A FRAMEWORK FOR DETECTING DIFFERENT KINDS OF DDOS ATTACKS AND DIFFERENTIATING THEM FROM FLASH CROWDS

JALAJA VISHNUBHOTLA[1]

P L SRINIVASA MURTHY[2]

Dept of CSE, Institute of Aeronautical Engineering, Dundigal, Hyderabad – 500043.

jalajaravikanth@gmail.com, plsrinivasamurthy@gmail.com

ABSTRACT: Distributed computing technology paved way for realization of various applications that run in multiple servers across the globe. When the application has its large scale presence and impact on the user base across the globe, it is often target of Distributed Denial of Service (DDoS) attacks. DDoS is an organized effort of adversaries to deny service of a major Internet application that is used by millions of users. The servers rendering such application are targeted by the attackers. The genuine services provided by the servers are not reached to users due to the attack. There is another activity that resembles DDoS attack but not an attack. It is known as flash crowd which generates large traffic patterns that look like DDoS attack traffic patterns. Differentiating them from DDoS attacks is also important. In the present literature many contributions were found on both of them. However, a framework that can detect multiple kinds of DDoS attacks and discriminating them from flash crowds is very much desired. This paper provides required methodology to achieve this. NS2 simulations are used to demonstrate proof of the concept. Results of the experiments showed that the proposed system is useful as part of protecting distributed systems from DDoS attacks.

Key Words –DDoS attack, flash crowd, entropy, discrimination between flash crowed and DdoS

I. INTRODUCTION

II.

Cyber security is a biggest Challenge. Safeguarding digital resources is to be given paramount importance. DDOS attacks are launched by adversaries using bot net, a collection of nodes that have been compromised by adversaries. Compromised nodes are a set of nodes controlled by a botnet. DDOS attack is a most popular threat known as volumetric attack. In this kind of attack, the target system is overwhelmed with flood of requests resulting in large scale denial of service. In DDOS attack large number of machines act cooperatively under the supervision of one or more bot masters. These bots may be malicious users by themselves or maybe preliminarily infected.

Various vulnerabilities of the network are exploited by the attacker. Such vulnerabilities can be the weakness of the protocols like Telnet, FTP, TCP and HTTP. An array of compromised nodes is called zombies or bots. There might be a network to assist the attack. The network may be of P2Pnetwork or IRC network. Each node communicates with the other node in the network. The attack rate can be increasing, constant rate or variable rate where a huge data for short period of time repeatedly transmitted for regular or irregular intervals of time.

Application layer attacks target server side resources to be exhausted and there will be large denial of service. The attacks interestingly use less bandwidth. Examples of such attacks include hidden field manipulation, buffer overflow, SQL injection, cookie poisoning, and cross site scripting. Here a compromised node is the node that contains malicious code to launch attacks. Each of such node acts as a bot HTTP is used to bypass security mechanism while performing communications. Using HTTP it is not easy to understand attack traffic from that of genuine one. In this context botmaster takes responsibility to ensure botnet is saved and not discovered.

Normal traffic patterns can be recognized by certain patterns and the bot character can be found in such traffic. The malicious attacks are disguised as normal ones. By considering the observation of monitoring of messages that come from bots, anomalous behavior can be established. In order to identify a botnet proper identification threshold should be designed. These bots are capable to learn possible interactions of application layer. Thus the attacker gets ability to make attack without being identified.

Building new possibility by introducing a formal model for DDOS attacks where the botnet gets its emulation dictionary. Each bot owns a unique dictionary. The traffic analyst collects traffic patterns across the network. All the malicious flow seems to be well behaved. Crucial network monitoring is known as dynamic anamalography. The botnet is given a strong power of learning. With this, as time goes on, emulation dictionary becomes bulky and shows difference across messages. The empirical dictionary of two normal users is weakly correlated. It is understood that distinct users generally share distinct surfing experience.

In the literature many solutions came into existence. They include statistical approaches [1], fusion of information flows [5], sparity and low ranking [13], traffic and anomaly maps [14] and information metrics [16]. However, there is common thread in all the solutions. Most of the researchers focus on specific type of DDoS attack. There is no framework that can cater to the needs of detecting multiple kinds of attacks and also discriminate them from flash crowds for well informed decision making. The following are the contributions of this paper.

1. We proposed a framework with underlying detection mechanisms to detect different kinds of DDoS attacks like DDoS attacks, high volume DDoS attacks, spoofed DDoS attacks, sophisticated DDoS attacks and flash events.
2. We implemented the framework using NS2 simulations to demonstrate proof of the concept. The implementation methodology includes consideration of detection rate, classification rate, and false positive rate and so on. The results revealed the utility of the proposed framework.

The remainder of the paper is structured as follows. Review of literature on DDoS attacks is made in Section 2. Section 3 presents the proposed framework. Section 4 covers the experimental design and environment in which empirical study is made. Section 5 presents results of experiments. Section 6 concludes the work and provides guidelines for future work.

III. RELATED WORK

This section provides review of literature pertaining to DDoS attacks and the methods to detect and prevent them. The performance of the methods depends on network conditions and is influenced by many parameters. There should be a generic method to defend most of the attacks irrespective of the protocol used. A trace back mechanism should be implemented with customization support. It should be cost effective without compromising Quality of service [9].

A mathematical model to detect shrew attacks was proposed by taking into account in such way that there will be well defined TCP behavior with respect to adaptation mechanism of congestion window [3]. Attack pattern in the given network is used to determine the attack effect. The results of analysis suggest the tuning of attack parameters in order to maximize attack effect and the procedure to configure network in order to reduce shrew attacks [16].

Information distance is calculated between attack traffic and legitimate traffic [3]. There are several methods to identify DDOS attacks both at the core of the network and also at the edge of the routers. It is achieved by computing frequency and entropy of sorted distribution as explored in [1]. More details on the relationship of nodes in botnet can be found with monitoring and techniques based on botnet. Volumetric attacks have a severe impact on data plane but not on controller. The impact is visible only in attack phase[9]. Protocol exploitation doesn't have effect on network band width. The effect is on consumption of resources like logical ports. More detailed detection system is proposed which will analyze where the attack occurred either in transit or source. The dynamic nature of the stealthy attacks is studied because the technique benefits from increased correlation arising under shifting patterns in network traffic[2].More investigation is required to evaluate the trade - -offs among space and time granularity of monitoring the number of observations and the ability to detect attacks under decreasing levels of intensity[2]. TCP SYN attack consumes data structure on the server operating system [3]. Retransmission leads to severe congestion and finally time out. Once a malicious host is detected the packets are filtered and the services get resumed. Anomaly detection is done by various statistical methods, machine learning and soft computing. Routers can be configured via the access control list to access the network and drop suspected traffic If you filter all incoming ICMP traffic to broad cast address at the router none of the machines will respond and the attack will not work. Based on macroscopic level a hierarchical method is proposed in order to capture traffic patterns with spatial-temporal domains [2]. Macroscopic characteristics found in network traffic are one of the ways to detect DDoS. When this approach is coupled with dynamic monitoring capabilities, it will have higher utility. The solution in [2] could provide warnings when detection is made. The model used to launch attack was made with minimal cost and attacks are prevented for showing the performance of the approach.

Packet marking is done by either probabilistic or deterministic method. The information that has been marked is used to identify source of attack. Information inspection procedure is launched by router in order to have successful forwarding of packets. In this context the randomness of the traffic is measured using entropy and the given time interval is exploited by the router for the same. Spoofed IP addresses are dropped which does not belong to the local network. Dynamically change the IP address of a victim to evade attackers. Fault tolerance should be applied at all 3 levels hardware level, software level and system level. The methodology provided is used to capture the behavior of congestion of TCP on the side of victim. The attack patterns and the present network environment can provide the presence of botnet. Then there needs further research on the tradeoffs possible between inference performance and learning ability. From the review of the literature, it is understood that the solutions are useful for detecting DDoS attacks. However, there is not hybrid approach that can detect different kinds of DDoS attacks and provide more useful framework in such a way that it and discriminate DDoS from flash crowds as well.

IV. PROPOSED MODEL

This section provides details of the proposed system. Finding anomaly in patterns of traffic is the main model operandi in the proposed system. It identifies actual DDoS attacks and differentiates the same from the flash crowds. Packets that flow in network are examined by correlating them from different places. Patterns are reflected due to man-made traffic based attacks. The packets related to DDoS attacks differ from the normal packets in the order and general structure. The randomness or the degree of the disorder is known as entropy. Therefore entropy is an important measure used for making detection decisions. Entropy also helps in finding source IP distribution. The proposed approach in this paper is illustrated in Figure 1. It is meant for classifying network traffic into various types of DDoS attacks or flash crowd.

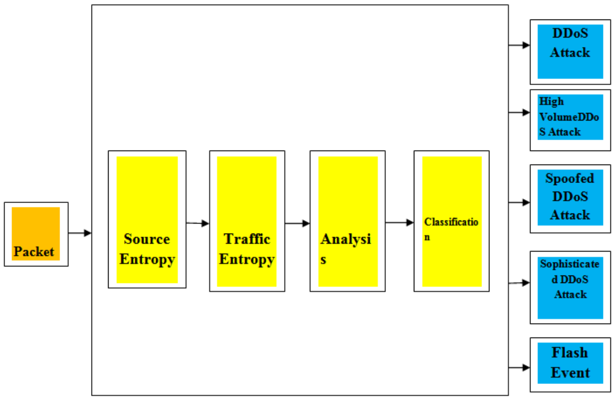
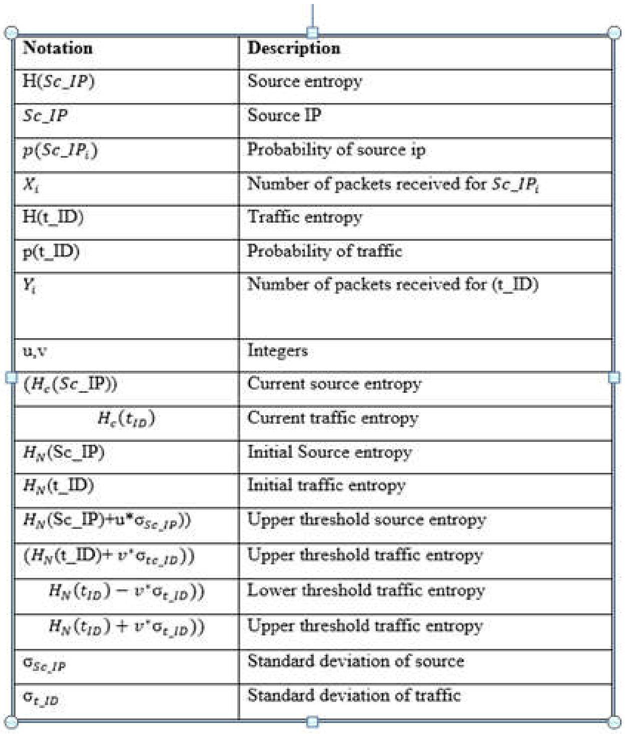


Fig 1: Syetm Architecture

When adversaries launch DDoS attack, the status of network is changed gradually in terms of entropy. I can be estimated with temporal differences. For instance at time 1 the traffic condition is recorded. Later on at time t2, known as end time, the traffic dynamics are changed. When there is gradual increase in traffic, it needs some time to detect DDoS attack. The time difference between t2 and t1 is known as the detection delay time. The number of source IP address that caused the DDoS attack can be called as source entropy. When traffic is considered coming from nodes of same network, such thing is traffic cluster. Entropy related to such cluster is called as traffic entropy. The network traffic characteristics and server load features are analyzed generally. The latter includes different parameters like utilization of CPU by users, utilization of CPU at system level, CPU load and memory utilization. The former on the other hand includes notion of traffic cluster and computational complexity. Traffic entropy is employed in order to derive thresholds used to make well informed decisions. Thus false positives can be decreased and the rate of detection of DDoS attacks is increased.

When simulation is started, the traffic entropy and source entropy are computed initially. After generating packets (normal user or attacker), the traffic gets increased. Then the entropies like traffic entropy and current source entropy are computed. Other aspects considered are traffic deviation and source deviation. Based on conditions provided below, the decisions are made to determine high volume DDoS, sophisticated DDoS, spoofed DDoS and DDoS. The mathematical conditions used different notations presented in Table 1.



Notation	Description
H(Sc_IP)	Source entropy
Sc_IP	Source IP
$p(Sc\_IP_i)$	Probability of source ip
X_i	Number of packets received for Sc_IP_i
H(t_ID)	Traffic entropy
p(t_ID)	Probability of traffic
Y_i	Number of packets received for (t_ID)
u,v	Integers
(H_c(Sc_IP))	Current source entropy
$H_c(t_{ID})$	Current traffic entropy
H_N(Sc_IP)	Initial Source entropy
H_N(t_ID)	Initial traffic entropy
H_N(Sc_IP)+u*σ_{Sc_IP}))	Upper threshold source entropy
(H_N(t_ID)+ $v^*\sigma_{tc\_ID}$))	Upper threshold traffic entropy
$H_N(t_{ID}) - v^*\sigma_{t\_ID}$))	Lower threshold traffic entropy
$H_N(t_{ID}) + v^*\sigma_{t\_ID}$))	Upper threshold traffic entropy
σ_{Sc_IP}	Standard deviation of source
σ_{t_ID}	Standard deviation of traffic

Table 1: Shows notations used

Source IP address is denoted as Sc_IP. It is a logical address made up of 4-bytes. The notation t represents traffic cluster while H(Sc_IP) denotes entropy of source address. There is a random variable denoted as H(Sc_IP) which can have different values like Sc_In1, Sc_IP1, Sc_IP2 and so on in various packets. The number of packets received from source IP address are denoted as X1, X2, X3, …, X_n.

H(Sc_IP)=-$\sum_{i=1}^{N} p(Sc\_IP_i)\ log_2(Sc\_IP_i)$

p (Sc_IP) ={p(Sc_IP_1) p(Sc_IP_2)............p(p(Sc_IP_n)

$p(Sc\_IP) = \frac{X_i}{S}$ where $S=\sum_{i=1}^{n} X_i$

Similarly traffic entropy is defined as

$H(t\_ID)=-\sum_{i=1}^{M} p(t\_ID_i)\, log2p((tc\_ID_i)$

$p(t\_ID)=p(t\_ID_1) \quad p(t\_ID_2) \ldots\ldots p(t\_ID_n)$

$p(t\_ID)=\frac{Y_i}{S} \quad S=\sum_{i=1}^{n} Y_i$

DDOS attack condition

$(H_c(Sc\_IP))>(H_N(Sc\_IP)+u*\sigma_{Sc\_IP}))$

[current source entropy>upper threshold source entropy]

$(H_c(t\_ID)>(H_N(tc\_ID)+v^*\sigma_{tc\_ID}))$

[current traffic entropy > upper threshold current entropy]

Flash event condition

$(H_c(Sc\_IP))>(H_N(Sc\_IP)+u*\sigma_{Sc\_IP}))$

[current source entropy>upper threshold source entropy]

$(H_c(t\_ID)<(H_N(t_{ID}) - v^*\sigma_{t\_ID}))$

[current traffic entropy<lower threshold traffic entropy]

Spoofed DDOs Attack Condition

$(H_c(Sc\_IP))>(H_N(Sc\_IP)+u*\sigma_{Sc\_IP}))$

[Current source entropy>upper threshold source entropy]

$(H_c(t\_ID)>(H_N(t\_ID)-v^*\sigma_{t\_ID}))$

[current traffic entropy > upper threshold current entropy]

High volume DDoS Attack condition:-

$(H_c(Sc\_IP))<(H_N(Sc\_IP)+u*\sigma_{Sc\_IP}))$

[current source entropy<upper threshold source entropy]

$(H_c(sc_{IP})) < (H_N(sc_{IP}) - u^*\sigma_{sc\_IP})$

[current source entropy < lower threshold source entropy]

Sophisticated DDoS Attack Condition

$(H_c(Sc\_IP))<(H_N(Sc\_IP)+u*\sigma_{Sc\_IP}))$

[current source entropy<upper threshold source entropy]

$(H_c(t\_ID)>(H_N(tc\_ID)+v^*\sigma_{tc\_ID}))$

[current traffic entropy > upper threshold current entropy]

Standard Deviation

This measure is used to know dispersion of different values given in data from the mean value. It is computed as square root of variance which find different among data points given relating to the computed mean value. It is computed as follows.

$$\sigma=\sqrt{\frac{\sum(x-\bar{x})^2}{N}}$$

Here the value of packet is denoted as x. Packet value can have a mean value which is represented by $\bar{x}$. And N represents number of packets while the computation of mean of packet value is computed as follows.

as$\bar{x} = \sum \frac{x}{N}$.

V. EXPERIMENTAL DESIGN

This section described experimental design. The design of network reflects Interne with a transit-hub. As DDoS attacks occur in wide area networks where different networks are grouped together, this kind of design is preferred. The whole network is divided into different domains. Sub networks with host systems are considered to connect to Internet. Transit stub network is used to have inter-connection of transit networks. Stub networks are made up of nodes to generate network traffic. Stub networks also have user and attack users connected tto them. The purpose of this design is to investigate the DDoS attacks and prevention measures proposed in this paper.

GT-ITM is used as simulation topology with as many as 8 ISPs and 12 routers of transit networks. Nodes that generate traffic of FE and DDoS besides genuine ndoes are connected to transit networks. As many as 400 nodes are used to generate traffic pertaining to genuine users, flash events and DDoS attacks. The environment is created using NS2. The total simulation time considered is 70 seconds. Random approach is used to generate the three kinds of traffic with some interval. HTTP protocol is used to generate legitimate web traffic while UDP is employed to have DDoS attack traffic. The attacker nodes are used to modify TCP protocols in order to achieve DDoS attacks with high success rate.

The attacker nodes generate huge amount of traffic so that web servers receive of high volumes of attack traffic. In NS2 settings CBR mode is configured to get UDP flows. The attack is observed with an adjustable time window. Computation of entropy of traffic cluster is made with different size of windows. Consideration of different performance metrics such as false positive rate, false negative rate, detection rate and classification rate is done based on the server capacity. Period for attacks is considered to be 30-35 seconds. Start and end times considered for flash traffic are 25-30 second and 45-50 seconds. Whereas the start and end times for legitimate traffic are considered randomly between 1-20 seconds and 70 seconds.

VI. RESULTS AND DISCUSSION

Simulated experiments are made as per the design described above. Source address entropy and traffic cluster entropy are used to form a detection metric with logical AND/OR. Various observations are recorded in experiments. There is increase in the source address when DDoS attacks and FEs are made simultaneously. Thus it is determined that the majority of detection decision is made using traffic cluster entropy. It is used for not only detecting DDoS attacks but also to find whether it is FE. Six sigma method is employed to have thresholds for source address entropy. For traffic cluster entropy ROC curve is used to obtain values. Simulations are made with normal web traffic and attack traffics.

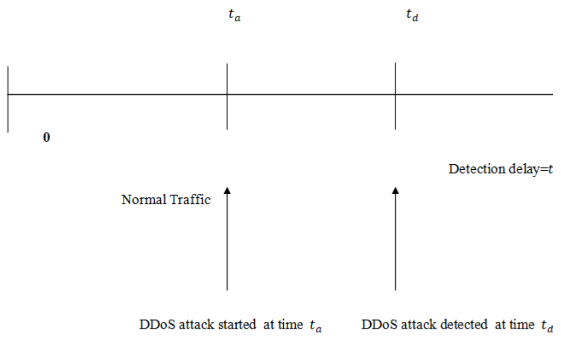

t_a
t_d
0
Detection delay=t
Normal Traffic
DDoS attack started at time t_a
DDoS attack detected at time t_d

Figure 2: Detection delay computation

As can be seen in Figure 2, there start and end times are used to know detection delay time. The time difference between DDoS attack detection time and its start time is considered as detection time.

Detection delay=$t_d - t_a$

There is another measure known as tolerance factor which reflects deviation that is required to have DDoS attack to be raised. This is the basis for finding normal detection, best false positive rate and best detection rate. A low value set for tolerance factor or best detection rate. Thus detection rate is improved besides reducing false negatives. It also considers the difference between normal states and attack states to have accurate detection of attacks. When tolerance factor is kept high, it led to achievement of best false positive rate. However, it leads to decreased detection rate. When there is increase in the detection of false positives, the detection rate of attack is decreased. Therefore, the tolerance factor is set to medium for normal detection scenario. Thus it is able to balance the false positives and false negatives. The range of tolerance factor is 1 to 10. Different kinds of traffics are used for experiments. Time window 0.6 is used for attack detection. For normal defence of attacks tolerance factor is set to either 5 or 6. For best detection rate, it is set at 7 or 8 or 9. For achieving bet false positive rate, the tolerance factor is set at 10. Detection rate is computed as follows.

$$D_R = \frac{TP}{TP+FN}$$

False positive rate is computed as follows.

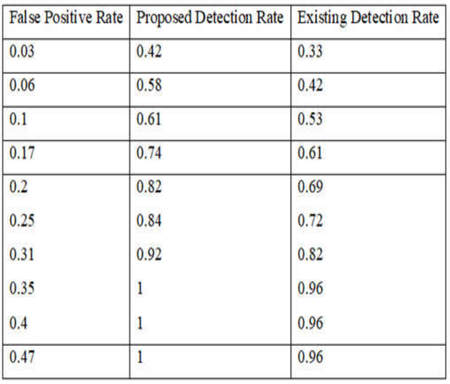
$$FP_R = \frac{FP}{TN+FP}$$

The classification rate is computed as follows.

$$C_R = \frac{TP+TN}{TP+TN+FP+FN}$$

The observations are presented in the remainder of this section of this section.

Table 2: Shows detection rate performance



False Positive Rate	Proposed Detection Rate	Existing Detection Rate
0.03	0.42	0.33
0.06	0.58	0.42
0.1	0.61	0.53
0.17	0.74	0.61
0.2	0.82	0.69
0.25	0.84	0.72
0.31	0.92	0.82
0.35	1	0.96
0.4	1	0.96
0.47	1	0.96

Table 2 : shows that the detection rate is influenced by the false positive rate. For this reason, when false positive rate is increased, detection rate is increased

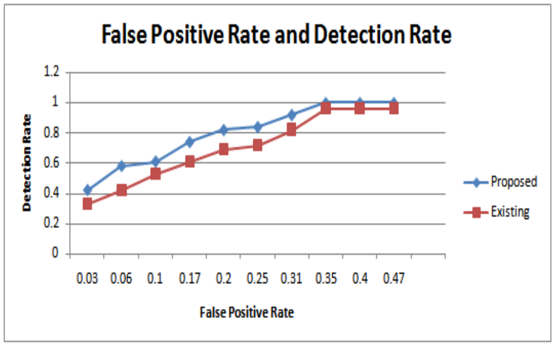
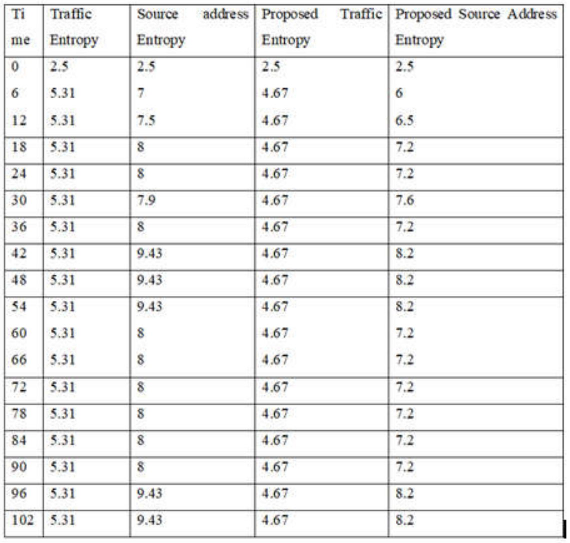


Figure 3: Dynamics of detection rate and false positive rate

Figure 3 shows the behaviour of two metrics known as detection rate and false positive rate for both existing and proposed systems. The performance of the proposed system is better than that of existing system.



Ti me	Traffic Entropy	Source address Entropy	Proposed Traffic Entropy	Proposed Source Address Entropy
0	2.5	2.5	2.5	2.5
6	5.31	7	4.67	6
12	5.31	7.5	4.67	6.5
18	5.31	8	4.67	7.2
24	5.31	8	4.67	7.2
30	5.31	7.9	4.67	7.6
36	5.31	8	4.67	7.2
42	5.31	9.43	4.67	8.2
48	5.31	9.43	4.67	8.2
54	5.31	9.43	4.67	8.2
60	5.31	8	4.67	7.2
66	5.31	8	4.67	7.2
72	5.31	8	4.67	7.2
78	5.31	8	4.67	7.2
84	5.31	8	4.67	7.2
90	5.31	8	4.67	7.2
96	5.31	9.43	4.67	8.2
102	5.31	9.43	4.67	8.2

Table 3: Shows temporal variation in presence of normal web traffic with FE

Table 3 shows time, traffic entropy, source address entropy, proposed traffic entropy and proposed source address entropy.

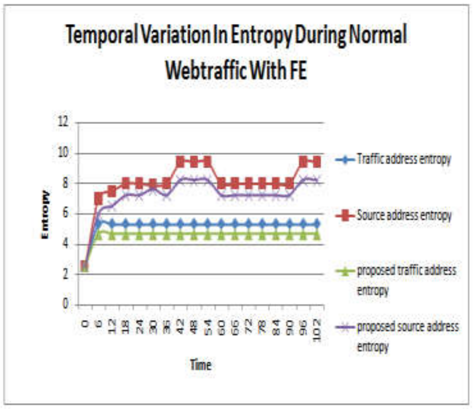
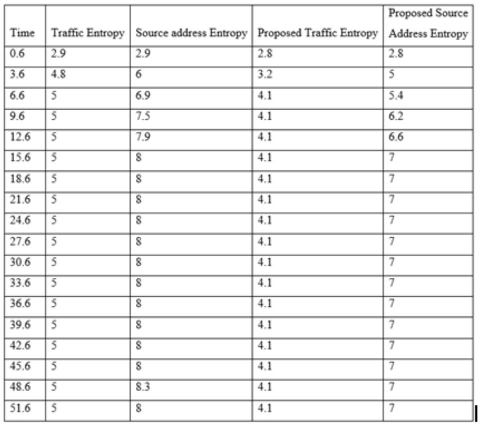


Figure 4: Shows temporal variation in entropy observations with normal web traffic (FE)

As shown in Figure 4, temporal variation of entropy is presented for both existing and proposed systems and the proposed system shows better performance over the existing system.



Time	Traffic Entropy	Source address Entropy	Proposed Traffic Entropy	Proposed Source Address Entropy
0.6	2.9	2.9	2.8	2.8
3.6	4.8	6	3.2	5
6.6	5	6.9	4.1	5.4
9.6	5	7.5	4.1	6.2
12.6	5	7.9	4.1	6.6
15.6	5	8	4.1	7
18.6	5	8	4.1	7
21.6	5	8	4.1	7
24.6	5	8	4.1	7
27.6	5	8	4.1	7
30.6	5	8	4.1	7
33.6	5	8	4.1	7
36.6	5	8	4.1	7
39.6	5	8	4.1	7
42.6	5	8	4.1	7
45.6	5	8	4.1	7
48.6	5	8.3	4.1	7
51.6	5	8	4.1	7

Table 4: Shows entropy variation with normal web traffic

Table 4 shows the observations made on the entropy variations with respect to simulation time. The observations are made with the normal web traffic.

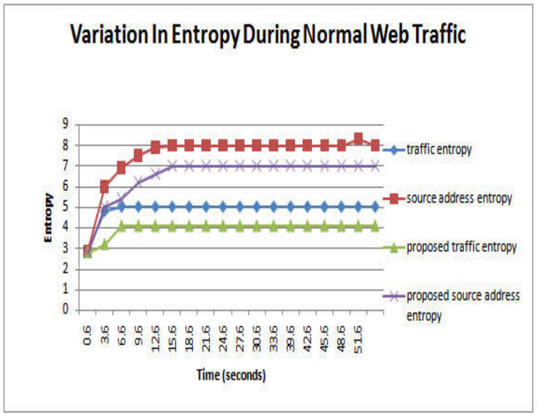


Figure 5: Entropy variations with normal web traffic

As presented in Figure 5, the entropy variations of normal web traffic are presented with existing and proposed approaches. The differences of traffic and source address entropies are presented.

VII. CONCLUSIONS AND FUTURE WORK

Distributed Denial of Service (DDoS) attacks in wide area networks are attacks made by adversaries with the help of thousands of compromised nodes or zombies. Thus DDoS attacks are essentially made with large scale denial of service intentions. Thus DDoS attacks became potential risk to Internet wide applications. In this paper we proposed a framework to detect different kinds of DDoS attacks and also provided mechanism to discriminate the flash events from DDoS attacks. The framework makes use of different kinds of entropies mathematically to detect and differentiate attacks. For instance, it used source address entropy and traffic cluster entropy to identify DDoS attacks. Different measures are used to achieve this. Evaluation of the proposed framework is also made with classification rate, detection rate, and false positives and so on. Experimental design is provided explicitly and the simulations are made using NS2 to show the effectiveness of the proposed framework. Experimental results revealed the ability of the framework to detect different DDoS attacks accurately. In future, we intend to have a real case study to evaluate our work in a realistic environment. Thus the proposed framework can be improved further to achieve accuracy in real time environments.

VII. REFERENCES

[1] Laura Feinstein, Dan Schnackenberg and RavindraBalupari, Darrell Kindred. (2003). Statistical Approaches to DDoS Attack Detection and Response1. IEEE, p1-12.

[2] Jian Yuan and Kevin Mills, Senior Member, IEEE. (2005). Monitoring the Macroscopic Effect of DDoS Flooding Attacks. IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING. 2, p324-335.

[3] JingtangLuo, Xiaolong Yang, Senior Member, IEEE, Jin Wang, Member, IEEE, JieXu, Member, IEEE, Jian Sun, Member, IEEE, and Keping Long, Senior Member, IEEE. (2014). On a Mathematical Model for Low-Rate Shrew DDoS. IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. 9, p1069-1083.

[4] Ashish Dutt, MaizatulAkmar Ismail, and TututHerawan. (2016). A Systematic Review on Educational Data Mining. IEEE, p1-15.

[5] AmeyaAgaskar, Ting He, Member, IEEE, and Lang Tong, Fellow, IEEE. (2010). Distributed Detection of Multi-Hop Information Flows With Fusion Capacity Constraints. IEEE TRANSACTIONS ON SIGNAL PROCESSING. 58, p3373-3383.

[6] Mauro Barni and Fernando P´erez-Gonz´alez. (2013). COPING WITH THE ENEMY: ADVANCES IN ADVERSARY-AWARE SIGNAL PROCESSING. IEEEp1-5.

[7] Mauro Barni, Fellow, IEEE, and BenedettaTondi, Student Member, IEEE. (2014). Binary Hypothesis Testing Game With Training Data. TRANSACTIONS

ON INFORMATION THEORY. 60, p4848-4866.

[8] Ting He, Member, IEEE, and Lang Tong, Fellow, IEEE. (2008). Distributed Detection of Information Flows. IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. 3 , p390-403.

[9] NazrulHoque, Dhruba K Bhattacharyya and Jugal K Kalita. (2015). Botnet in DDoS Attacks: Trends and Challenges. IEEE., p1-29.

[10] BhavyaKailkhura, Student Member, IEEE, Swastik Brahma, Member, IEEE, BerkanDulek, Member, IEEE, Yunghsiang S Han, Fellow, IEEE, Pramod K. Varshney, Fellow, IEEE. (2015). Distributed Detection in Tree Networks: Byzantines and Mitigation Techniques. IEEE., p1-13.

[11] Stefano Marano, Vincenzo Matta, and Lang Tong, Fellow, IEEE. (2009). Distributed Detection in the Presence of Byzantine Attacks. IEEE TRANSACTIONS ON SIGNAL PROCESSING. 57 , p16-29.

[12] Stefano Marano, Vincenzo Matta, Ting He, Member, IEEE, and Lang Tong, Fellow, IEEE. (2013). The Embedding Capacity of Information Flows Under Renewal Traffic. IEEE TRANSACTIONS ON INFORMATION THEORY. 59 , p1724-1739.

[13] MortezaMardani, Student Member, IEEE, Gonzalo Mateos, Member, IEEE, and Georgios B. Giannakis, Fellow, IEEE∗. (2011). Dynamic Anomalography: Tracking Network Anomalies via Sparsity and Low Rank†. IEEE., p1-37.

[14] MortezaMardani, Student Member, IEEE, and Georgios B. Giannakis, Fellow, IEEE. (2015). Estimating Traffic and Anomaly Maps via Network Tomography. IEEE,p1-15.

[15] ParvathinathanVenkitasubramaniam, Member, IEEE, Ting He, Member, IEEE, and Lang Tong, Fellow, IEEE. (2008). Anonymous Networking Amidst Eavesdroppers. IEEE TRANSACTIONS ON INFORMATION THEORY. 54 , p2770-2784.

[16] Yang Xiang, Member, IEEE, Ke Li, and Wanlei Zhou, Senior Member, IEEE. (2011). Low-Rate DDoS Attacks Detection and Traceback by Using New Information Metrics. IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. 6, p426-437.