2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering

DDoS Flooding Attack Detection Based on Joint-entropy with Multiple Traffic Features

Jiewen Mao*, Weijun Deng and Fuke Shen*
Department of Computer Science and Technology
East China Normal University, China

***Abstract*—Distributed Denial of Service (DDoS) attacks are still considered as severe threats to the Internet. Previous works have used information entropy to detect DDoS flooding attacks. However, these methods usually only used source address as the feature of packets, and ignored other features. Besides, the entropy with single variable also has restricts in abnormal detection. In this paper, we propose a new joint-entropy-based DDoS detection solution with multiple features of packets. We choose flow duration, packet length, source address and destination port as the key features to detect different types of DDoS flooding attacks. We carry out the experiments with simulated campus network based on Software-defined Networking (SDN) architecture. The results show that our proposed method can effectively detect attacks of both forged and non-forged source address, and outperforms the previous single-entropy methods in terms of accuracy and false positive rate.**

***Index Terms*—DDoS, Detection, Joint-entropy, Flow duration, Software-defined Networking**

I. Introduction

Nowadays, Distributed Denial of Service (DDoS) attacks are still considered as severe threat of networks, and have resulted in more and more losses to enterprises and governments in recent years. Because it is much easier to launch a DDoS attack but still hard to defense it. Some of such attack events and their consequences are listed in a survey [1]. In all types of DDoS attacks, flooding attack is still a widely-used approach. Once a flooding attack occurs, the huge flow traffic will soon exhaust the victims. Therefore a real-time detecting method is needed. However in recent years, researchers focus on analysing existing data sets but ignore application in real networking systems.

Information theory-based methods have been applied in recent years. The key attractions of using information theory-based methods can be described as follows: (a) minimized features selection, (b) lower complexity in computation, (c) high scalability and high sensitivity and (d) convenient deployment [2]. We can process multiple packet features simultaneously. However, most information theory-based metrics are limited in traditional networking frameworks, and only one feature of packets are considered. That restricts the advantages of information theory.

In this paper, we proposed a novel multidimensional joint-entropy-based DDoS attacks detection method. It utilized the flow-duration, which was also a newly proposed packet feature, consisting other important features in packet heads to detect DDoS flooding attacks of TCP, UDP and ICMP.

To efficiently deploy the proposed algorithm, we introduce software-defined networking (SDN) architecture [3]. In SDN, a centralized controller manages distributed forwarding devices (SDN switches). This kind of architecture makes network policies and algorithms deployed agilely. Moreover, the global view of network in SDN can help the detection methods to coordinate behavior of switches on demand intelligently. In this paper, our DDoS detection algorithm will be installed as a component of the SDN controller, and the switches only transmit packets according to the flow table generated by the algorithm. That will extremely improve the convenience of deploying.

The main contributions of this paper can be summarized as:

- We propose a novel Joint-Entropy-based algorithm in detecting spoofed and non-spoofed DDoS attacks. The experiment results shows that our method can reach higher accuracy and lower false positive rate as compared to existing single-entropy-based methods in detecting real-time DDoS attacking.
- The proposed algorithm utilized flow duration, which will be proved as an effective feature of recognizing whether a flow is attacking flow or legitimate flow. Besides, the algorithm also utilized traditional source IP address, packet length and destination port as features to compute joint-entropy.
- The proposed detection algorithm is deployed in an SDN-based simulated campus network architecture, and acts as one of the modules installed in the SDN controller.

The remainder of this paper is organized as follows. The related works are described in Section II. In Section III we introduce the definition of information entropy firstly. Then we explain what features we choose to detect different types of attacks and why. Section IV explains our algorithms with joint-entropy for detecting typical DDoS flooding attacks. Section V shows the experiment setup, experiment results and the evaluation of measurements. The discussion about experiment results are shown in Section VI. Finally Section VII concludes our work and presents the future directions.

II. Related Work

Li et al. [4] used two approaches to examine the affect of entropy on DDoS detecting. The first method is cumulative entropy detection and the second is time-based entropy detection.

*Jiewen Mao is the first author of this paper. E-mail: paulexe@163.com. Fuke Shen is the corresponding author. E-mail: fkshen@ecnu.edu.cn

2324-9013/18/31.00 ©2018 IEEE
DOI 10.1109/TrustCom/BigDataSE.2018.00045

These two methods both use entropy of packets in a sliding time window instead of the entire packets. The experimental results showed that the cumulative or time-based entropy can detect the change of average entropy and they can lead to more accurate and effective detection.

A series of surveys [5]–[9] also summarized and introduced previous works. Zargar et al. [5] classified mainstream D-DoS flooding defense mechanisms into two classes: based on deployment location and point in the time that defense takes place. Spatially, the detection methods based on entropy can be deployed near destination. Temporally, these methods can be executed during or after the attack. The literature [6] presented a comprehensive survey of attack detection methods, including some statistical approaches. However, the authors did not discuss the methods based on entropy or SDN clearly. Ahmad et al. [7] described the DoS attack on the control plane in SDN. But it did not elaborate the possible solutions about this aspect. The authors of [8] analyzed the security challenges of SDN-based cloud network, discussed the reason why DDoS attacks grow substantially in cloud, and introduced some available defense mechanisms. This literature explained that SDN has good features in detecting and reacting DDoS attacks in cloud computing. A new survey [9] published in 2017 expounded the DDoS attack detection techniques based on entropy in SDN and introduced some available related works regarding this method. Besides, this work provided other detection approaches including machine learning, traffic pattern analysis, connection rate and integration of snort. Also, this work presents a new proactive DDoS defence framework, but it is only a conception, which has not been verified experimentally.

The authors of [10] used entropy of four individual features of a flow (source address, destination address, source port and destination port) to detect three types of network anomalies which are DDoS attack, worm propagation and port scan. A detection framework with SDN and sFlow [11] was presented. The theory and method of this work is referable for our paper, but according to the experiment results, false positive rate will be relatively high when the detection accuracy is 100%. A high false positive rate will influence the judgement of normal traffic. Lu et al. [12] also exploited SDN environment and sFlow to collect traffic flows. This method defined two-dimension statistical metric named distribution-collaboration degree (DCD) and intensity of a flow to detect DDoS attack and distinguish it from the flash crowd traffic. The key limitation of this research is that it may only adapt to small network, because the computation complexity of its algorithm is $O(M^2N^2)$, with the network scale grows, the computation may be slow and intolerable.

There are also two works [13], [14] discussed about using entropy-based mechanism to detect DDoS attack in SDN environment. They both used single-dimension entropy, and satisfactory results are reached. However, to the author's best knowledge, the affection of multi-dimension entropy was not considered. Another reference [15] also talked about DDoS attack detection under SDN context. It leverage on SDN's flow monitoring capability and balance the coverage and granularity of attack detection.

H.Rahmani et al. [16] presented an approach using joint-entropy to detect DDoS attack. In this literature, the author analyzed the entropy of IP flows and packets, and they applied the models of exponential distribution and gamma distribution to describe the probability of two random variables. This method can effectively distinguish DDoS attack and flash crowd traffic. The theory of joint-entropy is referable for our paper, too. The shortcoming of their study is that they make no explanation of other characteristics of network traffic.

A new paper [17] presented ϕ-Entropy and ϕ-Divergence to detect DDoS attacks and Flash Events (FE). They compared their methods to existing used Generalized Entropy (GE) and Generalized Information Divergence (GID) metrics. They also used entropy difference to detect different types of DDoS attacks and FEs. However, they still only used number of packets, source IP addresses and time window as the features used for detecting. Besides, their deployment was still adapted to traditional networking framework. That is the difference of our proposed method and the work in [17].

III. Information Entropy for Attack Detection

In this section, firstly we will introduce the basic concept of information entropy, then the features to be calculated in joint-entropy will be listed and described.

A. Definition of Information Entropy

The concept of information entropy was presented by Shannon [18] in 1948. The information entropy, or Shannon entropy is defined as

$$H(X) = -\sum_{i=1}^{N} p_i \log_2(p_i) \tag{1}$$

where p_i is the probability of random variable $x_i (i = 1, 2, ..., N)$.

If we consider two random variables simultaneously, the joint-entropy of random event X and Y is defined as

$$H(XY) = -\sum_{i=1}^{N}\sum_{j=1}^{M} p(x_i y_j) \log_2(p(x_i y_j)) \tag{2}$$

where $p(x_i y_j)$ is the probability of event $(X = x_i, Y = y_j)$, $i = 1, 2, ..., N$ and $j = 1, 2, ..., M$.

In network detection, for all captured packets in a certain time window T, we can calculate the probability distribution of every flow feature that we need. By choosing reasonable features and calculate their joint-entropy, we can analyze the status of current traffic, and judge whether an attack exists in this time window. The chosen features and their descriptions are explained in the next subsection.

However, it is meaningless to compare the joint-entropy with the single entropy directly, because they hold two different sample spaces. To make joint-entropy and single-entropy comparable with each other, we employ the normalized entropy to limit the value of entropy in interval $[0, 1]$. The

TABLE I: Features used in joint-entropy algorithm

Attack Type	Forged Source Address		Non-forged Source Address	
	First feature	Second feature	First feature	Second feature
ICMP Flooding	flow duration	packet length	source IP	packet length
UDP Flooding		destination port		destination port
TCP SYN Flooding		packet length		packet length

expressions of the normalized single-entropy and joint-entropy are represented as Equation 3 and Equation 4, respectively.

$$H'(X) = \frac{H(X)}{\log N} \tag{3}$$

$$H'(XY) = \frac{H(XY)}{\log MN} \tag{4}$$

In the following, we will use H to denote normalized joint-entropy instead of H'.

In normal traffic, the packets can be seen as arriving randomly, so the entropy of normal traffic is usually high. Conversely, in a DDoS attack, the distribution of some features in attack packets will trend to centralized, and the entropy of these feature will decrease. If the difference of the normalized joint-entropy between current traffic H_c and the normal traffic H_n exceeds the preset threshold t, a potential DDoS flooding attack may have happened.

B. Features used for detecting

Table I lists all the features we have used in the algorithms. To detect the attacks with the forged source address, we choose flow duration as the first feature, while we choose source IP address to detect the attacks with the non-forged source address. We then choose packet length as the second feature to detect the ICMP flooding and TCP SYN flooding attacks, and destination port for UDP flooding attack, respectively. Both packet length and destination port can be used for detecting attacks with forged and non-forged source address.

A flow can be denoted as a tuple with five elements which are (SrcIP, DstIP, SrcPort, DstPort, protocol). When any of the elements in the tuple model change, the new five tuple can be seen as a new flow. The flow duration (FD) is defined as the difference between the arriving time of the last packet and the first packet of this flow. Let T_{last} and T_{first} be denoted by the two arrival times respectively, the flow duration FD can be written in the form $FD = T_{last} - T_{fisrt}$. In engineering, NetFlow [19] provides the ability to obtain T_{first} and T_{last} of every captured flow automatically as it enters an interface. Each time the packets which conform to a particular flow keep arriving at the detection devices, the flow duration will keep increasing, which is not related to the sampling interval. According to our observation, when the target host encounters the forged source address DDoS attacks, the flow duration of these attack flows with illegitimate source IP addresses is shorter comparing to the durations of normal flows, because the probability that the attack packets are generated with the same flow features is rather low. In the view of our detection system, a large amount of flows with one packet will emerge fast. The last and the first packets of these flows are the same packets, which lead to a zero flow duration. Thus the probability distribution of flow duration will become concentrated and the calculated entropy value will decrease significantly. Hence we can use flow duration to calculate entropy value to detect such DDoS attacks.

If an attacker does not spoof the source IP address to conduct a DDoS attack, the entropy of source IP address will decrease, because the distribution of source IP will concentrate on the addresses of attackers. This kind of attack often occurs in intranet. To put in another way, both the attacker and victim locate in the same local area network. Furthermore, Our proposed method can defense against such an attack.

Considering the diversity of the packets in the normal traffic pattern, the distribution of packet length should be dispersive. Moreover, in normal traffic pattern, ICMP packets only account for a little in all packets. When an ICMP flooding attack or a TCP SYN flooding attack occurs, the packet length will also concentrate to a particular value and results in the entropy value of packet length being decreased remarkably. It is obvious that we can use packet length as a key feature to detect ICMP flooding attack and TCP SYN flooding attack.

To detect UDP flooding attack, we choose destination port as the second feature, due to the reason that the target of UDP flooding attack is usually a client running a particular service based on UDP on the victim. For example, the default port of DNS protocol is 53, and the attackers will send large amounts of attack packets to to saturate the resources from the targeted network via this port and makes it unavailable for other users. The distribution of destination port in attack flow will concentrate to the service port, and the entropy value yields the same result as above. From this we conclude that destination port can be used to detect UDP flooding attack.

IV. ALGORITHM

In this section, the algorithm of attack detection with joint-entropy will be presented. It should be noted that the algorithm is a framework. In this method, the feature of the flows can be chosen on demand. In the pseudo code below, we use f_1 and f_2 to denote the value of appointed features. t is a preset threshold which represents the maximum decrease of entropy that we can tolerate. Besides, this algorithm runs every time that the Netflow sample packets are sent to our Attack Detection Module, which will be described in Section V.

We use two dictionaries (or hash map) named $count\_table$ and pdt to store the counts of every different value of the

appointed features, and its corresponding probabilities, respectively. The algorithm can be described as Algorithm 1.

Algorithm 1 Attack detection with joint-entropy

Input:
NetFlow samples $Flows$, the value of feature f_1, f_2, threshold t, normalized entropy of normal traffic H_N;
Output:
A boolean value to infer whether there is an attack.

Initialize dictionary of counting $count\_table$ and dictionary of probability density pdt;
for each $flow$ in $Flows$ **do**
  if $(flow.f_1, flow.f_2) \notin count\_table$ **then**
    $count\_table$.add($(flow.f_1, flow.f_2)$, 1);
  else
    $count\_table[(flow.f_1, flow.f_2)] += 1$;
  end if
end for
$total = \sum(count\_table.\text{values}())$;
for each c in $count\_table$ **do**
  pdt.add(c.key, c.value / $total$);
end for
Joint-entropy $h = 0$;
for each p in pdt.values() **do**
  $h += p * \log_2(p)$;
end for
Normalized entropy $H = -h/\log_2(pdt.\text{count}())$;
$dH = H_N - H$;
if $dH > t$ **then**
  $is\_attacked = True$
else
  $is\_attacked = False$
end if
return $is\_attacked$

According to Equation 2, we need calculate the double summation for the entropy, but in fact the double summation can be transformed to a single summation. This makes our algorithm perform no slower than single-entropy methods, which is the ensurance of real-time. After the attacks are detected, the statistics of protocols information will be calculated and the specific type of attack will be classified.

V. Experiments and Evaluation

In this section, we describe our experiments about the algorithm with joint-entropy, and evaluate 1) the change of entropy with time, 2) the average decrease of entropy based on different attack intensity, and 3)detection rate, and false positive rate of the proposed method. All these evaluations are compared to the existing single-entropy-based methods. In the experiments, we use OpenVSwitch(OVS) [20] and Ryu controller [21] to deploy our algorithm module.

A. Experiment Setup

We use Mininet [22] to simulate the network environments. The whole system is installed on a server with four-core 2.30 GHz Intel Core-i5 CPU and 16 GB RAM running Ubuntu 14.04. The network topology is shown in Figure 1. There are 40 hosts in the topology. We pick one host to act as a server and victim. Then we divide the rest into 2 groups. One group including 20 hosts acts as a botnet that controlled by one host. The other group including 19 hosts acts as generating normal traffic to server. This OVS switch is controlled by Ryu Controller, which runs router application and our NetFlow Collector module.

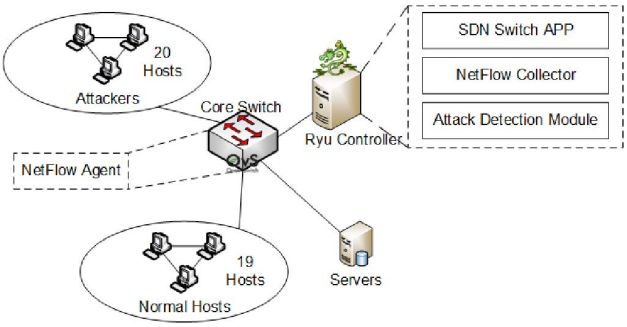


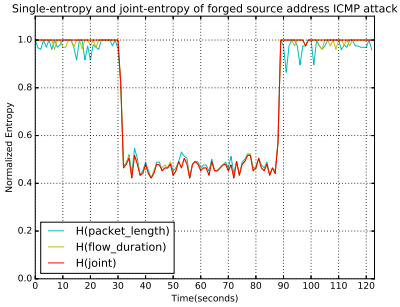
Fig. 1: Experiment Topology

B. Results

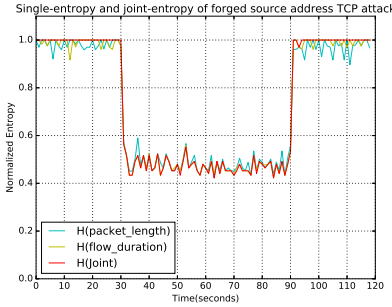
Three groups of simulation experiments were carried out. The first group is to validate that the entropy value will drop down when DDoS attacks occurs. Moreover, the purpose of the first group is to compare the change of joint-entropy and single-entropy values with time. The second group is to compare the average decrease of joint-entropy and single-entropy with different attack intensities. At last, we conduct another group of experiments to examine the comparison of our proposed method and traditional single-entropy-based methods, in terms of detection rate and false positive rate. The simulation results are shown as follows:

1) Change of entropy based on time: In this experiment, we compare the change of our joint-entropy-based method and previous single-entropy-base method with separate features. We simulate normal traffic according to the pattern of CAIDA 2007 dataset [23] for 120 seconds, and use TFN2K [24] to generate attack traffic for about 60 seconds, from the 30th second to the 90th second. We set active timeout of NetFlow to 1 second in order to get detection granularity as small as we can. As the Figure 2 shows that when traffic is normal, the joint-entropy is higher than single-entropy of one feature, and when the attack occurs, the decrease of joint-entropy value is larger than the decrease of single-entropy value. So the joint-entropy value is more sensitive than single-entropy value. In the other hand, the curve of joint-entropy has less jitters than the curve of single-entropy, especially in Figure 2d, Figure 2e and Figure 2f. We can infer that our method performs more stable than single-entropy methods.

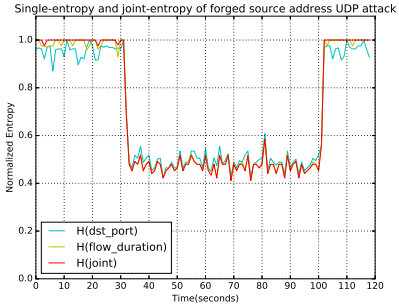
2) Average decreases of entropy under different attack intensity: The average decreases of entropy is defined as the difference between average ideal normal entropy and average



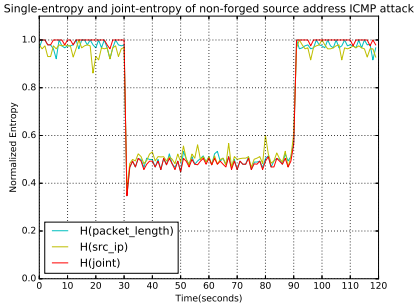
(a) Entropy change of forged source address ICMP attack



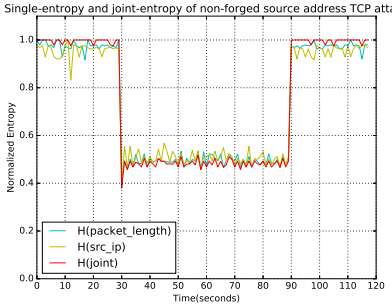
(b) Entropy change of forged source address TCP attack



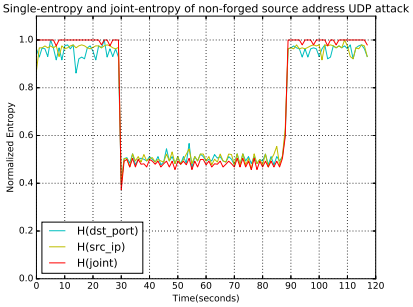
(c) Entropy change of forged source address UDP attack



(d) Entropy change of non forged source address ICMP attack



(e) Entropy change of non forged source address TCP attack



(f) Entropy change of non forged source address UDP attack

Fig. 2: Change of entropy based on time under forged or non-forged source address DDoS flooding attacks

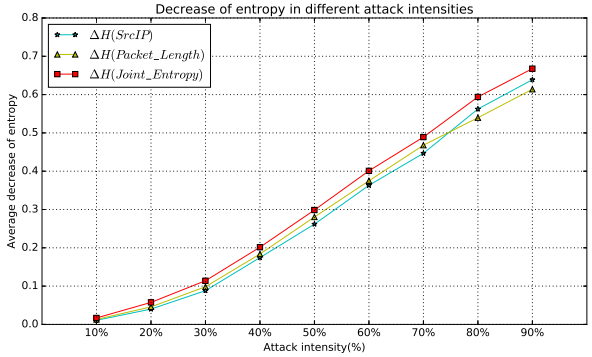


Fig. 3: Change of average decrease of entropy with different attack intensity

actual entropy. Let the former be denoted by $\overline{H}_N$ and the latter be denoted by $\overline{H}_A$, the average decreases of entropy $\Delta\overline{H}$ can be expressed as

$$\Delta\overline{H} = \overline{H}_N - \overline{H}_A \tag{5}$$

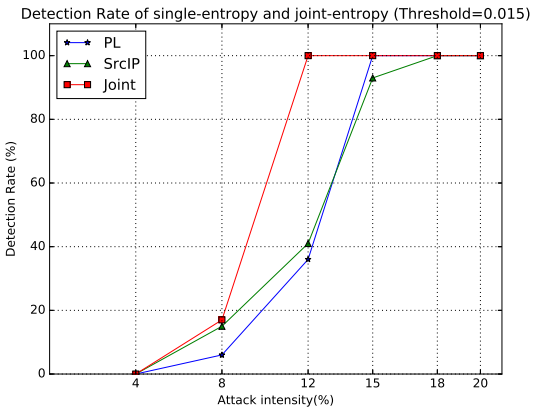
We adjust the attack intensity from 10% to 90% (i.e. the attack traffic ratio changes from 10% to 90% of total), and discover that the average decrease of joint-entropy is always larger than the average decrease of single-entropy, shown in Figure 3.

3) Detection Rate, False Positive Rate & False Negative Rate: In this experiment, the detection rate, false positive rate and false negative rate will be illustrated.

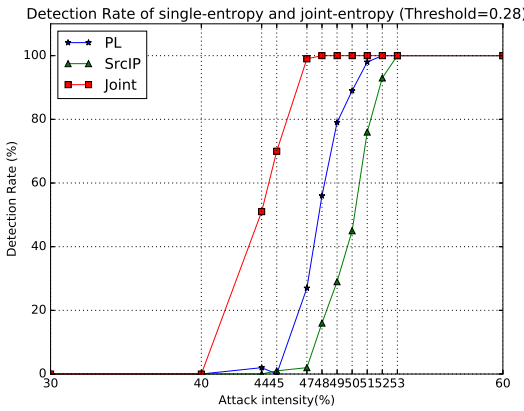
We define three typical types of attacks: low-rate attack, medium-rate attack and high-rate attack. The thresholds t of these 3 types of attacks are set to 0.015, 0.28 and 0.6, respectively. We carry out 100 times of experiments under every attack intensity and calculate the number of true positive events. The results is shown in Figure 4. In these figures, we use PL to denote Packet Length for short.

Figure 4b shows the situation in medium-rate attack. We choose this situation as a typical case. In interval $[44\%, 53\%]$, our method can archive better detection rate at every point than the other two single-entropy methods. When attack intensity is lower than 44%, the detection rates of all three methods are 0%. When attack intensity is higher than 53%, the detection rates are all 100%. The similar results are also shown in Figure 4a and Figure 4c.

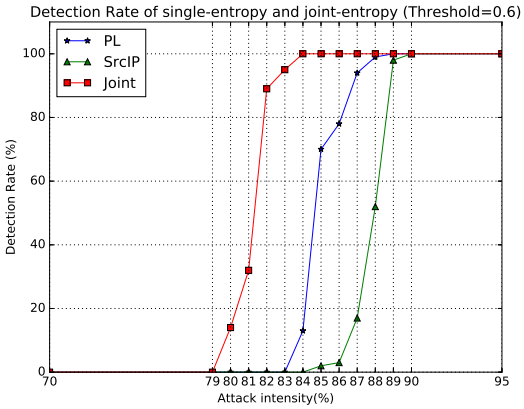
Figure 4d shows the comparison of false positive rate in different thresholds. It is not related to attack intensity. We adjust the threshold from 0.004 to 0.040, and observe the false positive events in normal traffic pattern. The result proves that our method always have less false positive events than the other two methods at the same threshold. Furthermore, all the three entropy methods will reach 0% false positive rate at 0.040, which is a small enough threshold.



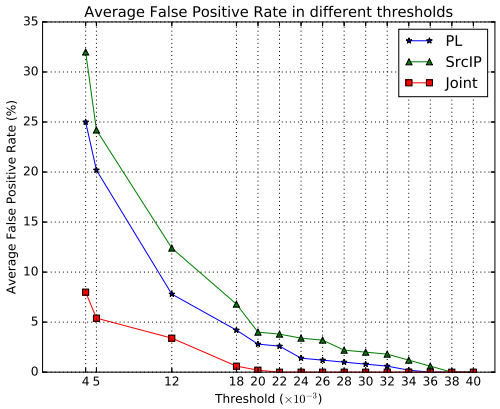
(a) Detection Rate of low rate attacks



(b) Detection Rate of medium rate attacks



(c) Detection Rate of high rate attacks



(d) False Positive Rate with different thresholds

Fig. 4: Detection Rate (DR) and False Positve Rate (FPR) of single-entropy and joint-entropy

VI. Discussion

In the first group of the experiments, The results show that the shapes of the six line charts are similar. Before and after attack, the traffic pattern in the network is random, so the normalized entropy is high. At the time from the 30th second to the 90th second, the attack traffic was injected into the network. Because of the packets in attack traffic is concentrated on some features, such as abnormal flow duration, simplex packet length, or concentrated source IP address or destination port, the entropy of traffic decreases significantly. Regardless the features we choose, the whole trend of entropy will have same shape.

Let us observe the Figure 2 again. The jitter of single-entropy is larger than joint-entropy. Because the single-entropy only considers one variable, according to the Equation 1. In the joint-entropy, according to the Equation 2, it should be noted that here X and Y are not independent variables, in other words X and Y interact with each other. So if we consider the joint probability of two features of network traffic, the jitter is controlled in a smaller range. Hence the joint-entropy algorithm performs more stable.

As mentioned earlier, the factor to judge whether a DDoS attack occurs is the decrease of entropy value. So the Figure 3 shows that the difference between the average decrease of joint-entropy and single-entropy becomes larger with the increase of attack strength. In low attack rate, the ratio of attack packets is low. So average decrease of joint-entropy and single-entropy is not large, too. In higher attack rate, the ratio of attack packets becomes higher. Because joint-entropy method uses two dimensions, it performs more sensitive than single-entropy. So the average decrease of joint-entropy is larger than single-entropy. That is why the distance of curve $\Delta H(joint)$ and the other two curves becomes larger with the increase of attack intensity.

Next is the analysis of Figure 4. We have discovered previously that the entropy will jitter at every sample point. According to Equation 5, assume that the entropy H_N is constant, the lower attack intensity is, the larger H_A is, then ΔH becomes smaller. When ΔH is always smaller than threshold t, the detection rate will be 0%. Similarly, the higher attack intensity is, the smaller H_A is, which makes ΔH larger.

If ΔH is always larger than t, the detection rate will be 100%. In other points, our joint-entropy method outperforms the other two single-entropy methods in terms of detection rate, too. Because we calculate two-dimension variables simultaneously, whether a flow is attack flow will be restricted by two features, while single-entropy methods only take into one feature. When attack occurs, the entropy decrease of joint-entropy will be larger than single-entropy, such that ΔH becomes larger than t more easily. Hence we can conclude that the ability of detecting attacks of joint-entropy is better than single-entropy.

According to Figure 2, in normal traffic, the jitter of single-entropy is larger than joint-entropy in terms of frequency and degree. Hence the possibility that H_A takes smaller value will increase, and ΔH will increase more easily. So single-entropy will exceed threshold more frequently and the false positive rate will be larger. Conversely, because joint-entropy method considers two variables simultaneously, the entropy value will barely exceed threshold and make the false positive rate lower. But when threshold is set too low, because of the inevitable jitter, the false positive rate will be larger in this situation.

VII. Conclusion and Future Works

DDoS attacks still impact the security and stability of the Internet. In the past, great effort has been devoted to the study of DDoS attack detection based on the single variable entropy. However, these methods could not take a full advantage of other packet features, and they did not consider to combine multiple features to get more sensitive and more accurate results. In this paper, we have proposed a joint-entropy-based DDoS detection mechanism with the following packet features: flow duration, packet length, source IP address and destination port. This joint-entropy-based algorithm can be used to detect different types of DDoS flooding attacks. The experiment results demonstrate that the proposed mechanism results in a better detection rate with lower false positive rate, comparing to existing single-entropy based methods.

For future works, we plan to find more features, and use them to product more accurate and faster attack detection with joint-entropy method. Moreover, the threshold in the current method is static, therefore future research into dynamic threshold value is still required. The proposed method is weak for detecting unknown or new types of DDoS flooding attacks, so the enhanced algorithm for detecting new types of attacks will also be our future work. Furthermore, continuing research on how to mitigate the DDoS flooding attacks based on the proposed joint-entropy method appears fully justified.

References

[1] N. Z. Bawany, J. A. Shamsi, and K. Salah. DDoS attack detection and mitigation using SDN: Methods, practices, and solutions. *Arabian Journal for Science And Engineering*, 42(2):425–441, (2017).

[2] Monowar H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita. An empirical evaluation of information metrics for low-rate and high-rate ddos attack detection. *Pattern Recognition Letters*, 51(Supplement C):1–7, (2015).

[3] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, (2008).

[4] Liying Li, Jianying Zhou, and Ning Xiao. *DDoS Attack Detection Algorithms Based on Entropy Computing*, pages 452–466. Springer Berlin Heidelberg, Berlin, Heidelberg, (2007).

[5] S. T. Zargar, J. Joshi, and D. Tipper. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Communications Surveys & Tutorials*, 15(4):2046–2069, (2013).

[6] N. Hoque, M. H. Bhuyan, R. C. Baishya, D. K. Bhattacharyya, and J. K. Kalita. Network attacks: Taxonomy, tools and systems. *Journal Of Network And Computer Applications*, 40:307–324, (2014).

[7] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov. Security in software defined networks: A survey. *IEEE Communications Surveys & Tutorials*, 17(4):2317–2346, (2015).

[8] Q. Yan, R. Yu, Q. Gong, and J. Li. Software-Defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. *Communications Surveys & Tutorials, IEEE*, PP(99):1–1, (2015).

[9] Truong Thu Huong and Nguyen Huu Thanh. Software defined networking-based one-packet DDoS mitigation architecture. In *IMCOM'17, January 05-07, 2017, Beppu, Japan*, pages 1–7. ACM, (2017).

[10] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris. Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments. *Computer Networks*, 62:122–136, (2014).

[11] sFlow. http://www.sflow.org/.

[12] Yiqin Lu and Meng Wang. An easy defense mechanism against botnet-based ddos flooding attack originated in SDN environment using sFlow. In *CFI'16,June 15-17, 2016, Nanjing, China*, pages 14–20. ACM, (2016).

[13] R. Wang, Z. Jia, and L. Ju. An entropy-based distributed D-DoS detection mechanism in software-defined networking. In *Trustcom/BigDataSE/ISPA, 2015 IEEE*, volume 1, pages 310–317, (2015).

[14] S. M. Mousavi and M. St-Hilaire. Early detection of DDoS attacks against SDN controllers. In *Computing, Networking and Communications (ICNC), 2015 International Conference on*, pages 77–81, (2015).

[15] Y. Xu and Y. Liu. DDoS attack detection under SDN context. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, (2016).

[16] H. Rahmani, N. Sahli, and F. Kammoun. Joint entropy analysis model for DDoS attack detection. In *Information Assurance and Security, 2009. IAS '09. Fifth International Conference on*, volume 2, pages 267–271, (2009).

[17] Sunny Behal and Krishan Kumar. Detection of DDoS attacks and flash events using novel information theory metrics. *Computer Networks*, 116(Supplement C):96–110, (2017).

[18] C.E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, (2001).

[19] NetFlow. https://en.wikipedia.org/wiki/Netflow.

[20] OpenVSwitch. http://openvswitch.org/.

[21] Ryu Controller. https://osrg.github.io/ryu/.

[22] Mininet. http://mininet.org/.

[23] The CAIDA UCSD "DDoS Attack 2007" Dataset. http://www.caida.org/data/passive/ddos-20070804_dataset.xml.

[24] TFN2K. https://en.wikipedia.org/wiki/Tribe_Flood_Network.