International Journal of Computer Networks and Communications Security

VOL. 8, NO. 1, January 2020, 1–9
Available online at: www.ijcncs.org
E-ISSN 2308-9830 (Online) / ISSN 2410-0595 (Print)

Study to Validate the Performance of Flooding Based Distributed Denial of Service Attacks

SATWINDER SINGH[1], Er. ABHINAV BHANDARI[2], Dr. KRISHAN KUMAR SALUJA[3], Dr. A.L. SANGAL[4]

[1] Student of M.tech (CE) Regular, UCOE, Punjabi University, Patiala

[2] Assistant Professor, Deptt. Of Computer Engineering, University College of Engineering, Punjabi University Patiala

[3] Associate Professor, S.B.S.C.E.T Ferozpur, India

[4] Professor and Principal, Dr. B. R. Ambedkar NIT, Jalandhar, India

[1]*s.sohi19@gmail.com*, [2]*bhandarinitj@gmail.com*, [3]*k.saluja@rediffmail.com*, [4]*sangal62@yahoo.com*

ABSTRACT

Network technology has experienced explosive growth in the past decades. The generally accepted viewpoint in the security world is that no system or network is totally protected which makes network security an important concern. The work done in this paper focuses on Distributed Denial of Service Attacks (DDoS) where legitimate users are prevented from accessing network services. Distributed Denial of Service (DDoS) Attacks has been increasingly found to be disturbing the normal working of organizations causing billions of rupees of losses. Organizations are trying their best to reduce their losses from these systems. The focus of this paper is to providing our results of experiments in this regard for flooding based DDoS attacking in the simulation environment with NS2.35. This paper point out that the bandwidth may be more easily flooded by UDP-type attacking than by TCP-type one at different attack strength. In this paper we done the validation of DropTail and RED queue under the flooding based DDoS attack, which is better for the legitimate user under the flooding attack.

Keywords: *DoS, DDoS, TCP, UDP, DropTail, RED.*

1 INTRODUCTION

In the field of computer security, Denial of Service (DoS) attacks is certainly a very serious problem in the Internet, whose impact has been well confirmed in the computer network literature. The main aim of DoS attack is the confusion of services by attempting to bound access to a machine or service instead of subverting the service itself. This kind of attack goals at depiction a network incapable of providing normal service by targeting either the network's bandwidth or computer assets. These attacks accomplish their goal by sending at a victim a stream of packets that swamps his network or processing capacity denying access to his usual customers. In the not so distant past, there have been some large-scale attacks targeting most common Internet sites [1–3]. Distributed Denial of Service (DDoS), is a comparatively simple, yet very powerful method to attack Internet assets. DDoS attacks add the many-to-one feature to the DoS problem making the prevention and improvement of such attacks more difficult and the impact proportionally severe.

DDoS attacks are comprised of packet streams from dissimilar sources. These attacks connect the power of a vast number of coordinated Internet hosts to consume some critical resource at the target and deny the service to legitimate clients. Traffic of DDos attack is behave like a regular legitimate traffic. The traffic is typically so aggregated that it is tricky to distinguish legitimate packets from attack packets. More considerably, the attack amount can be larger than the system can handle. Unless particular care is taken, a DDoS victim can suffer from damages ranging from system shutdown and file fraud, to total or partial beating of services. Attackers always modify their tools to bypass security systems developed by system managers and researchers, who are in a constant ready to modify their approaches to handle new attacks.

2 HISTORY OF DOS

A. The Morris Worm

On November 2 1988 the first DoS attack was launched on the electronic world. As a result about 15% (about 6.000) of the systems connected to the network were infected and blocked running. Morris Worm was self replicating and self propagating.

B. SYN Floods

SYN Floods have existed as TCP has existed. They are a straight outcome of TCP specifications. It is consequently possible to say that SYN Floods are part of TCP just as spoofing is part of UDP. Trouble-free to implement, effective and tough to traceback to the actual source, Denial of Service attacks are still valued by malicious Internet abusers, running to launch 100's of megabits, occasionally more than one gigabit, of SYNs targeted to a lone service.

3 DDOS ATTACK

DDoS (Distributed Denial-of-Service) attacks still are one of the most critical attacking means and the sources of mass trouble on internet. DDoS attacks typically occur when a large number of internet packets from compromised hosts (zombies) overflow the bandwidth or resources of a single target (victim) and the flood of incoming messages to the victim essentially forces it to respond so slowly as to be rendered effectively unavailable and even to shut down [4]. Distributed denial-of-service (DDoS) attacks are simply denial-of-service attacks performed from multiple subverted machines (agents). In the straw man and most frequently used scenario, all machines are engaged simultaneously and start generating as many packets as they can toward the victim. A large number of 10 participating agents enable the attacker to overload resources of very highly provisioned victims, with reserved capabilities of agent machines. Figure 1: depicts a simple distributed denial-of-service attack scenario in which attacking machines A and B send streams of malicious packets to victim V, denying its tune to legitimate clients C1 and C2.

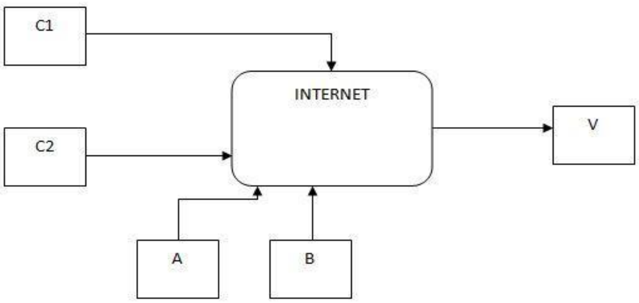

Fig. 1. This figure shows the attack scenario of DDoS.

4 DDOS CHARACERISTIC

There are numerous features of DDoS attacks that obstruct their successful detection and defense:

- DDoS attacks generate a large volume flow to overwhelm the target host. The victim cannot shield itself even if it detects this event. So the detection and defense of DDoS should ideally be near the source of the attack or somewhere in the network.
- It is difficult to distinguish attack packets from legitimate packets. Attack packets can be identical to genuine packets, since the attacker only needs volume, not content, to inflict damage. in addition, the volume of packets from individual sources can be low enough to escape notice by restricted administrators. Thus, an exposure scheme based on a single site will have either high positive or high negative rates.
- DDoS traffic generated by available tools often has identifying uniqueness, making the result based on figures study possible. Though, given the inherently busty nature of Internet, detecting DDoS attacks is slip prone.

5 DDOS ARCHITECTURE

A Distributed Denial of Service Attack is made of four elements.
• The real attacker.
• The handler or master, which is compromised [28] hosts with a special program running on them, capable of controlling multiple agents.
• The attack daemon agents or zombie hosts, who are compromised hosts that are successively a special program and are responsible for generating

a flow of packets towards the projected victim. Those machines are commonly external to the victim's personal network, to avoid capable response from the victim, and outdoor to the network of the attacker, to keep away from liability if the attack is traced back.

- A victim or target host. [5]

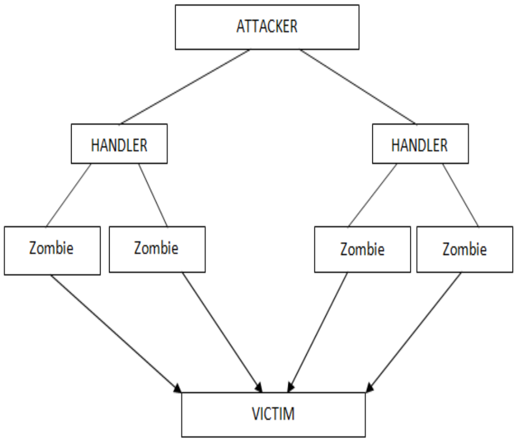

Fig. 2. Architecture of a DDoS Attack [6]

A typical DDoS attack process can be described as follows. An attacker first scans a large range of networks to find vulnerable hosts that have weak defenses against a hateful intrusion. The number of these hosts is determined by the strength of the attack that an attacker intends to initiate. Next, the attacker installs "Master" or "Agent" programs on these vulnerable hosts. A machine with an "Agent" program is called a "Zombie", which carries out the authentic attack. A machine installed with a "Master" program is able to communicate with a number of "Zombies" and serves as a control-handler of the attacker. An attacker can command several "Masters" directly, and "Zombies" are activated by these "Masters" at the designated time for an attack. Figure 2 shows this three-layer control. The reason for using such architecture is to keep the attacker safe and difficult to trace. The attacker has to wait for an appropriate time to launch his DDoS attack. When a defensive server suspects that it is underneath a DoS attack, it can only find numerous genuine connection requests received from a large number of legitimate IP addresses, intense all the resources of the server. However, the real owners of these "Zombies" are unwitting [27] accomplices, and do not know what has actually happened on their machines.

6 DDOS ATTACK CLASSIFICATION

A. Trinoo

Trinoo [7] is a bandwidth depletion attack tool that can be used UDP flood attacks to halt the victim. Trinoo [8] is the first DDoS attack tool to be widely distributed and used. It consists of an attacker system, several compromised systems, which include one or more masters (referred to as handlers), one or more daemon systems (referred to as agents), and one or more victims. The attack begins by loading the Trinoo program on one or more compromised systems. These systems do something as handlers and agents. The agents send a UDP packet to let the handler know that the agent systems are ready. When the attack system sends the attack authority, the handler sends a message to the agents to begin the attack. After receiving the command to launch an attack, the agent sends a UDP overflow to arbitrary port [26] numbers on the victim. This attack was experienced in 1999 by University of Massachusetts.

B. Tribe Flood Network

Tribe Flood Network (TFN) [9], printed in 1999, is a DDoS attack tool so as to provides the attacker with the ability to earnings both bandwidth depletion and resource reduction attacks. It uses a command line interface to converse between the attacker and the control master program but offers no encryption between agents and handlers or between handlers and the attacker. In adding up to Trinoo's UDP flooding it also allows TCP SYN and ICMP flood as well as smurf attacks. Handlers are accessed using usual TCP links like telnet. Other alternatives are ICMP tunnelling tools like LOKI [10, 11]. Communication between the handler and the daemons is expert with ICMP ECHO REPLY packets, which are more difficult to detect than UDP packets and can often pass firewall systems. TFN launches synchronized Denial of Service attacks that are especially difficult to counter as it can generate multiple types of attacks and it can generate packets with spoofed source IP addresses and also randomize the mark ports. It is capable of spoofing either one or all 32 bits of the IP source address, or just the very previous 8 bits. Some of the attacks that can be launched by TFN include: Smurf, UDP overflow, TCP SYN overflow, ICMP echo request overflow, and ICMP bound for broadcast.

C. TFN2K

The TFN2K [12] is a DDoS attack tool based on the TFN structural design. The TFN2K attack tool adds encrypted messaging between all of the attack components [13]. Targets are attacked via UDP, ICMP_ECHO flood, TCP SYN or smurf attack, and the attack type can be varied throughout the attack. Instructions are sent from the master to the agent via TCP, ICMP, UDP, or all 3 at random, making it harder to detect TFN2K by scanning the system.

D. Mstream

The mstream [14] tool uses spoofed TCP packets with the ACK flag set to attack the target. Statement is not encrypted and is performed as of end to end by using TCP and UDP packets. Access to the handler is secret code protected. This program has a feature not found in other DDoS tools. It informs all attached clients of access, successful or not, to the handler(s) by competing parties.

E.Shaft

Shaft [15] uses TCP, ICMP or UDP flood to carry out the attack, and it can install all 3 styles concurrently. UDP is used for communication between handlers and agents, and messages are not encrypted. Shaft randomizes the source IP address and the source port in packets. The packets size ruins set all over the attack. A new characteristic is the ability to change the handler's IP address and port during the attack.

F. Code Red

The Code Red [16] worm is self-propagating malicious code that exploits a known weakness in Microsoft IIS servers for transmission. It achieves a coordinated attack by preprogramming the onset and abort time of the attack, attack method and target addresses (i.e., no handler/agent architecture is involved).

G. Stacheldraht

Stacheldraht [17] (German term for ‘‘barbed wire’’) is based on early versions of TFN. Stacheldraht is a combination of Trinoo and TFN attack and relies on TCP for transport. It also has the ability to perform updates on the agents without human intervention (mechanically). This way that the attacker can provide the installation file on an nameless server and when each agent system turns on (or logs on to the Internet), the mediator will mechanically look for updates and install them. The handlers and agents occasionally exchange ICMP reply packets. It encrypts the announcement between the attacker and the masters and performs automated update of the agents. It can implement Smurf, SYN overflow, ICMP flood and UDP flood attacks.

7 SIMULATION RESULTS AND DISCUSSIONS

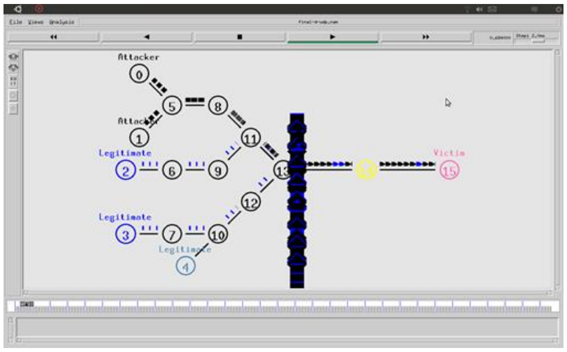

Fig. 3. Attacking Structure for simulation

Experiment is performed in NS2.35 Simulation environment. The experimental platform is with Sony PC, the system of core linux10.0, the CPU of Intel Centrino 1.83GHz, hard disk of 200GB and integrated network card. There are 16 nodes in the network from 0 to 15. Each node represents a network system in the internet. In the network nodes for host computers are 0, 1, 2, 3, 4 and nodes for the routing systems are 5, 8, 11, 6, 9, 7, 10, 12, 13, 14, and 15. In this simulation structure node 0 and 1 represented as attackers while node 2, 3, and 4 represented as legitimate client. Node 15 is a target node. In the routing nodes we can set algorithms for the routing, size of the buffer and so on. In the network links, we can set bandwidth for the link, delay and so on. In the host nodes, we can set the algorithms for the queuing, delay, buffer size and etc.

For the discussions of result, we use the simple network model of Fig. 3 as that shown in Fig. 4.

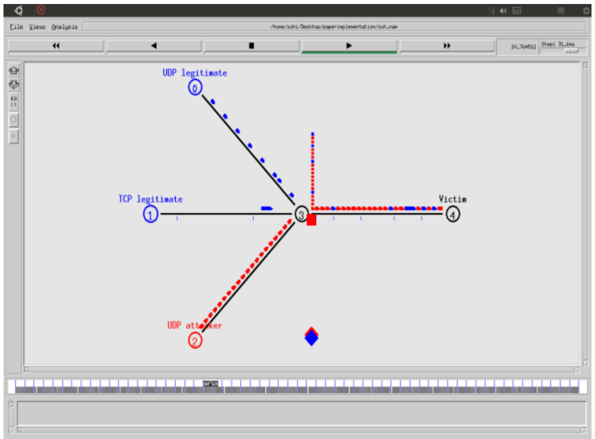

Fig. 4. Simple Attacking Structure for simulation

Table 1: Parameters of simulation

No. of Node	No. of Legitimate user	No. of attacker	No. of links used	Bandwidth used in the links	Delay in the links
5	2	1	4	1Mb	100ms

In the simplified network structure the link bandwidth for node 0, 1, 2, 3, 4 is 1Mb. The Nodes 0 and 1 marked as UDP legitimate client and the TCP legitimate client respectively. The node 2 represents the attacker, node 3 for the router and node 4 for the target denoted by victim.

Traffic flow sent from source node is characterized by the bandwidth. For case, node 0 sends 28% information towards node 4 that is target node means node 0 occupies 0.28 Mb bandwidth. Therefore parallel, node 1 sends 40% data to node 4, and nose 2 sends data 58% to target node, the total exceeds 26% of the bottleneck link bandwidth between node 3 and node 4. In this case, some packets from the legitimate clients may be dropped at node 3. As a result, node 2 achieves its goal of the flooding attack towards the victim node means node 4.

8 DISCRIPTIONS OF FLOODING ATTACK

These days near about 80% of the internet data flow is based on TCP protocol [18]. There are many types of attacks [19-25]. In this section, we check the behavior of legitimate traffic under the flooding attack namely, TCP and UDP. At starting we evaluate the attack free traffic. Then, under the different attack strength with either TCP traffic or UDP one.

1.1 Test 1: attack free traffic

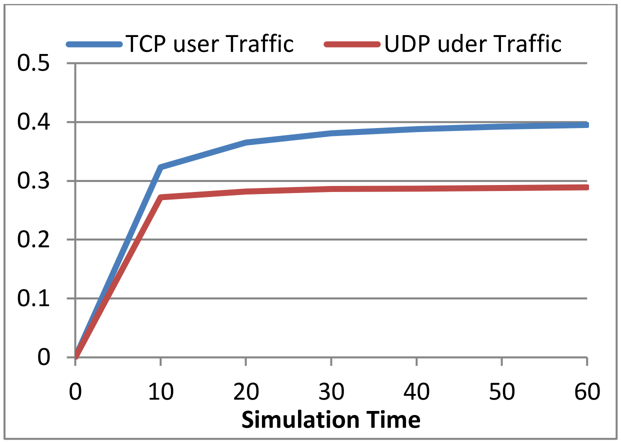

Fig. 5. Attack Free Traffic

With the help of Simplified attack model in Fig 4, node 0 sends 28% of the UDP traffic towards node 4 and node 1 sends 40% TCP data flow to node 4 respectively. Node 2 sends none. Hence, attack free. Nodes 0 and 1 occupy only 68% of the total bandwidth, so there is no congestion. Abovegraph shows the results throughput in Mbps Vs simulation time in sec.

1.2 TEST 2: TCP and UDP user traffic under UDP-type attacking

With the attack model of Fig 4.2: node 0 sends 28% UDP traffic to node 4, node 1 sends 40% of TCP traffic to node 4. In this experiment, node 2 sends 58% UDP data flow, we can say that it produces the attack strength of 26% of the UDP type. In simulation, attacker sends the attack traffic after the 20 sec of the simulation period. The datafrom node 3 to node 4 exceeds 26% of the bandwidth. In this case, some packets of the legitimate may be dropped at node 3. Below graph shows the simulation results.

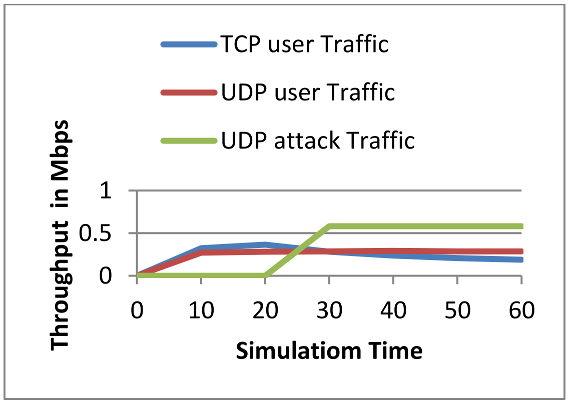

Fig. 6. Traffic under 26% of UDP-type attack

1.3 TEST 3: TCP and UDP user traffic under TCP-type attacking

With the attack model of Fig 4: node 0 sends 28% of the UDP data flow towards target node 4, node 1sends 40% of TCP data flow to node 4 and node 2 sends 58% of TCP attack traffic to node 4, concurrently. In this case, traffic from bottleneck link exceeds 26% of the total bandwidth. Accordingly, some packets from legitimate client may be dropped at node 3. In this case, we can say that node 2 produces attack strength of 26% of the TCP-type. Below graph shows the results throughput in Mbps Vs simulation time.

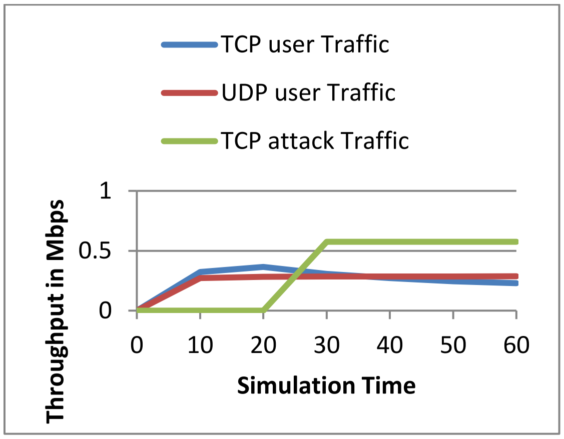

Fig. 7. Traffic under 26% of TCP-type attack

1.4 Comments on TEST 8.1-8.3

From test 1-3, we conclude that 26% attack strength of TCP type does not affect UDP user traffic because it requests only little bit service from the server but it reduce the legitimate traffic of the TCP user. But on the other hand, 26% attack strength of UDP type reduce the TCP user traffic more than TCP type attacking and does not affect the UDP user service

9 LEGITIMATE BEHAVIOURS UNDER DIFFERENT ATTACK STRENGTH

1.1 TCP and UDP user traffic under 36% of UDP-type attacking.

In this case, node 0 and node 1 sends 28% UDP data, 40% TCP data to target node 4 respectively. Node 2 sends 68% UDP type attack traffic to target node 4, we can say that node 2 produces attack strength of 36%. Below graph shows the simulation results.

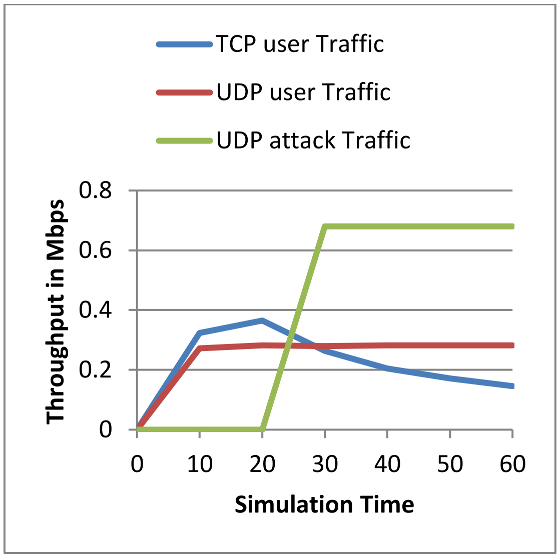

Figure 8: Traffic under 36% of UDP-type attack

1.2 TCP and UDP user traffic under 46% of UDP-type attacking

In this case, node 0 and node 1 sends 28% UDP data, 40% TCP data to target node 4 respectively. Node 2 sends 78% UDP type attack data to target node 4, we can say that node 2 generates attack strength of 46%.

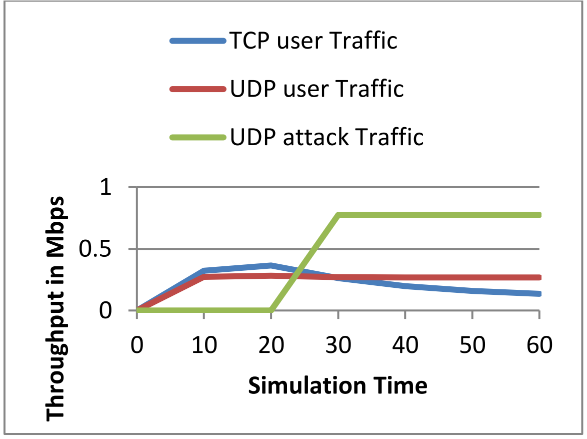

Fig. 9. Traffic under 46% of UDP-type attack

1.3 TCP and UDP user traffic under 36% of TCP-type attacking

In this case, node 0 and node 1 sends 28% UDP data, 40% TCP data to target node 4 respectively. Node 2 sends 68% TCP type attack traffic to target node 4, we can say that node 2 produces attack strength of 36%. Below graph shows the simulation results.

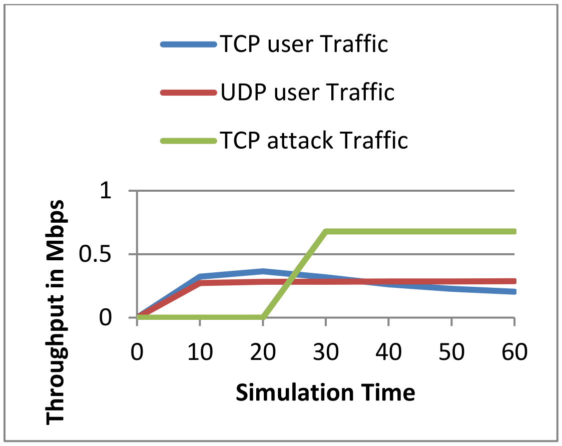

Fig. 10. Traffic under 36% of TCP-type attack

1.4 Experiment of TCP and UDP user traffic under 46% of TCP-type attacking

In this case, node 0 and node 1 sends 28% UDP data, 40% TCP data to target node 4 respectively. Node 2 sends 78% TCP type attack traffic to target node 4, we can say that node 2 produces attack strength of 46%.

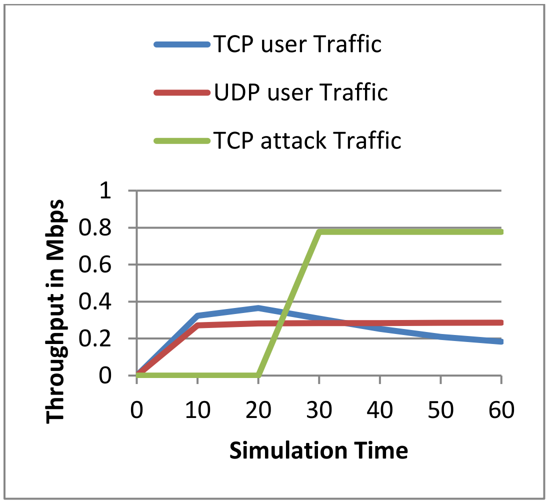

Fig. 11. Traffic under 46% of TCP-type attack

10 EXPERIMENT TO VALIDATE TCP AND UDP USER TRAFFIC USING DROPTAIL AND RED ALGORITHM UNDER TCP-TYPE ATTACKING

In this case, first we check the performance of TCP and UDP user traffic using the DropTail algorithm, after that using RED algorithm. With the attack model of Fig 4.2: node 0 sends 28% UDP traffic to node 4, node 1 sends 40% of TCP traffic to node 4 using DropTail algorithm. Node 0 sends 28% UDP traffic to node 4, node 1 sends 40% of TCP traffic to node 4 using RED algorithm. In this experiment, node 2 sends 58% TCP data flow, we can say that it produces the attack strength of 26% of the TCP type. The traffic from node 3 to node 4 exceeds 26% of the bandwidth. In this case, some packets of the legitimate may be dropped at node 3.

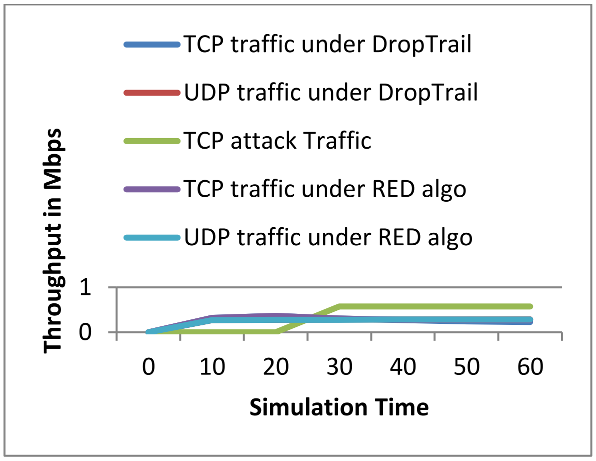

Fig. 12. Traffic using RED and DropTail algo under TCP attack

11 EXPERIMENT TO VALIDATE TCP AND UDP USER TRAFFIC USING DROPTAIL AND RED ALGORITHM UNDER UDP-TYPE ATTACKING

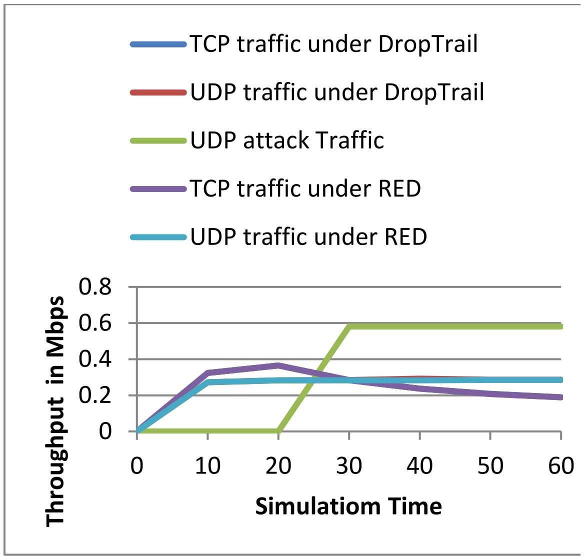

Fig. 13. Traffic using RED and DropTail algo under UDP attack

12 CONCLUSION

The work done in this paper focuses on Distributed Denial of Service Attacks (DDoS) where legitimate users are prevented from accessing network services. Distributed Denial of Service (DDoS) Attacks has been increasingly found to be disturbing the normal working of organizations causing billions of rupees of losses. From the results it is concluded that 28 % attack strength of TCP type does not affect UDP clients but it reduce the TCP traffic of the legitimate client. On the other hand, 28 % attack strength of UDP type does not affect UDP traffic of the legitimate client but it reduce the more TCP traffic of the legitimate client as compared to TCP type attacking. We can say that the bandwidth may be more easily Flooded by UDP-type attacking than by TCP-type one. UDP type attack is more powerful as compared to TCP type attack.

It is also suggested that RED active queue management enhances the performance of TCP and UDP traffic of the legitimate client than the DropTail queue under the TCP type attacking. So RED queue is better as compared DropTail in case when DDoS attack occurs in the network.

13 REFERENCES

[1] CERT Coordination Center, Denial of Service attacks, Available from <http://www.cert.org/tech_tips/denial_of_service.htl.

[2] Computer Security Institute and Federal Bureau of Investigation, CSI/FBI Computer crime and security survey 2001, CSI, March 2001, Available from <http://www.gocsi.com>.

[3] D. Moore, G. Voelker, S. Savage, Inferring Internet Denial of Service activity, in: Proceedings of the USENIX Security

[4] Le Li, Wanlei Zhou, Ping Li, Jing Hai and Jianwen Liu, "Distinguishing DDoS Attacks From Flash Crowds", Third International Conference On Network Systems Security, 2009.

[5] Christos Douligeris, Aikaterini Mitrokotsa," DDoS attacks and defense mechanisms: classification and state-of-the-art", Elsevier, Received 9 October 2003; accepted 13 October 2003.

[6] Cs3 Inc., "What is DDoS and How It is Hurting the Internet, E-commerce and Business", http://www.cs3-inc.com/pk_whatisddos.html, Retrieved 01/19/09.

[7] D. Dittrich, The DoS Project_s ''trinoo'' Distributed Denial of Service attack tool, University of Washington, October 21, 1999.

[8] P.J. Criscuolo, Distributed Denial of Service Trin00, Tribe Flood Network, Tribe Flood Network 2000, and Stacheldraht CIAC-2319, Department of Energy Computer Incident Advisory (CIAC), UCRL-ID-136939, Rev. 1, Lawrence Livermore National Laboratory, February 14, 2000.

[9] D. Dittrich, the Tribe Flood Network Distributed Denial of Service attack tool, University of Washington, October 21, 1999, Available from <http://staff.washington.edu/dittrich/misc/trinoo.analysis.txt>.

[10] Phrack Magazine 7 (49), File 06 of 16 [Project LOKI], Available from <http://www.phrack.com/search.phtml?View&article¼p49-6>.

[11] Phrack Magazine 7 (51) September 01, 1997, article 06 of 17 [LOKI2 (the implementation)], Available from <http://www.phrack.com/search.phtml?

[12] J. Barlow, W. Thrower, TFN2K—an analysis, 2000.

[13] CERT Coordination Center, Center Advisory CA-1999-17 Denial of Service tools.

[14] D. Dittrich, "The 'mstream' distributed denial of service attack tool," http://staff.washington.edu/dittrich/misc/mstream.analysis.txt

[15] S.Dietrich, N. Long and D. Dittrich, "An Analysis of the "Shaft" distributed denial of service tool," http://www.adelphi.edu/~spock/shaft_analysis.txt.

[16] CERT Coordination Center, "CERT Advisory CA-2001-19 'Code Red' Worm Exploiting Buffer Overflow In IIS Indexing Service DLL," http://www.cert.org/advisories/CA-2001-19.html.

[17] D. Dittrich, The _Stacheldraht_ Distributed Denial of Service attack tool, University of Washington, December 1999.

[18] J. Markovic, J. Martin, and L. Reiher, "A taxonomyof DDoS attack and DDoS defense mechanisms,"ACM SigComm Computer Communication Review,Vol. 34, No. 2, 2004, pp. 39-53.

[19] S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," Computers & Security, Vol. 24, No. 4, 2005, pp. 295-307.

[20] S. H. Oh and W. S. Lee, "An anomaly intrusion detection method by clustering normal user behavior," Computers & Security, Vol. 22, No. 7, 2003, pp. 596-612.

[21] A. Lakhina, M. Crovella, and C. Diot, "Characterization of network-wide anomalies in traffic flows," Internet Measurement Conference 2004, Oct. 2004, pp. 201-206, Italy.

[22] P. Barford and D. Plonka, "Characteristics of network traffic flow anomalies," ACM SIGCOMM Internet Measurement Workshop 2001, Nov. 2001, San Francisco, USA.

[23] J. McHugh, "Testing intrusion detection systems: a critique of the 1988 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln laboratory," ACM Trans. Information System Security, Vol. 3, No. 4, 2000, pp. 262-294.

[24] J. W. Haines, L. M. Rossey, R. Lippmann, and R. K. Cunningharm, "Extending the DARPA off-line intrusion detection evaluations," DARPA Information Survivability Conference and Exposition II, vol. 1, IEEE, June 2001, 77-88, Anaheim, California.

[25] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," Computer Networks, Vol. 34, No. 4, 2000, pp. 579-595.

[26] http://support.novell.com/techcenter/articles/nc2000_04c.html

[27] F. Kargl, J. Maier, and M. Weber. Protecting web server from distributed denial os service attacks. In preceedings of the 10thinternational WWW conference, Hong Kong, May 1-5, 2001.

[28] S. Singh, Review on PPM a traceback technique.