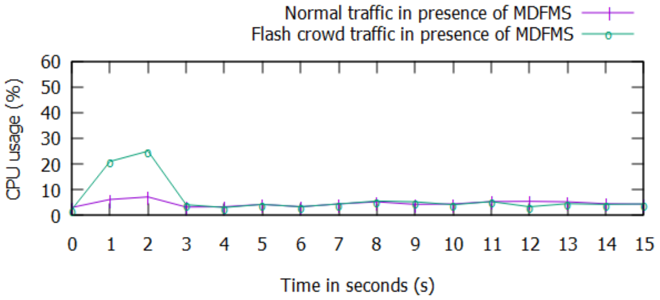


Fig. 8. b) MDFMS computing load on SDN controller

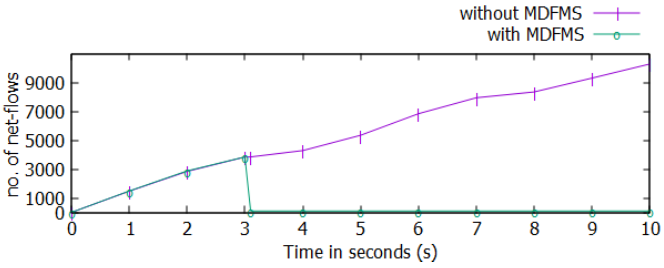


Fig. 9. Detection of traditional DDoS traffic by MDFMS

4.3.4 HTTP Response Time: In this scenario, we have attempted to find the effect of T-DDoS and MMM-DDoS attack in presence of MDFMS by the genuine hosts trying to establish http connection to the web server. 10 hosts <h1-h10> were configured to launch FCGT, 25 hosts <h11-h35> were configured to launch T-DDoS and rest 25 hosts <h36-h60> were configured to launch MMM-DDoS traffic. It has been shown in Fig. 11. that MDFMS could detect and mitigate DDoS traffic in the iteration of 3 seconds and bring the SDN functionalities in healthy state. It is observed that after malicious traffic blocked, FCGT hosts could established http connection in an average time of 0.052 seconds which is almost near to no-attack condition http establish time.

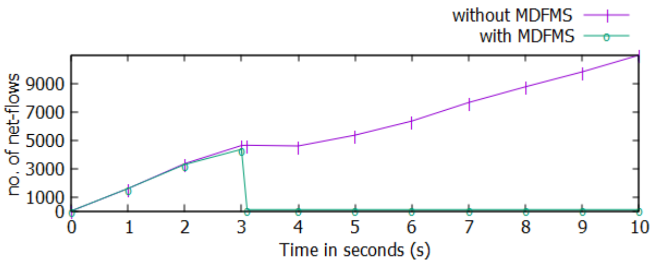


Fig. 10. Detection of MMM-DDoS traffic by MDFMS

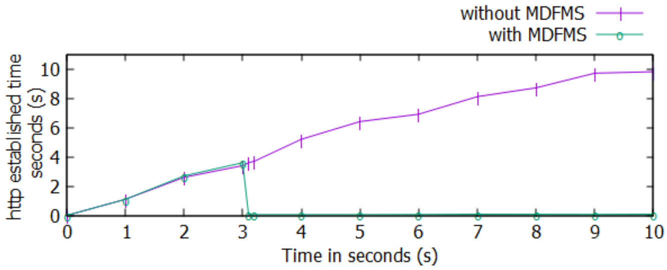


Fig. 11. HTTP established time experienced by flash crowd hosts in presence DDoS traffic

5 Discussion and comparison

There are various mechanisms are already in existence to detect and mitigate T-DDoS attacks. In this paper MMM-DDoS a unique type of attack has been considered. In SDN environment, attackers are interested in SDN controller. To detect and mitigate MMM-DDoS attack hosted by compromised hosts using spoof source IPs are very much similar to normal traffic (Please refer section 3.3). MDFMS get activated after 50% because of two reasons i) to avoid addition computing by MDFMS and ii) during peak hours CPU usage may reach to 50% CPU. 50% of CPU is derived by performing series of experiments. MDFMS works on association of SIP with SMA and able to detect and block precisely compromised hosts which helps in achieving no false positive. False positive is normally the case in machine learning and entropy based solutions to combat DDoS attacks in SDN environment. MDFMS has been compared with OPERETTA, SAFETY and Thresholding Algorithm models. Various performance parameters are shown in Table 2.

Table 2. Comparison of MDFMS with models available in the literature

Features	[10]	[13]	[14]	MDFMS
Detect and mitigate DDoS (TCP, UDP, ICMP)	Support multi targets but for TCP connections only.	Support only single target for TCP connections.	-Doesn't locate nor mitigate the attack flows -Doesn't support MMM-DDoS	-Support MMM-DDoS (TCP, UDP, ICMP) traffic -Locate and mitigate the source of attack flows.
Mechanism Used	Verify TCP-three way connections at controller plane	Count flow of SYN, ACK, RST packet and calculate entropy for the DIP	-Dynamic threshold algorithm -Entropy on destination IP	Analyse the association of SMA with SIP at controller plane
False positive	No	Possible	Possible	No
Additional CPU usage on SDN controller	Yes (model is implemented at controller)	Yes (model is implemented at controller)	Yes (model is implemented at controller)	No (Avg. 0.12% on MDFMS machine)
Attack detection time	Avg 1.5 seconds	Avg 0.8 seconds	Avg. 3 seconds	Avg. 3.210 seconds

6 Conclusion

In this paper, a unique type of attack MMM-DDoS on SDN based networking has been discovered. MMM-DDoS is almost similar to FCGT, MDFMS could successfully differentiate FCGT from malicious traffic (MMM-DDoS + T-DDoS). Association of SIP and SMA is the basic key of proposed model MDFMS. Proposed model has been evaluated under different scenarios and found a light weight solution to detect, locate and mitigate malicious traffic. On an average it takes 3 seconds to complete the differentiation process. In presence of attack with MDFMS, http response time experienced by genuine users are found almost similar to no-attack condition i.e. 0.052 seconds. This got possible because MFGMS does not overload SDN controller and perform its functionality based on simple logic. MDFMS get activated only when CPU usage of SDN are found above 50%. Performance and detection capability of MDFMS has been compared with OPERETTA [10], SAFETY [11] and Thresholding Algorithm [14] (please refer Table II).

Downside of MDFMS is, it blocks the inPORT on detection malicious traffic which consequently blocks user's benign traffic also. Towards the futuristic work in MDFMS, threshold of CPU usage which is 50% in the proposed work should be made dynamic by incorporating Artificial/Machine learning intelligent mechanisms. We would also like to deploy MDFMS in real hardware and analyze its functionality in real network.

References

[1] Latif, Z., Sharif, K., Li, F., Karim, M. M., Biswas, S., Wang Y. :A comprehensive survey of interface protocols for software

defined networks. Elsevier Journal of Network and Computer Applications vol. 156 (2020)

[2] Jafarian, T., Masdari, M., Ghaffari, A., Majidzadeh, K. : A survey and classification of the security anomaly detection mechanisms in software defined networks. Cluster Computing Journal of Networks, Software Tools and Applications vol. 24, pp. 1235–1253, (2021)

[3] Velliangiri, S., Premalatha, J. : Intrusion detection of distributed denial of service attack in cloud. Cluster Computing vol. 22(5), 10615–10623 (2019)

[4] Badotra, S., Panda, S.N. : SNORT based early DDoS detection system using Opendaylight and open networking operating system in software defined networking. Cluster Computing (2019)

[5] Sun, G., Jiang, W., Gu, Y., Ren, D., Li, H. : DDoS Attacks and Flash Event Detection Based on Flow Characteristics in SDN. IEEE Conference on Advanced Video and Signal Based Surveillance, New Zealand (2018)

[6] Zhou, L., Liao, M., Yuan, C., Zhang, H. : Low-rate DDoS attack detection using expectation of packet size. Secure Communication Network (2017).

[7] Zhijun W., Qing X., Jingjie W., Meng Y., Liang L. : Low-rate DDoS attack detection based on factorization machine in software defined network. IEEE Access vol. 8 pp. 17404-17418 (2020)

[8] Hoque, N., Bhattacharyya, D., Kalita, J. :A novel measure for low-rate and high-rate DDoS attack detection using multivariate data analysis. In: COMSNETS, pp. 1–2, India (2016)

[9] Cheng H., Liu J., Xu T., Ren B., Mao J. :Machine learning based low-rate DDoS attack detection for SDN enabled IoT networks. International Journal of Sensor Networks vol. (34)1 (2020)

[10] Fichera, S., Galluccio, L., Grancagnolo, S., Morabito, G., Palazzo, S. :OPERETTA: An OPEnflow-based Remedy to mitigate TCP SYN FLOOD attacks against Web servers. Elsevier Computer Networks, vol. 92, no. 1, pp. 89–100, (2015).

[11] Mohammadi, R., Javidan, R., Conti, M. :SLICOTS: An SDN-Based Lightweight Countermeasure for TCP SYN Flooding Attacks. IEEE Transaction, vol. 14, No. 2, pp. 487 – 497, (2017).

[12] Paxson, Allman, M., Chu, J., Sargent, M. :Computing TCPs Retransmission Timer. IETF, Fremont, tech. rep. RFC 6298, CA USA (2011).

[13] Kumar, P., Tripathi, M., Nehra, A., Conti, M., Lal, C. :SAFETY: Early Detection and Mitigation of TCPSYN Flood Utilizing Entropy in SDN. IEEE Transaction, vol. 15, No. 4, pp. 1545 – 1559, (2018).

[14] David, J., Thomas, :Discriminating flash crowds from DDoS attacks using efficient thresholding algorithm. Elsevier Journal of Parallel and Distributed Computing vol. 152 pp. 79–87 (2021)

[15] Behal, S., Kumar, K., Sachdeva, M. :Characterizing DDoS attacks and flash events: Review, research gaps and future directions. Comp. Sci. Rev. 25 pp. 101–114 (2017)

[16] Behal, S., Kumar, K., Sachdeva, M. :Discriminating flash events from DDoS attacks: A comprehensive review. International Journal Network Security vol. 19(5), pp. 734–741 (2017).

[17] Gera J., Battula B. P. :Detection of spoofed and non-spoofed DDoS attacks and discriminating them from flash crowds. Springer Journal on Information Security, (2018)

[18] Open Networking Foundation, Homepage, http://www.opennetworking.org, last access 2021/02/22

[19] Wazirali R., Ahmad R., Alhiyari S. :SDN OpenFlow Topology Discovery - An Overview of Performance Issues. MDPI Special Issue Next Generation Inter-Domain Policy Routing (2021)

[20] ONF White Paper Software Defined Networking, Homepage, https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf, last access 2021/04/11

[21] OpenFlow Switch SpecificationVersion1.1.0, Homepage, http://archive.openflow.org, last access 2021/06/25

[22] Mininet, Homepage, http://www.mininet.org, last access 2021/05/10

***