IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

Detection and Mitigation of (D)DoS Attacks in SDN Environment Using Entropy

[1]Akanksha Shah, [2] Riddhi Ghate, [3] Sakshi Kalekar, [4] Ruchi Nitsure, [5]Jibi Abraham, [6]Ashwini Matange
Department of Computer Engineering[1,2,3,4,5,6]
College of Engineering, Pune, India
[1]shahas18.comp@coep.ac.in, [2]ghaterp18.comp@coep.ac.in, [3]kalekarss18.comp@coep.ac.in,
[4]nitsurerm18.comp@coep.ac.in, [5]ja.comp@coep.ac.in, [6]asm.comp@coep.ac.in

Abstract: *Software Defined Networking (SDN) is a paradigm for the networks, where the control planes and data planes are separated. It provides centralized network control by separating the network's control logic from the underlying hardware devices. However, like traditional networks SDN is also susceptible to Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. This paper aims to detect and mitigate DoS and DDoS attacks in an SDN environment using an entropy-based approach. The proposed mechanism calculates the entropy of the network over the collected traffic, and derives a dynamic threshold according to the network traffic conditions to determine whether the environment is subject to DoS or DDoS attacks. In the event of the attack, the proposed mechanism installs a drop flow rule into underlying forwarding devices, discarding the traffic sent from attacking host to victim host.*

Keywords: Software Defined Networking, Denial of Service, Entropy, Dynamic Threshold, POX

I. INTRODUCTION

The traditional networks consist of networking devices with tightly coupled control plane and data plane. Control plane constitutes the network's brain and the data plane is responsible for routing packets to their destination. These networks are widely used and are popular but have several drawbacks. Software defined networking (SDN) [10] decouples the control plane and data plane and provides a centralized view of the distributed network for more efficient automation of network services, improved network resource usage, simplified management, lowered operating costs and increased flexibility. The SDN controller provides the communication between the separated planes. The control layer, which can be centralized or distributed, manages network states globally via network policies. OpenFlow [3] is one of the protocols used to establish communication between separated data plane and control plane in an SDN environment. Fig. 1. depicts the SDN architecture.

Despite the advantages that SDN brings to network management and flexibility, lack of security solutions, standardization, and low level maturity of SDN prevents organizations and enterprises from adopting it. SDN is vulnerable to many security threats, however it is most vulnerable to the Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. A DDoS attack is a distributed and coordinated attack that originates from multiple network sources. Fundamentally, the strategy of this attack is to send a sheer volume of spoofed IP packets from different sources in order to make the network resources unavailable to legitimate users.

Over recent years, the attackers have gotten smarter and have been continuously improving and using advanced DDoS attack techniques to inflict more economical and financial damages. According to Arbor's security survey report, the DDoS attacks have increased exponentially in size in recent years and have been causing a tremendous amount of damage to the enterprise networks and data centers. Latin America, along with Asia Pacific, regions saw 14% a rise in DDoS attack frequency. There were 16794 attacks per day, 700 attacks per hour.[16] . These attacks are capable of causing further significant damage to the largest enterprises, cyber–physical systems, data centers, fog computing and service providers where the SDN technology has just begun to sprout and take its shape.

The recent research in defense mechanisms against the DDoS attacks is mainly divided into four different categories depending upon the type of metric and detection mechanism used, namely: Entropy based DDoS defense solutions, Machine learning based, Artificial Neural Networks based and other defense solutions such as statistical based

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

approach, blockchains, etc. This work focuses on the detection and mitigation of DoS and DDoS attacks on hosts in an SDN environment using an Entropy-based approach.

The current gaps concerning the entropy-based detection mechanism includes high dependability of detection mechanism on the static threshold making the detection mechanism quite predictable to certain network traffic simulation, increased congestion between control plane and data plane due to usage of OpenFlow protocol to collect traffic statistics, inability to detect simultaneous attacks on single or multiple hosts and in addition to this, decreased detection accuracy rate due to false positives observed.

The proposed methodology describes the design of detection and mitigation of DDoS attacks. To effectively reduce the congestion between the control plane and data plane, sFlow-RT is incorporated with SDN architecture, emphasis has been given on use of dynamic threshold in entropy-based detection, increasing detection accuracy rate and identifying the victim and attacker hosts to mitigate the attack.

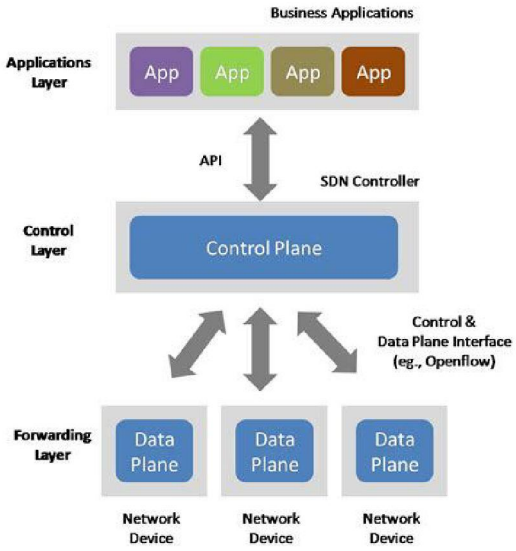

Fig. 1. Software Defined Networking (SDN) Architecture

The rest of the paper is organized as follows, Section II presents the literature, Section III describes the proposed DDoS detection and mitigation solution, Section IV contains the experimental setup, Section V contains the results and discussion and Section VI concludes the research.

II. RELATED WORK

A two stage mechanism for detection and mitigation against DDoS attacks, using Shannon Entropy and static threshold is proposed in [7] and mitigation technique used involves installation of flow rules in switches in order to block the attack traffic at source. This mechanism assumes the attacker sends spoofed ip addresses towards the victim. The processing delay in this method is undesirable as it analyzes only the packet-in messages received in a window size of 50 packets.

Nada M AbdelAzim, Sherif F Fahmy, Mohammed Ali Sobh, and Ayman M BahaaEldin in [1] have proposed a two model technique. First model calculates entropy over source and destination IP. Second model uses KL-divergence to find the change in the normal distribution of network traffic. Both the models are used to detect the DoS attack by comparing it against a pre-computed static threshold. It assumes all nodes in a network are equally likely to communicate with each other for uniform distribution of the traffic within the network.

[8] presents a method to detect and mitigate particularly against TCP SYN flooding attacks in SDN networks .The proposed mechanism clubs the destination IP address and TCP flags to calculate entropy and compares it against an adaptive threshold to detect attacks, and in addition it also has proposed port based mitigation technique to prevent the attack.

Fast Entropy Approach for Detection [5] proposed detection of DDoS attacks based on fast entropy method using flow based analysis. Detection method is based on flow aggregation for doing flow based attack detection and it uses

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

Adaptive Threshold Algorithm to improve detection accuracy. Fast Entropy calculation significantly reduces computational time compared to conventional entropy computation. The paper only provides the detection mechanism, does not discuss a way to traceback attackers and mitigate the attack and even can not differentiate between the attack traffic and flash crowd.

Guo-Chih Hong, Chung-Nan Lee, and Ming-Feng Lee in [6] introduced a DDoS detection mechanism which used adaptive thresholds of entropy and traffic to continuously analyze and evaluate the communication of the whole SDN environment and provide load balancing of the traffic. It calculates the dynamic threshold from a traffic time window and then compares it with the current collected transmission traffic. The mechanism makes use of the flow entry of OpenFlow to deal with DDoS attacks.

III. PROPOSED METHODOLOGY

The proposed methodology describes the design of detection and mitigation of (D)DoS attacks. The emphasis has been given on the use of dynamic threshold in entropy-based detection, increasing the detection accuracy rate and identifying the victim and attacker hosts to mitigate the attack.

3.1 Architecture of Proposed Mechanism

The traffic analysis tool used in this study is sFlow. It uses the functions provided by POX [13] and sFlow to obtain the network topology information and traffic status of the network. The (D)DoS detection and defense mechanism communicates with sFlow and POX through REST API.

The architecture of the proposed system contains three modules as shown in Fig. 2.

1. Flow Statistics Collection.
2. Attack Detection.
3. Attack Mitigation.

These modules are deployed in the application layer of the SDN architecture.

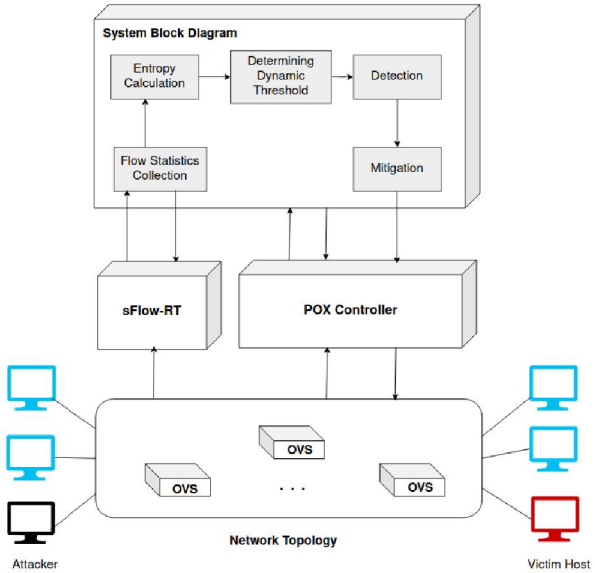

Fig. 2. System Architecture

3.2 Flow Statistics Collection using sFlow-RT

The traffic flows can be collected using sFlow-RT [9]. It samples the packets from the switches that have sFlow agents embedded in them and makes it accessible using REST API to the flow collection module. Flows are collected in a window of 100 packets. Window can be considered as time period, but in this case, window is based on the number of packets. The statistics of these packets are collected in the form of actionable metrics such as Source IP, Destination IP,

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

Source MAC address, Destination MAC address, Source Port, Destination Port and Protocol used. Fig. 3. Depicts the algorithm for flow statistics collection.

3.3 Entropy and Dynamic Threshold Calculation

3.3.1 Relation between Entropy, Threshold And Attack Detection

In the case of normal traffic scenarios, each host is equally likely to communicate with every other host, as a result of which the randomness in the distribution of packets in the network increases. Entropy metrics such as Shannon entropy is used to measure randomness in the behavior of the network. In case of attack traffic scenarios, the concentration of packets at the victim host with source ip as the attacker host increases, as a result of which the randomness in the distribution of the packets in the entire network decreases to a value less than it would have been in case of normal traffic. The proposed attack detection mechanism uses entropy as a measure of randomness and dynamic threshold, which modulates itself according to changing network conditions, to determine the presence of the attack.

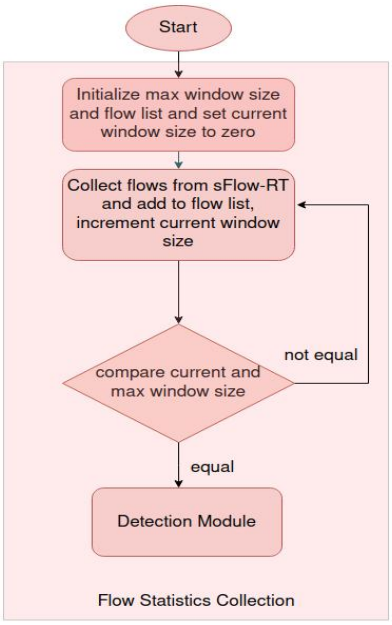

Fig. 3. Flow Statistics Collection

The detection module deployed on the POX controller detects the attack in two steps:

3.3.2. Determining Dynamic Threshold

The description of the notations used in the mechanism are shown in Table I.

TABLE I: Table of Notation

Notation	Description
μ	Average value of traffic flow count
σ	Standard deviation of traffic flow
n	Total number of destination IP addresses captured
i	The i-th destination
c_i	Number of packets destined to i^{th} destination IP address captured in the window
λ	Traffic parameter, an integer constant

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

d_i	The *i-th* destination IP address
$H(d_i)$	Entropy at destination d_i
j	The j-th traffic source
m	Total number of sources contributing traffic to d_i
P_{ji}	Probability of the traffic coming from j^{th} source to d_i
c_{ji}	Number of packets from source j to destination i
H_w	Window Entropy
K	The number of entropy rounds

The threshold used here is adaptive to network traffic and is not static. The dynamic threshold value is calculated over the window of 100 packets using the average value of traffic flow count μ and the traffic standard deviation σ. Parameters σ and μ are calculated according to 1 and 2 respectively.

$$\mu = \frac{\sum_{i=1}^{n} c_i}{n} \quad (1)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (c_i)^2}{n} - \mu^2} \quad (2)$$

Where,

c_i = Number of packets destined to the i^{th} destination IP address captured in the window.

n = Total number of unique destination IP addresses captured in the window.

Threshold is calculated as Equation 3 where λ is the factor that determines how far from the mean the threshold is.

$$Threshold = \mu + \lambda * \sigma \quad (3)$$

Where,

λ = Traffic parameter, an integer constant.

The proposed mechanism uses experimentally determined values of $\lambda = 3$ as determined in Section 4.

3.3.3. Calculation of Entropy

Entropy is used to measure the randomness in the behavior of the network. For each window of 100 packets, entropy is calculated in two steps :

a. Calculating the Destination Entropy

For every destination captured in the window, Shannon Entropy is calculated as Equation 4.

$$H(d_i) = -\sum_{j=1}^{m} P_{ji} \log_2 P_{ji} \quad (4)$$

Where,

$d_i = i^{th}$ destination captured in the window

$H(d_i)$ = Entropy at destination d_i

m = Total number of sources contributing traffic to d_i

P_{ji} = Probability of the traffic coming from j^{th} source to d_i.

P_{ji} is defined as,

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

$$P_{ji} = \frac{c_{ji}}{c_i} \qquad (5)$$

Where,
c_{ji} = Number of packets from source j to destination i.
c_i = Total number of packets received by destination d_i from all its sources.

b. Calculating the Window Entropy
After calculating the entropy for each destination in the window, window entropy is calculated as follows

$$H_w = \sum_{i=1}^{n} H(d_i) \qquad (6)$$

Where, H_w = Window Entropy.

3.4 Attack Detection

To detect the attack, the computed dynamic threshold for a window is compared with the entropy calculated for the window. If the value of entropy is greater than the threshold computed, no attack is detected. But, if the value of entropy drops down to a value less than that of the threshold, there is a possibility of an attack. During this, there is a possibility of observing false positives because of the very nature of the dynamic threshold trying to adapt itself to the network conditions. Moreover, there is a chance of seeing false positives due to the delayed collection of logged flows from sFlow-RT. Therefore, in order to confirm the presence of an attack, we look for three consecutive windows to show the attack behavior. These consecutive windows are referred to here as the number of entropy rounds (K). This implementation uses the value of $K = 3$. Fig. 4. demonstrates the attack detection algorithm. If an attack is detected, the system enters the mitigation phase to cease the attack.

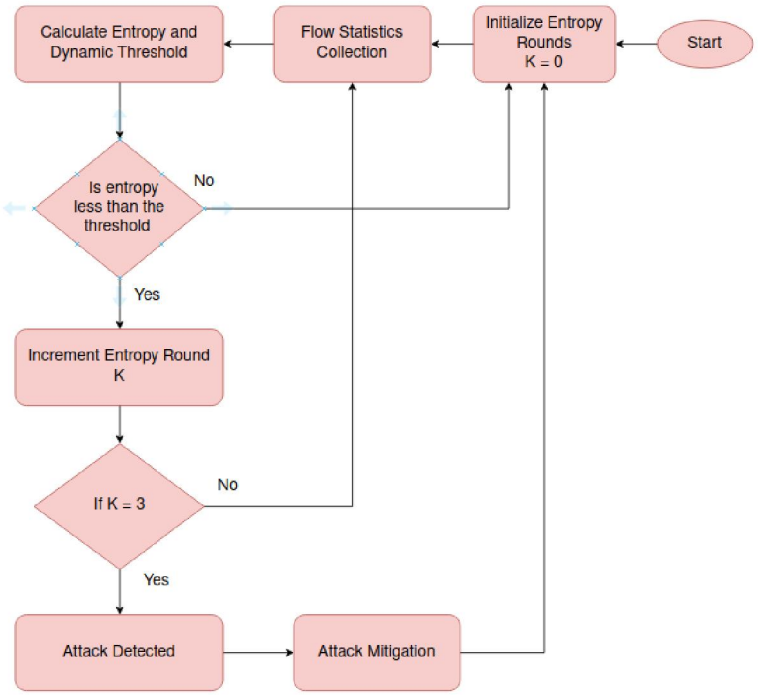

Fig. 4. Attack Detection

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

3.5 Attack Mitigation

The attack mitigation phase blocks the traffic flows from the attacker. The best way to mitigate a (D)DoS attack is by identifying the attacker and then blocking the packets generated from the attack source. The mitigation phase mitigates the attack using a two-stage method.

3.5.1. Identification of victim and attacker host in case of DDoS attack

The mitigation phase identifies the victim and attacker host by analyzing the header information such as Source IP, Destination IP, Source MAC, etc from the packets collected in a set of three windows for which attack is detected. The required header information is stored in three map-like data structures. Fig. 5. demonstrates the data structure used.

Each pair in Map-I stores the set of source IP addresses with the same MAC address. Map-II is used to store the destination IP addresses and the collective number of packets received from multiple sources. The key field in Map-III stores the tuple (source IP address, destination IP address). The value field stores the count of packets from the corresponding source IP address to the destination IP address that is stored in the key. Map-II is used to identify the victim host in case of DoS and DDoS attacks. The destination IP address with maximum frequency in Map-II is detected as the victim host. To identify the attacker host in case of DoS attack, Map-III is used. Source IP, which contributes the maximum number of packets to the identified victim host, is identified as the attacker. In case of DDoS attack, Map-I is used. Size of the value field greater than one indicates that the source generates the packets using spoofed IP addresses, as more than one IP address is being mapped to the same MAC address. The hosts corresponding to such MAC addresses where invalid mapping is detected are identified as attacking hosts.

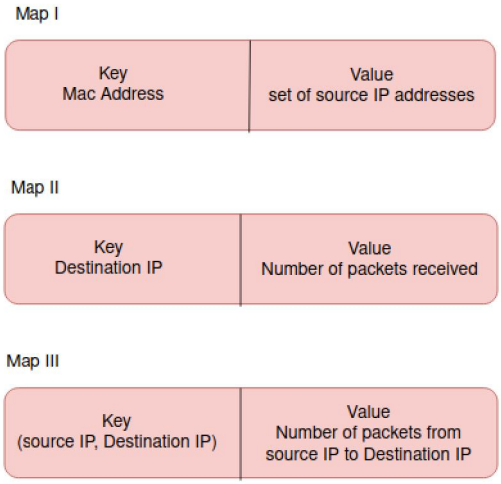

Fig. 5. Data Structures : Map-I, Map-II, and Map-III

3.5.2. Mitigation of attack

Once the victim host and attacking host or hosts are identified, the next step is to block the packets destined from the attackers to the victim host. The SDN controller during the mitigation phase issues an OpenFlow FLOW_MOD message to the switches that establishes the corresponding ”drop” flow table entries with an idle timeout of 60 seconds. The FLOW_MOD message includes parameters such as IP address and MAC address of identified attacker and victim host, protocol used, source port, destination port, etc. which drops the packets from the attacker to the victim. In case of a DDoS attack, multiple sources spoof the IP addresses, in this case, MAC addresses identified as attackers from Map-I are used as key parameters to block the attack.

IV. EXPERIMENTAL SETUP

4.1 Network Topology

For conducting the experiment, Linux Ubuntu 20.04 was used along with Mininet [15] as the network emulator and python based POX as the SDN controller. The proposed methodology was simulated using 17 hosts and 6 OpenFlow enabled switches as shown in Fig.6..

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

In the Fig. 6, s1, s2,..., s5, s6 are OpenFlow switches, each of which is sFlow enabled to allow flow statistics collection and is connected to the centralized POX controller (c0) and h1, h2, h3,..., h16, h17 are the hosts used.

4.2 Network Traffic Simulation

Traffic generation was done using Scapy [14]. It is a powerful python based tool for manipulation of packets. It was used to simulate UDP packets for both normal and attack traffic through the SDN network.

4.2.1 Normal Traffic Generation

In this case, each host sends UDP packets with source port as 2 and destination port as 80 to all other hosts with the same probability to avoid any certainty in the network traffic. Each chosen source generates a random number of packets. These sets of transmissions were deferred by an idle time of 5-10 seconds, and each packet was transmitted at the rate of 0.5 packets/sec.

4.2.2 Attack Traffic Generation

In case of DDoS attacks, Scapy was used to generate packets with spoofed IP addresses. For attack traffic generation, the packets were generated with source port as 80 and destination port as 1. These packets were specifically directed towards the chosen victim host with a rate of 100 packets/second.

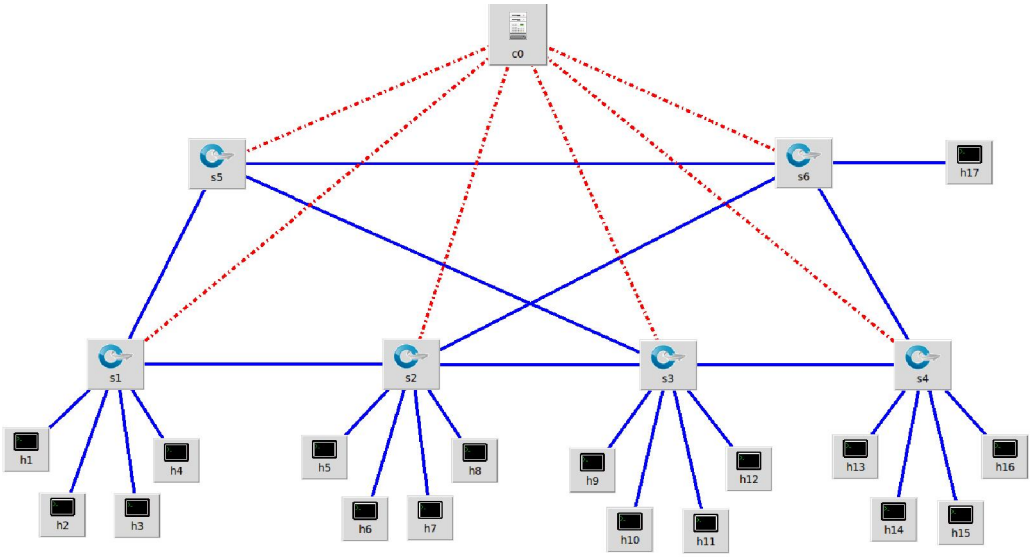

Fig. 6. Network Topology

4.3 Flow Collection

sFlow-RT, an analytics engine that provides real-time visibility in Software Defined Networking is used for the purpose of flow collection. It receives the traffic flow from the switches that have sFlow agents embedded in them [12]. The default sFlow port used is UDP 6343. The sFlow-RT engine converts the collected traffic statistics into actionable metrics that are accessible through RESTflow API. The RESTflow API used for the experiment is "/flows/json" that gets the most recently logged flows along with the parameters namely name, flowID, maxFlows and timeout. The flowkeys used for logging the flows are : Source IP address, Destination IP address, Source MAC address, Destination MAC address, Source Port, Destination Port. Fig.7. displays the logged traffic flows using sFlow-RT.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

```
{
"flowID" : 127
"sourceIP" : "10.0.0.1"
"destinationIP" : "10.0.0.11"
"sourceMAC" :  "0EBEB58F103A"
"destinationMAC" : "6A5A0366EBAE"
"udpsourceport" : "2"
"udpdestinationport" : "80"
"agent" : "127.0.0.1"
"value" : 10
"dataSource" : "99"
},
{
"flowID" : 128
"sourceIP" : "10.0.0.5"
"destinationIP" : "10.0.0.12"
"sourceMAC" :  "EA090E1411E9"
"destinationMAC" : "CA8703BEB77D"
"udpsourceport" : "2"
"udpdestinationport" : "80"
"agent" : "127.0.0.1"
"value" : 20
"dataSource" : "101"
}
```


Fig. 7. Flow Definition

4.4 Experimental Parameters

All the parameters used during the simulation are listed in Table II.

TABLE II: Experimental Parameters

Experimental Parameter	**Value**
SDN Controller	POX
Number of OpenFlow Switches	06
Number of Hosts	17
Number of Victim Hosts	01
Window Size	100 packets
Normal Traffic Rate	0.5 packets/second
Attack Traffic Rate	100 packets/second
Number of Entropy Rounds (K)	03
OpenFlow Protocol Version	1.0
sFlow Sampling Rate	10
sFlow Polling Rate	10

V. RESULTS AND DISCUSSION

5.1. Window Size Analysis

Choosing the right size of window is necessary for the proposed detection algorithm to get accurate results. A window size of 100 packets was chosen for implementation. Table III. demonstrates the computation carried out for several window sizes. It shows the difference in the entropy of H_N and H_A where H_N is the average normal traffic entropy, H_A is

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

average attack traffic entropy and H_N - H_A is the difference between the respective entropies. Window size of 150 offered a greater difference, but it takes a lot of time in computing the window entropy for 150 packets as compared to that of a window of 100 packets. Window sizes of 75 and 100 look close but, the difference obtained between normal and attack traffic entropy was greater in case of window size of 100 as compared to window size of 75.

Table III: Comparison of Window Sizes

Window Size	H_N	H_A	$H_N - H_A$
50	8.3933	7.67862	0.7147
75	12.7336	11.7009	1.0327
100	17.9933	16.6945	1.2988
150	23.7707	20.0405	3.7302

5.2 Determining λ

Fig.8. demonstrates variation between the threshold calculated with different values of λ and calculated entropy over a series of 20 windows. Each window is a collection of 100 packets. During the analysis, normal traffic was stimulated from window 1 to 10 and attack traffic was introduced into the network at 11th window. The appropriate value of threshold is the one, which in case of normal scenarios is less than the entropy calculated and greater than the entropy calculated in case of attack scenarios. It can be observed from Figure. 9 that the threshold calculated using $\lambda = 3$ was less than the entropy calculated for windows 1 to 10 and greater than the entropy calculated for the rest of the windows. $\lambda = 2$ and $\lambda = 4$ seemingly were the right choices respectively for normal and attack traffic scenarios, but these values of λ did not satisfy the required variation between the threshold and entropy calculated for all traffic scenarios. Therefore, the least possible integer value of $\lambda = 3$ which satisfied the required variation for all windows was chosen for implementation.

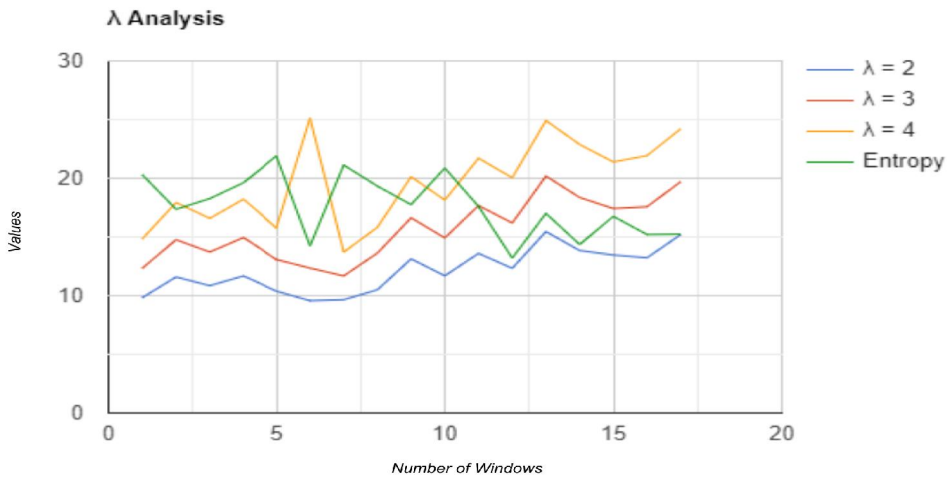

Fig. 8. Determining λ

5.3. Variation of Entropy with Dynamic Threshold

Fig. 9. depicts the graphical variation of entropy and dynamic threshold calculated for 15 consecutive windows. In accordance with the detection mechanism proposed it is expected that in case when there is an attack, the value of entropy calculated should be less than the dynamic threshold calculated over the window of 100 packets and vice versa in case of normal traffic conditions.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

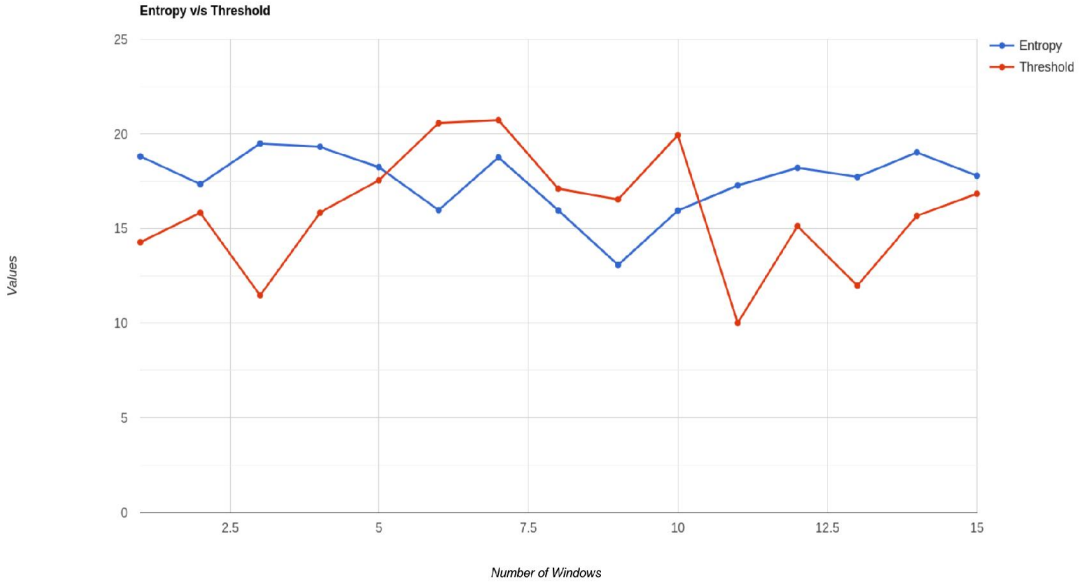

Fig. 9. Entropy v/s Dynamic Threshold

In the above experiment, normal traffic was simulated from window 1-5, during which the network entropy (6) was greater than the dynamic threshold (3) calculated. Attack traffic was simulated in the network for window 5-10, during which the observed value of entropy was less than the value of dynamic threshold calculated for each window. Once the attack was detected and mitigated by the proposed mechanism, the observed value of entropy was again greater than the dynamic threshold calculated for each window from window 10-15. This observation attests to the concept of the variation of entropy with threshold in case of changing network conditions.

5.4. Detection Accuracy

The detection accuracy describes the correctness of the results given by the detection module after analyzing the network traffic. Choosing the correct value of *K* (number of entropy rounds) influences the detection accuracy rate greatly. During the experiment, the detection module analyzed the network traffic flow for consecutive 24 rounds.

The detection accuracy is calculated as

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

Where,

TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

Confusion matrix obtained for $K = 1$ is shown in Table 4 and for $K = 3$ is shown in Table 5.

Table IV: Confusion Matrix for $K = 1$

	Attack Detection		
Attack Simulation	Positive	Negative	Total
Positive	TP = 9	FN = 0	9
Negative	FP = 3	TN = 12	15

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

Total	12	12	24

*Accuracy = ((12 + 9) / 24) * 100 = 87.5%*

Table V: Confusion Matrix for *K = 3*

	Attack Detection		
Attack Simulation	Positive	Negative	Total
Positive	TP = 3	FN = 0	3
Negative	FP = 0	TN = 15	15
Total	3	15	18

*Accuracy = ((3 + 15) / 18) * 100 = 100%*

Comparing the results in Table 3 and Table 4 and the results of further rounds, it was observed that the minimum percentage increase in accuracy rate is 12% when the value of *K* is increased from *K = 1* to *K = 3*. For *K = 1*, the overall accuracy of the system was dependent on the number of false positives observed. As the number of false positives decreased, the accuracy increased. Using *K = 3* to detect the presence of attack instead of *K = 1* eliminated the unnecessary false positives observed, and thus the improved detection accuracy.

False positives were observed when the normal traffic was simulated through the network. Longer the time normal traffic flowed through the network, greater was the possibility of observing false positives in case of *K = 1*. This caused the accuracy of the detection mechanism with *K = 1* to decrease. As a result of which, when the detection algorithm was used with *K = 3*, greater percentage increase in accuracy was obtained.

5.5 Effect of Mitigation

During the experiment, the attack traffic was initiated from host h16 to h17 and hosts h1, h5, h9 and h13 stimulated normal traffic through the network. Attacking host (h16) is connected to switch S4 as shown in Fig. 6. In case of attack scenarios, before the mitigation phase is reached, the attacking packets, i.e. the packets that are initiated from attacker host to victim host, float across the network, increasing the traffic load. In the case of POX controllers, routing is done by constructing the spanning tree based on network topology and depending upon the current network conditions. As the network condition changes, the spanning tree also changes. As a result of which the path taken by the attacking packets from source to destination changes, making it difficult to keep the track of the switches lying within its path. Therefore, after detection of attack, during the mitigation phase, the drop flow rule entries are installed on every switch within the network, with an idle timeout of 60 seconds.

Fig. 10. Drop flow rule installation in non attacking switch (S1)

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

Fig.10. and Fig. 11. Demonstrates the drop flow rule installed in switches. This flow rule installation causes the floating attacking packets across the network to be dropped. A stage is reached when no packets match the installed flow rule entry into the switches and the idle timeout expires, causing the flow rule to be flushed out of the switch's memory.

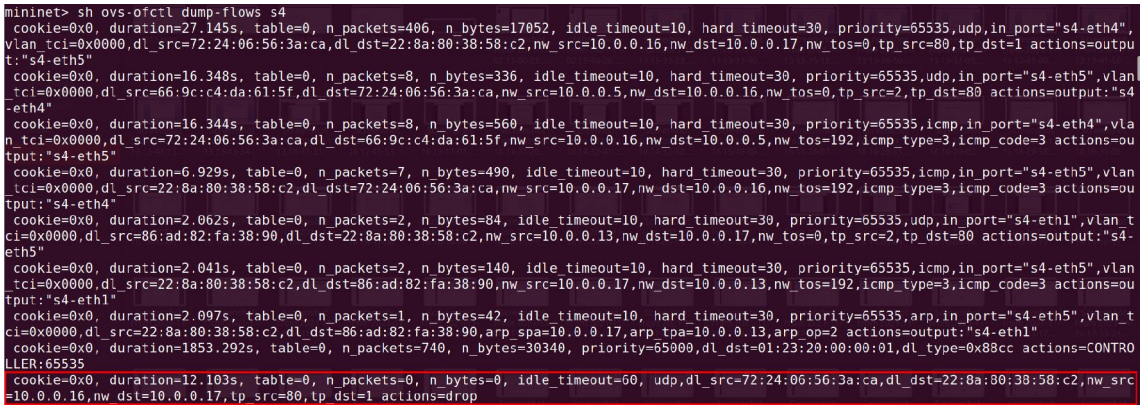

Fig. 11. Drop flow rule installation in attacking switch (S4)

Eventually, only the switch to which the attacking host is connected (S4), has the drop flow rule installed, as the incoming attacking packets match the drop flow rule and refreshes the idle timeout. Therefore as long as the attacker is still trying to send the attacking packets into the network, they are being dropped at the switch to which the attacking host is connected. The drop flow rule prevails until the attacker stops targeting the victim host. As a result, the mitigation phase tries to cease the attack near to source, reducing the congestion and unnecessary traffic load on the network. Fig. 12. And Fig. 13. Demonstrates the flow rule entries in the attacking and non attacking switch after the expiration of Idle timeout.

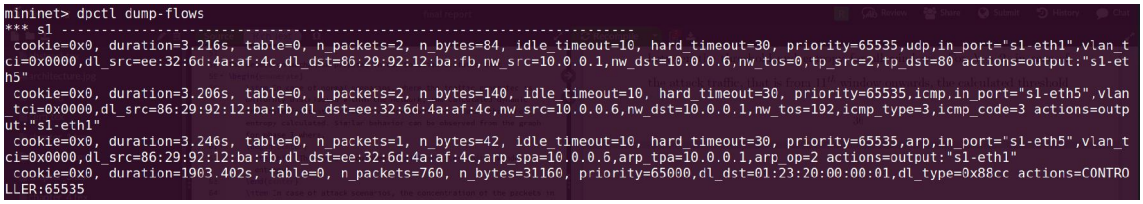

Fig. 12. Removal of drop flow rule from non attacking switch (S1) after Idle timeout

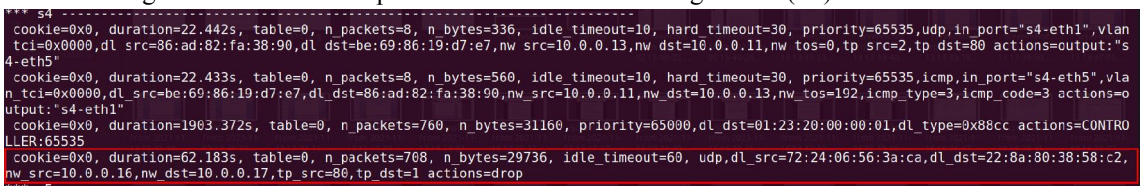

Fig. 13. Prevailing drop flow rule in attacking switch (S4) after Idle timeout

VI. CONCLUSION AND FUTURE WORK

The proposed technique was successfully able to bypass the congestion during flow statistics collection using sFlow, modulate the threshold according to the network conditions, detect the presence of an attack on a single host and reduce the number of false positives observed by using three rounds of entropy detection ($K = 3$). Using this approach it achieved a minimum percentage increase of 12% in detection accuracy. Mitigation technique successfully traced back the attackers and blocked the attack near to source. The POX controller proved to be simple and powerful in establishing a test ground for the entropy based detection and mitigation mechanism. The increased computing power of the controller gave an upper hand to the algorithm presented in this paper. The future scope of the project includes the simultaneous detection of attacks on single or multiple hosts. In SDN, several controllers can be connected to each other, detecting and mitigating the attack in such cases would require establishing the inter-controller communication that would send attack alerts to other networks [2]. Adding this communication process to current implementation will be an extension and a topic for future work.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

REFERENCES

[1] Nada M AbdelAzim, Sherif F Fahmy, Mohammed Ali Sobh, and Ayman M BahaaEldin. A hybrid entropy-based dos attacks detection system for software defined networks (sdn): A proposed trust mechanism. Egyptian Informatics Journal, 22(1):85–90, 2021.

[2] ZakariaAbou El Houda, AbdelhakimSenhajiHafid, and LyesKhoukhi. Cochain-sc: An intra-and inter-domain ddos mitigation scheme based on blockchain using sdn and smart contract. IEEE Access, 7:98893–98907, 2019.

[3] Wolfgang Braun and Michael Menth. Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices.Published in Future Internet, 2014.

[4] J Dalou, Basheer Al-Duwairi, and M Al-Jarrah. Adaptive entropy-based detection and mitigation of ddos attacks in software defined networks. International Journal of Computing, 19(3):399–410, 2020.

[5] Jisa David and Ciza Thomas. Ddos attack detection using a fast entropy approach on flow-based network traffic. Procedia Computer Science, 50:30–36, 2015.

[6] Guo-Chih Hong, Chung-Nan Lee, and Ming-Feng Lee. Dynamic threshold for ddos mitigation in sdn environment. In 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pages 1–7. IEEE, 2019.

[7] Yajie Jiang, Xiaoning Zhang, Quan Zhou, and Zijing Cheng. An entropy based ddos defense mechanism in software defined networks. In the International Conference on Communications and Networking in China, pages 169–178. Springer, 2016.

[8] Prashant Kumar, MeenakshiTripathi, Ajay Nehra, Mauro Conti, and ChhaganLal. Safety: Early detection and mitigation of tcpsyn flood utilizing entropy in sdn. IEEE Transactions on Network and Service Management, 15(4):1545–1559, 2018.

[9] BabatundeHafisLawal and A. T. Nuray. Real-time detection and mitigation of distributed denial of service (ddos) attacks in software defined networking (sdn). pages 1–4, 2018.

[10] KashifNisar, Emilia Rosa Jimson, MohdHanafi Ahmad Hijazi, Ian Welch, Rosilah Hassan, AzanaHafizahMohdAman, Ali Hassan Sodhro, Sandeep Pirbhulal, and Sohrab Khan. A survey on the architecture, application, and security of software defined networking: Challenges and open issues. Internet of Things, 12:100289, 2020.

[11] Maninder Pal Singh and Abhinav Bhandari. New-flow based ddos attacks in sdn: Taxonomy, rationales, and research challenges. Computer Communications, 154:509–527, 2020.

[12] Rochak Swami, Mayank Dave, and VirenderRanga. Defending ddos against software defined networks using entropy. In 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), pages 1–5. IEEE, 2019.

[13] POX Installation and Documentation - https://noxrepo.github.io/pox-doc/html/

[14] Scapy network traffic generation and packet manipulation https://scapy.readthedocs.io/en/latest/index.html

[15] Mininet - http://mininet.org/

[16] https://www.netscout.com/report/