International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-1, Issue-5, November 2011

Entropy Based Detection of DDOS Attacks

Anusha. J

Abstract*—Distributed Denial of service (DDOS) attacks is a critical threat to the internet. Due to the memory less features of the internet routing mechanism makes difficult to trackback the source of the attacks. In this paper, I find out the source of the attack with the help of entropy variation in dynamic by calculating the packet size, which shows the variation between normal and DDOS attack traffic, which is fundamentally different from commonly used packet marking techniques. In comparison to the existing DDOS trackback methods, the proposed one posses dynamic entropy variations as per the clients behavior.

***Index Terms*— DDOS, Method, Router**

I. Introduction

To trace back the source of the DDOS attacks in the internet is extremely hard. It is one of the extraordinary challenge to trackback the DDOS attacks, that attackers generate huge amount of requests to victims through compromised computers(zombies), in order to denying normal services or degrading the quality of services.

Recent survey shows that than 70 internet operators in the world demonstrated that DDOS attack are increasing dramatically and individual attacks are more strong and sophisticated. IP trace back means the capability of identifying the actual source of any packet across the internet; with the help of IP trace back schemes identify the zombies from which the DDOS attack packets entered the internet.

A number of IP trace back approaches have been suggested to identify attackers. Among them two major methods for IP trace back, Probabilistic packet marking (PPM) and deterministic (DDPM). Both of these require routers to inject marks into individual packets. And also provides some limitations such as scalability, huge demands on storage space and vulnerability to packet pollution. Both PPM and DPM also require duplicate on the existing routing software which is extremely hard.

For the DDOS attack detection compare the packet number distribution of packet flows, which are out of the control of attackers once the attack is launched, and found the similarity of attack flows is much higher than the similarity among legitimate flows eg : flash crowds. Entropy growth rate as the length of a stochastic sequence increases.

In this paper, I also together propose flow entropy variation to avoid packet marking. Here the packets that is passing making. Here the packets that is through a router into flows that was defined by the upstream router where a packet come from, and the destination address of the packet. During non attack periods, routers are required to observe and routed entropy variations. Once the attackers is launched the entropy rate increases dynamically to identify the locations of zombies. Upstream routers helps to identify where the attack flow cause from based on their local entropy variations that are mentioned.

II. Samples Network with DDOD Attacks

DDOS attacks are targeted at exhausting the victim's resources, such as network between, computing power and operating system data structures.

III. Steps to Launch the DDOS Attack

1) Attacker first establishes a network which is responsible for huge volume of traffic to deny the series of normal users.
2) Attackers then discover vulnerable hosts of the network. Vulnerable host in the sense that the system running no anti viruses or out of date anti viruses software.
3) Attackers The now install new programs known as attack tools on the compromised hosts.
4) It can be shown by the growth of entropy rate from the point of attack.

The flow is determined by calculating the best paths by choosing the shortest path algorithms.

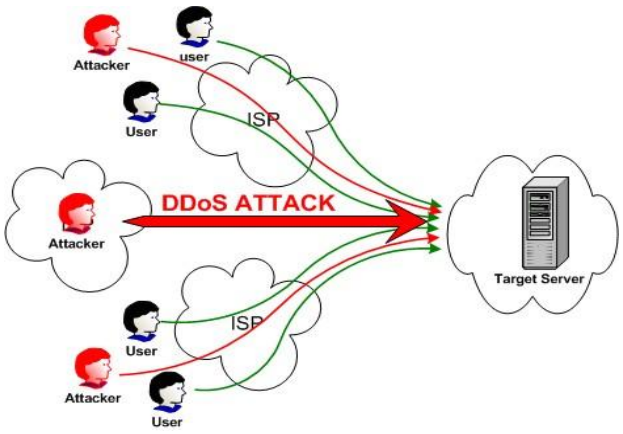


Fig.1: DDos Attack.

The fig:1 explain that number of client shares a server to exchange the information, among them one or more than one act as a attacker. This is notified with the help of entropy variation between the normal flow and attack flow.

Manuscript received May 04, 2012.

Anusha. J, M.E., Department of Computer Science & Engineering, Vins Christian College of Engineering, Nagercoil (Tamilnadu), India, (Email: anusha_88@ymail.com)

IV. Topology Creation

In order to communicate with one or more client with a server, network topology structures should be designed.

Fig:2 Network topology Structure

From the fig: 2, It shows that R1 and R2 act as the routers and S1 and S2 are the required server and c1 and c2 are the client. It also shows the connections among the server s1, Router R1, and Client C1 and as S2, R2, and C2 respectively. This network topology is responsible to share the information from one location to another.

V. System Transaction

Here the packets that are passing through a router into flows. A flow in the sense a pair that the upstream router where the packet came from and the destination address of the packet. Entropy which is an information theoretic concept, that helps to measure the randomness in the network. Here the entropy used to measure the changes of randomness of flows at a router for a given time interval with the help of packet size, used for transaction.

Generally, a router knows as local router in this network topology for e.g.: upstream router and downstream outers. The local area network attacked to the upstream routers. The router that used for investigating now as local router. The flow on a local router is denoted by $< u_i, d_j, t >, i, j \in I, t \in R$ where Ui is an upstream router of a local router Ri, dj is the destination of a group of packets that are passing through the local router Ri, and t is the current time stamp and i as the set of position integers, and R as the set of real numbers. If anyone router occurs two different incoming flows from the upstream routers, this kind of flow is denoted as transit flows.

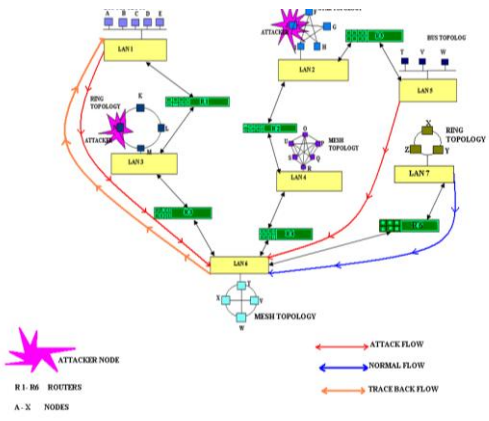


Fig: 3 Traffic flow is monitor at the upstream router

Therefore a flow at a router can be defined as follows;

$$f_{ij}(u_i, d_j)=\{<u_i, d_j, t> \mid u_i \in U, d_j \in D, i, j \in I\}$$

Where, ui, I € I as the immediate upstream router of the local router Ri which shown in fig:4

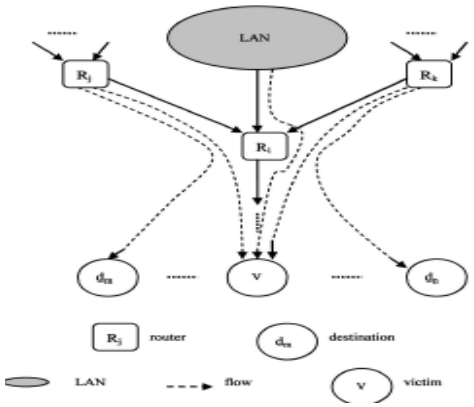


Fig:4 Traffic flows at a router on an attack path

In the fig: 4, all the incoming flows as input flows and all the flows that are leaving from router Ri as named as output flows. D represent the destinations of the packets that are passing through the local router Ri, Attacker is responsible for traffic flow at a router.

There by for a given time internal T, the variation of the number of packets for a flow as follows:

$$N_{ij}(u_i, d_j, t+\Delta T)=|f_{ij}(u_i\, d_j\, t+\Delta T)| - |f_{ij}(u_i, d_j, t)|$$

Here $|f_{ij}(u_i, d_j, t)| = 0$ therefore $N_{ij}(u_i, d_j, t+\Delta T)$ is the number of packets during the flow is f_{ij}.

Hence using the packet size variation, due to the attacker the entropy rate is defined as follows:

$$H(F) = -\sum_{ij} P_{ij}(u_i, d_j) \log P_{ij}(u_i, d_j)$$

Where $p_{ij}(u_i,d_j)$ as the probability of each flow at a router based on large number theorem .H(F) as the entropy variation used to measure the variations of randomness of flows.The flow design explain once the server is suffered by an attacker, it will install a new program known as attack tool when it is vulnerable host, due to this situation a huge amount of traffic is created at the upstream router. This can be estimate by monitoring the packet size, variation with the help of entropy rate. If the server is not suffered by any of an attacker then it shows the static entropy rate to explain the normal flow through the router.

VI. ALGORITHM FOR TRACEBACK MODEL

The local flow monitoring algorithm

1. initialize the local threshold parameter, C, δ, and sampling interval ΔT;
2. identify flows, $f_1, f_2, ..., f_n$, and set count number of each flow to zero, $x_1 = x_2 = ... = x_n = 0$;
3. when ΔT is over, calculate the probability distribution and the entropy variation as follows.

$$p_i = x_i \cdot \left(\sum_{i=1}^{n} x \right)^{-1}, \quad H(F) = -\sum_{i=1}^{n} p_i \log p_i;$$

4. save $x_1, x_2, ..., x_n$ and $H(F)$;
5. if there is no dramatic change of the entropy variation $H(F)$, namely, $|H(F) - C| \le \delta$, progress the mean $C[t] = \sum_{i=1}^{n} \alpha_i \cdot C[t-i]$, $\sum_{i=1}^{n} \alpha_i = 1$, and the standard variation $\delta[t] = \sum_{i=1}^{n} \beta_i \cdot \delta[t-i]$, $\sum_{i=1}^{n} \beta_i = 1$
6. go to step 2.

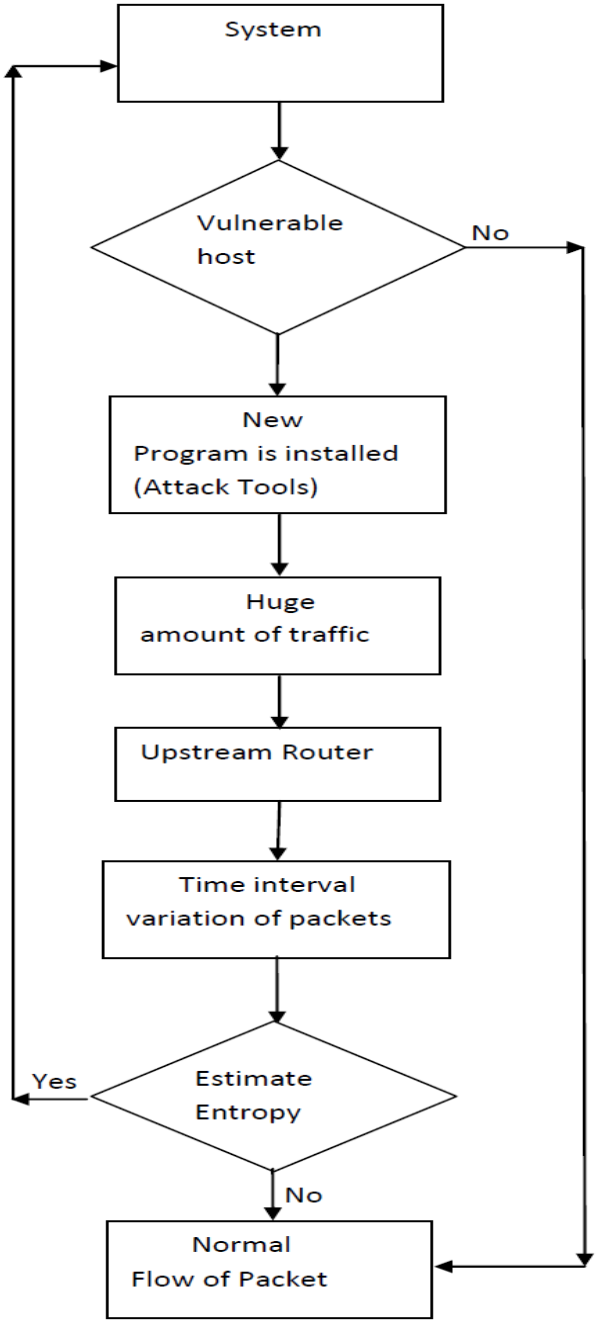
Fig. 6. The algorithm for local flow traffic monitoring

The IP traceback algorithm

1. initialize a set $A = \phi$, and obtain the local parameter of C and δ;
2. Let $U = \{u_i\}, i \in I$ be a set of the upstream routers, $D = \{d_i\}, i \in I$ be a set of the destinations of the packets, and V be the victim.
3. define attack flows, $f_i = <u_j, v>, i = 1,2,...,n, u_j \in U$, and sort the attack flows in the descent order, and we have $f_1', f_2', ..., f_n'$,
4. for i=1 to n
   {
   calculate $H(F \setminus f_i')$
   if ($|H(F) - C| > \delta$) then append the responding upstream router of f_i' to set A
   else break;
   end if;
   end for;
5. submit traceback requests to the routers in set A respectively, and deliver the confirmed zombies information, set A, to the victim.

Fig. 7. The IP Traceback algorithm on a router.

VII. ALGORITHM FOR TRACE BACK MODEL



In this section, the related algorithms according to our previous modeling and analysis. There are two algorithms in the proposed traceback suite, the local flow monitoring algorithm and the IP traceback algorithm. The local flow monitoring algorithm is running at the nonattack period, accumulating information from normal network flows, and progressing the mean and the standard variation of flows. The progressing suspends when a DDoS attack is ongoing. The local flow monitoring algorithm is shown as Fig. 6. Once a DDoS attack has been confirmed by any of the existing DDoS detection algorithms, then the victim starts the IP traceback algorithm, which is shown as Fig. 7. The IP traceback algorithm is installed at routers. It is initiated by the victim, and at the upstream routers, it is triggered by the IP traceback

requests from the victim or the downstream routers which are on the attack path. The proposed algorithms are independent from the current routing software, they can work as independent modules at routers. As a result, I do not need to change the current routing software.

VIII. Performance Evaluation

In this section the performance is evaluate the effectiveness and efficiency of the entropy variation based on IP Traceback mechanism here the First task is to show that the flow entropy variation is stable for non attack.

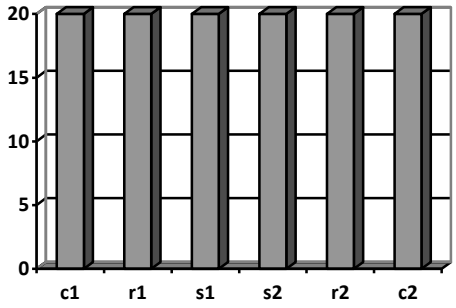


Fig:8 Graph that shows uniform entropy rate due to non attack

After estimating the first task I decided to find out the fluctuation for normal situations by adding an attacker at any one of the client (or) server, there by the second task is to demonstrate the relationship between the drop of flow entropy variation and the increase of attack strength, entropy rate due to the attacker at server1

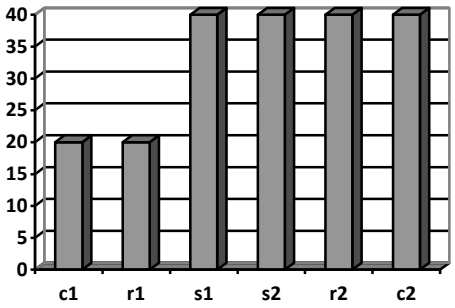


Fig:8 Graph explained the increase rate of flow

IX. Conclusion

In this paper I proposed an effective and efficient IP Traceback scheme against DDOS attacks based on entropy variations. Here the packet marking strategies is avoided, because it suffers a number of drawbacks. This paper employs by storing the information of flow entropy variations at routers. Once the DDOS attack has been identified it performs pushback tracing procedure. The Traceback algorithm first identifies its upstream router where the attack flows comes from and then submits the Traceback request to the related upstream router.

This procedure continues until the most far away zombies are identified. But in my existing case I used the static value to determine to determine the entropy rate. But in my proposed strategies I used dynamic value to determine the entropy rate which is based upon the packet size of the client's behavior.

REFERENCES

[1] "IP Flow-Based Technology," ArborNetworks, http://www.arbornetworks.com, 2010.
[2] C. Patrikakis, M. Masikos, and O. Zouraraki, "Distributed Denialof Service Attacks," The Internet Protocol J., vol. 7, no. 4, pp. 13-35, 2004.
[3] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems," ACM Computing Surveys, vol. 39, no. 1, p. 3, 2007.
[4] Y. Kim et al., "PacketScore: A Statistics-Based Packet Filtering Scheme against Distributed Denial-of-Service Attacks," IEEE Trans. Dependable and Secure Computing, vol. 3, no. 2, pp. 141-155, Apr.-June 2006.
[5] H. Wang, C. Jin, and K.G. Shin, "Defense against Spoofed IP Traffic Using Hop-Count Filtering," IEEE/ACM Trans. Networking, vol. 15, no. 1, pp. 40-53, Feb. 2007.
[6] Y. Chen and K. Hwang, "Collaborative Detection and Filtering of Shrew DDoS Attacks Using Spectral Analysis," J. Parallel and Distributed Computing, vol. 66, pp. 1137-1151, 2006.
[7] K. Lu et al., "Robust and Efficient Detection of DDoS Attacks for Large-Scale Internet," Computer Networks, vol. 51, no. 9, pp. 5036-5056, 2007.
[8] R.R. Kompella, S. Singh, and G. Varghese, "On Scalable Attack Detection in the Network," IEEE/ACM Trans. Networking, vol. 15, no. 1, pp. 14-25, Feb. 2007.
[9] P.E. Ayres et al., "ALPi: A DDoS Defense System for High-Speed Networks," IEEE J. Selected Areas Comm., vol. 24, no. 10, pp. 1864 - 1876, Oct. 2006.
[10] R. Chen, J. Park, and R. Marchany, "A Divide-and-Conquer Strategy for Thwarting Distributed Denial-of-Service Attacks," IEEE Trans. Parallel and Distributed Systems, vol. 18, no. 5, pp. 577- 588, May 2007.
[11] A. Yaar, A. Perrig, and D. Song, "StackPi: New Packet Marking and Filtering Mechanisms for DDoS and IP Spoofing Defense," IEEE J. Selected Areas Comm., vol. 24, no. 10, pp. 1853-1863, Oct. 2006.
[12] A. Bremler-Bar and H. Levy, "Spoofing Prevention Method," Proc.IEEE INFOCOM, pp. 536- 547, 2005.
[13] J. Xu and W. Lee, "Sustaining Availability of Web Services under Distributed Denial of Services Attacks," IEEE Trans. Computers, vol. 52, no. 2, pp. 195-208, Feb. 2003.
[14] W. Feng, E. Kaiser, and A. Luu, "Design and Implementation of Network Puzzles," Proc. IEEE INFOCOM, pp. 2372-2382, 2005.
[15] X. Yang, D. Wetherall, and T. Anderson, "A DoS-Limiting Network Architecture," Proc. ACM SIGCOMM, pp. 241-252, 2005.