DDoS Confirmation & Attack Packet Dropping Algorithm in On-Demand Grid Computing Platform

Muhammad Zakarya, Zahoor Jan, Imtiaz Ullah, Nadia Dilawar *and* Uzm

Abstract- Distributed denial of service (DDoS) attacks on the Internet in general and particularly in Grid computing environment has become a visible issue in computer networks and communications. DDoS attacks are cool to provoke but their uncovering is a very problematic and grim task and therefore, an eye-catching weapon for hackers. DDoS torrents do not have familiar characteristics therefore, currently existing IDS[3] cannot identify and discover these attacks perfectly. Correspondingly, there implementation is a puzzling task. In practice, Gossip based DDoS attacks detection apparatus are used to detect such types of attacks in computer networks, by exchanging stream of traffic over line. Gossip based techniques results in network overcrowding and have upstairs of superfluous and additional packets. Keeping the above drawbacks in mind, we have proposed a DDoS detection and prevention mechanism in that has the attractiveness of being easy to adapt and more trustworthy than existing counterparts. We have introduced entropy based detection mechanism for DDoS attack. Our proposed solution has no overhead of extra packets, hence resulting in good QoS[2]. Once DDoS is detected, any prevention technique can be used to prevent DDoS in Grid environment. In this paper we are going to extend our idea. A confirmation mechanism is introduced herewith.

Index Terms: Grid Computing, Packet Dropping, GridSim DDoS.

I. INTRODUCTION

Grid Computing is the application of numerous systems to a single gigantic problem at the same time, habitually to a scientific or technical problem that needs a large number of CPU processing cycles i.e. more CPU power or access to massive and bulky aggregates of data. One of the main Grid Computing strategies is to use diverse soft-wares to divide and distribute different pieces of a single program among several individual systems, may be up to many thousands. These systems, taking part in Grid System are called nodes.

Grids are called super computers for economically poor organizations. The GS4 consists of GN and a GNM5. When multiple GS are combined in such a way, that at least one of them registers its available services to a Broker and others

Grid Sites (GS) requests for such registered services from the Broker. The Environment is called On-Demand Grid Computing Environment, because customers only pay for only those services, they used [1]. Open systems and shared resources increase many security challenges, making safety and protection one of the foremost barriers for implementation of cloud computing technologies [2].

The rest of paper is organized as follows. In section I we give some introduction, II is about related work that we proposed in previous version. Section III sketches the specific solution, architecture and results that were noticed during our simulations, Section IV highlights the problem with our previous solution and V is about new DDoS Attack Confirmation and Packet Dropping Algorithm and proposed solution. VI describes statistical and simulation results. VII is about performance evaluation. We conclude in section VIII with major challenges and some future directions.

Muhammad Zakarya, Zahoor Jan, Imtiaz Ullah, Nadia Dilawar and Uzm, Abdul Wali Khan University, Mardan, Pakistan. mohd.zakarya@awkum.edu.pk. Manuscript received March 9, 2012; revised April 30, and November 19, 2012.

II. RELATED WORK

According to [3], any statements that have some shock and importance are called information. Some believe that information theory is to be a subset of communication theory, but we consider it much more. The word entropy is rented from physics, in which entropy is a measure of the chaos of a group of particles i.e. 2nd law of thermodynamics. If there are a number of possible messages, then each one can be expected to occur after certain fraction of time. This fraction is called the probability of the message. In [4], [5] Shannon proved that information content of a message is inversely related to its probability of occurrence. To summarize, the more unlikely a message is, the more information it contains. In [6], Entropy H(X) is given by

$$H(X) = -\sum_{x\in X}^{m} p(x) log p(x) \qquad (1)$$

The log is to the base 2 and entropy is expressed in bits. To say randomness is directly proportional to entropy i.e. more random they are, more entropy is there. The value of sample entropy lies between 0 and log(n). The entropy value is smaller when the class distribution belongs to only one & same class while entropy value is larger when the class distribution is more even. Therefore, comparing entropy values of some traffic feature to that of another traffic feature provides a mechanism for detecting changes in the randomness. We use traffic distribution like IP address & application port number i.e. (IP address, Port). If we wants to calculate entropy of packets at a single or unique source i.e. destination, then maximum value of n must be 232 for IPV4 address. Similarly, if we want to gauge entropy at multiple application ports then value of n is the total number of ports [7]. In similar way, p(x) where

x ϵ X, is the probability that X takes the value x. We randomly examine X for a fix time window (w), then p(x) = mi/m Where, mi is the total number we examine that X takes value x i.e

$$m = \sum_{i=1}^{n} mi \qquad (2)$$

Putting these values in entropy Equation 1, we get

$$H(X) = \sum_{i=1}^{n} \left({mi}/{m}\right) log\left({mi}/{m}\right) \qquad (3)$$

Similarly, if we want to calculate the probability p(x), then m is the entire number of packets, but mi is the number of packets with value x at destination as source [8]. Mathematically given as

$$\mathrm{P(x)} = \frac{\text{Number of pockets with x, as source(destination)address)}}{\text{Total number of pockets}} \qquad (4)$$

Again if we want to calculate probability p(x) for each destination port, then

$$\mathrm{P(x)} = \frac{\text{Number of pockets with x as source(destination)port}}{\text{Total number of pockets}} \qquad (5)$$

Remember that total number of packets is the number of packets observed in a specific time slot (w). When this calculation finishes, normalized entropy is calculated to get the overall probability of the captured flow in a specific time window (w). Normalized Entropy is given by

$$\text{Normalized entropy} = \left({\mathrm{H}}/{\log n_0}\right) \qquad (6)$$

Where n_o is the number of dissimilar values of x, in a specific time slot (w). During the attack, the attack flow dominates the whole traffic, resulting in decreased normalized entropy. To confirm our attack detection, again we have to calculate the entropy rate i.e. growth of entropy values for random variables, provided that the limit exists, and is given by

$$H(x) = \lim_{n\to\infty} \frac{1}{n} H(x_1, x_2 \ldots \ldots x_n) \qquad (7)$$

III. PROPOSED SOLUTION & RESULTS

In [1] the authors proposed a Grid Architecture and a DDoS detection mechanism that has the beauty of being easy to adapt and more reliable than existing counterparts. The author's claims, that their proposed solution has no overhead of extra packets, hence resulting in good QoS. The architecture is shown in Fig 2. The whole Grid environment is divided into multiple sites either on geographical or administrative base. Every 1GS is under the control of a powerful 2AS. Our 3ADS is installed on every edge router. Our confirmation algorithm needs to be installed on subsequent and attached router to the edge router. Once DDoS is detected at edge router, the flow is transferred to next neighboring router, where again the flow is checked against those information that were collected on edge router. If there is no change the attack is confirmed and the packet is discarded or dropped. Other wise, the packet is thrown to its destination on its way. We will use 4GridSim for simulation of our algorithm.

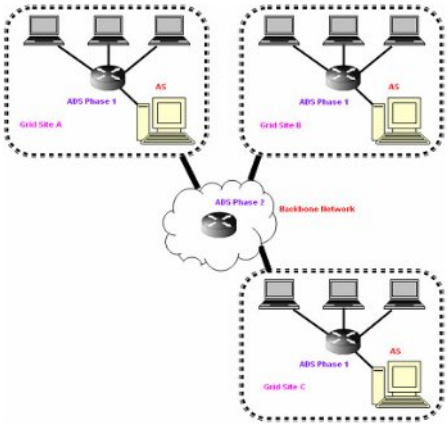

Fig .1. Proposed Grid architecture [1]

GridSim [9, 10, 11] was used for the evaluation of this approach. Results seen are of interest but high network access can lead to false positives. In next section, we are going to propose a confirmation algorithm to limit these false positives. Our 2ADS can detect 100% DDoS attack only in case of good threshold value, which is one of the most challenging tasks in developing any ADS. We conclude our story that a threshold value of 0.95 results in good detection rate. A value greater than 0.95, results in good detection rate i.e. 100 % DDoS detection but generate more false positive alarms, as the value is increased from 0.95 to 1.0.The steps in algorithm are as under. Fig 5 shows the flow diagram of detection algorithm.

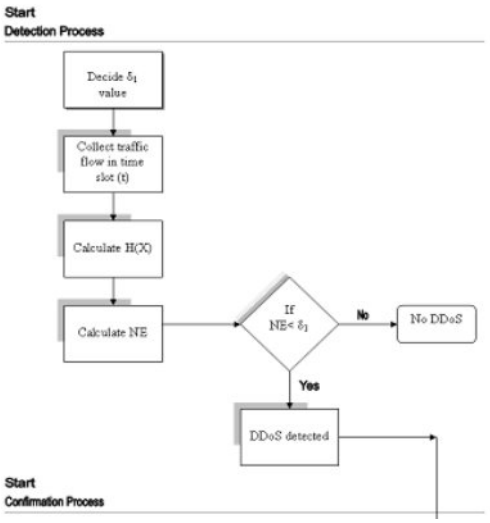

Fig. 2. Flow diagram [1]

IV. EXISTING PROBLEM

We have proposed a DDoS detection and prevention mechanism in [1], that has the beauty of being easy to adapt and more dependable than existing counterparts. As, in service level security issues DoS Attacks, DDoS & Network Overcrowding, are most important. Solving the dispute of DDoS attack also results in network High Availability as well as good QoS. The problem in that solution was that, in huge network usage or congested network flow our algorithm will raise the attack alarm i.e. false positives. But it is not always be the case. To confirm the attack flow and decide to flush out or washout the flow, we are going to propose a confirmation algorithm, in this paper.

V. PROPOSED PACKET DROPPING ALGORITHM

In [1] the authors proposed entropy rate for confirmation of the attack flow, but still no exact solution was proposed. Entropy rate shows the increase or decrease ratio of distribution. We are going to extend our idea in this article and will propose and study a DDoS confirmation algorithm. Based on the results of such a confirmation algorithm the router will decide either to allow the flow of packets or to discard and drop that packet flow. We need such an algorithm because during high network access our DDoS will generate false positives and will alert the next edge router for DDoS attack, but it might not be the case. Our ADS is installed on each edging router. Our affirmation algorithm needs to be installed on consequent and attached router to the edge router. Once DDoS is detected at edge router, the flow is transferred to subsequently adjacent router, where for a second time the flow is checked against those information that were claimed on edge router. If there is no alteration the attack is confirmed and the packet is superfluous one and hence needs to be dropped. Otherwise the packet is thrown to its target node or system on its own way. We will use GridSim for simulation of our algorithm and performance evaluation.

A simple and straightforward solution is to run the same algorithm on receiver side router. But the problem is that we are going to detect and drop the packet flow as early as possible i.e. near the source confirmation. Suppose in Fig 3 above the user ab1 sends 90 packets to cb1, 91 packets to

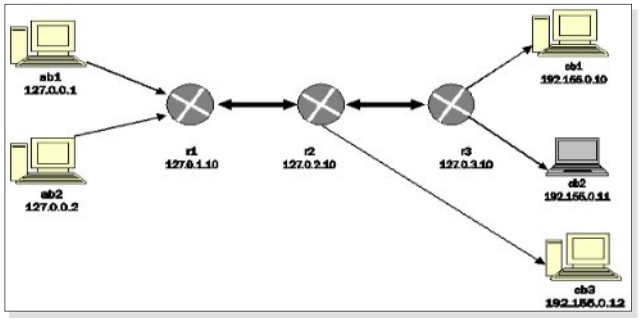

Fig. 3 Confirmation algorithm

cb2 and 34 packets to cb3. When entropy is calculated on r1, the attack is detected. When this flow reaches to r2, the packets that were addressed to cb3 are directed on different way. Again if we calculate entropy of ab1 on r3, no attack is detected. It results in, if we calculate entropy i.e. if we run our detection algorithm two times on edge router to sender and receiver, then to some extent we will accurately measure DDoS and can drop only attack packets.

If the algorithm calculates same values, it means the attack is confirmed otherwise the packets are forwarded to its destination. The problem is that we need to detect and confirm the attack near to the source, so that the bandwidth is not wasted. The goal cannot be achieved in this solution. We can run the same detection algorithm on next edge router but still if the network is so large consisted upon 100 routers. There is the possibility that the attack flow will remain on one path crossing over multiple routers. It will confirm the attack without any concern that in future the flow may be distributed over multiple paths.

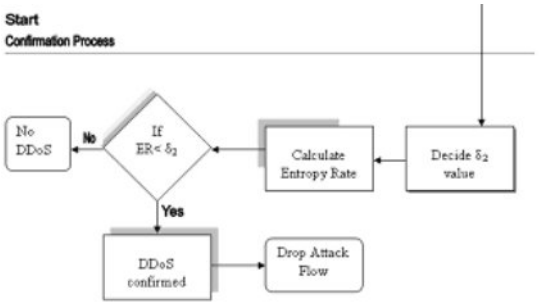

Fig .4. Flow diagram of confirmation algorithm

Following are the steps for confirmation of the DDoS attack.

- Decide a threshold value δ2
- Calculate entropy rate on edge router using Equation VII
- Compare entropy rates on that router, if =< δ2, DDoS confirmed
- Drop the attack flow

VI. SIMULATIONS STUDY & RESULTS

Fig. 5 shows the simulation environment that was created in GridSim Simulator.

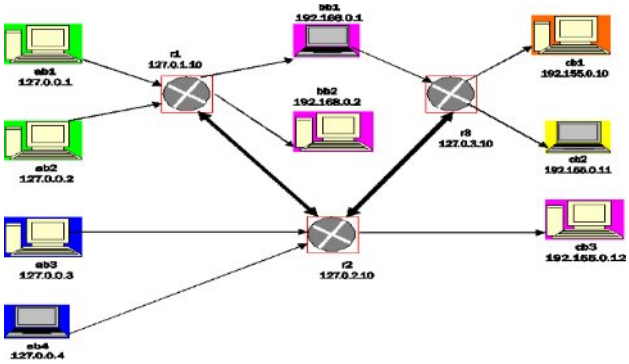

Fig. 5. Simulations study

The above simulation environment was designed and developed in GridSim simulation environment. Routers are connected to each other over a 10 Mbps link (⟷), while all other connections are made at 1 Mbps link (⟷). The reason behind this terminology is clear as router forward more data packets as compared to a single transmitting node.

TABLE 1 TRAFFIC AT ROUTER 1

Source node	Destination node	No of packets	R1	R3	Entropy (R1)	Entropy (R3)
AB1	CB1	20	12	8	0.35	0.27
AB2	CB1	20	4	16	0.17	0.40
AB1	BB1	30	15	15	0.39	0.39
AB2	BB2	40	32	8	0.52	0.28

Traffic entropy at R1 is 0.35 + 0.17 + 0.39 + 0.52 = 1.43

Traffic entropy at R1 is 0.28 + 0.40 + 0.39 + 0.28 = 1.35

Total entropy at ROUTER 1 = 1.43 + 1.35 = 2.78

Normalized entropy = 2.78 / $\log_2$ (8) = 0.93

TABLE 2 TRAFFIC AT ROUTER 2

Source node	Destination node	No of packets	R1	R3	Entropy (R1)	Entropy (R3)
AB3	CB1	10	3	7	0.16	0.29
AB4	CB1	20	11	9	0.37	0.33
AB3	CB3	40	21	19	0.49	0.47
AB4	CB2	20	18	2	0.46	0.12

Traffic entropy at R1 is 0.16 + 0.37 + 0.49 + 0.46 = 1.48

Traffic entropy at R1 is 0.29 + 0.33 + 0.47 + 0.12 = 1.21

Total entropy at ROUTER 1 = 1.48 + 1.21 = 2.69

Normalized Entropy = 2.69 / $\log_2$ (8) = 0.90

Considering a threshold value of δ1 = 0.93 will activate an alarm for DDoS at ROUTER 1. At the edge router i.e. ROUTER 2 the confirmation algorithm with threshold value δ2 = 0.90 will confirm the attack on CB1; hence it will drop the packets flow directed to CB1.

VII. PERFORMANCE EVALUATION

We observed that a threshold value of 0.95 results in good detection rate and a threshold value of 0.90 results in good confirmation. A value greater than 0.95 and 0.90, results in good detection rate and confirmation i.e. 100 % DDoS detection and confirmation, respectively but generate more false positive alarms, as the value is increased from 0.95 to 1.0 i.e. false detection alarm or 0.90 to 1.0 i.e. false confirmation alarm. The reports are shown in Fig. 6 and Fig 7, which are self explanatory. Our experiments show that as more attacks are detected, more attacks are also confirmed and vice versa. In some situations that might not be the case, as its not assured that more network traffic will always cause DDoS. Still the topic needs researcher's attention for further exploration and solutions.

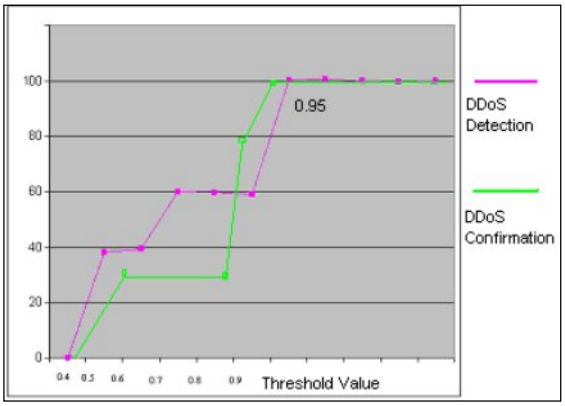

Fig .6. DDoS detection & confirmation rate

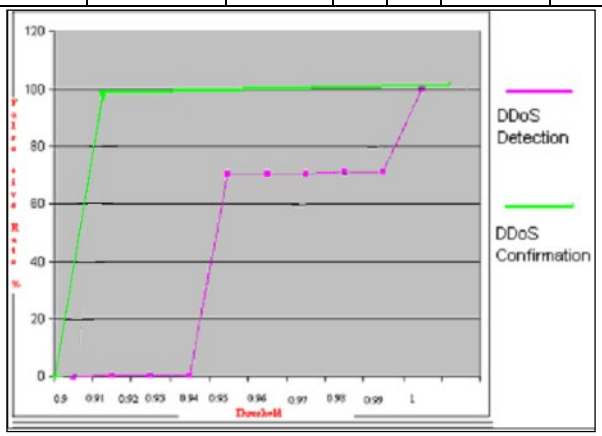

Fig. 7. DDoS false positive rate

IX. CONCLUSION & FUTURE WORK

In this paper, we have proposed a new solution and algorithm to DDoS attack confirmation and attack packet dropping for On-Demand Grid Computing platform. In previous version of this article we introduced an ADS for recognition & early prevention of DDoS attacks in our suggested architecture. The existing problem of huge network access might result in false positive alarms. That issue was subject of this article. Our DDoS attack packet dropping algorithm will confirm the attack flow, if it is an attack flow, the flow is discarded otherwise the flow is considered legitimate data packets and are forwarded to its destination, without any concern that it was targeted as a DDoS attack flow on the edge router. In future the proposed design and suggestion may be actually

implemented over Grid computing platform to precisely detect and confirm DDoS attacks. This idea may also be extended for recovery mechanism for DDoS attacks.

ACKNOWLEDGMENTS

The authors are thankful to student's contributions on behalf of iFuture. The credit also goes to Society for Advancement & Integration of Multiple Sciences (SAIMS), Abdul Wali Khan University, Mardan.

REFERENCES

[1] Muhammad Zakarya, Ayaz Ali Khan, Hameed Hussain, Grid High Availability & Service Security Issues with Solutions, ICIIT 2010, 978-1-4244-813 8-5/10 / $ 26.00 C 2010 IEEE.

[2] Cloud Security Alliance. Top Threats To Cloud Computing. Technical Report, March 2010. http://www.cloudsecurityalliance.org/topthreats.html.

[3] David Applebaum, Probability and Information (An Integrated Approach), Cambridge University Press, 2008.

[4] E. Claude Shannon, A Mathematical Theory of Communication, 1948.

[5] E. Claude Shannon, Communication Theory of Secrecy Systems, 1949.

[6] Thomas M. Cover, Joy A. Thomas, Elements of Information Theory, Second Edition, 2006.

[7] Dennis Arturo Ludeña Romaña, Yasuo Musashi, Entropy Based Analysis of DNS Query Traffic in the Campus Network, Japan.

[8] George Nychis, An Empirical Evaluation of Entropy-based Anomaly Detection, May 2007.

[9] Manzur Murshed, Rajkumar Buyya, Using the GridSim Toolkit for Enabling Grid Computing Education, Monash University, Australia.

[10] Anthony Sulistio, Uros Cibej, Srikumar Venugopal, Borut Robic, Rajkumar Buyya, A toolkit for modelling and simulating data Grids: an extension to GridSim, March 2008.

[11] Anthony Sulistio, Chee Shin Yeo, Rajkumar Buyya, Visual Modeler for Grid Modeling and Simulation (GridSim) Toolkit, 2003.