Realtime Network IP Traceback Mechanism against DDOS attacks

Sailakshmi Samudrala, S D Vara Prasad
Dept. of Computer Science & Engineering, GITAM University, Hyderabad, India
Assistant Professor, Dept. of Computer Science & Engineering, GITAM University, Hyderabad, India

Abstract:

Migration of DDOS attack is one the biggest challenge on the internet and lan networks. Many techniques has been proposed , including DDOS prevention, ipspoof etc., each of them has advantages and disadvantages. Source IP spoofing attacks are critical issues to the Internet.There has been active research on IP traceback technologies. However, the traceback from an end victim host to an end spoofing host has never yet been achieved, because of the insufficient traceback probes installed on each routing path. Recently a great number of prevention mechanism of a given detection and prevention have been developed, but it is difficult to trace the source ip of the attacker due to misconfiguration of network router or dynamic changes in the network. Also traditional methods are completely depends on network router. In order to overcome these problems, an improved iptraceback mechanism is introduced. Existing approach detect ip using offline CAIDA dataset which isn't suitable to dynamic networks. Experimental results show, proposed scheme performs well compare to traditional methods in terms of detection rate and time to detect source ip.

Keywords –DDOS, IP-Traceback, NIC.

I. INTRODUCTION

The internet rapidly develops on the past few years and significantly influences increasingly industry and business services. When popularity of the broadband, increases ,more high speed networks are linked to the web. Therefore, the difficulties of network security are believed. Currently, the primary threats of network security are coming from hacker intrusion; deny of service (DoS), malicious program, worm, spam, malicious code and sniffer as there are several weaknesses within the original design of IPv4. The common weakness is the idea that attackers could send IP spoofing packets which is he wishes to attack. To put it differently, the attackers modify the IP beginning with the true an individual to another IP field. If these IPs are randomly generated then it is an effort to trace the source of attacks from victims. Besides, the cunning attackers would not ever directly attack the targets. They will construct the botnet first after which organize them to attack the targets. However, it raises the damage measure of attack and tracing the attacks could well be more difficult. As a matter of fact, we could morally persuade the attackers or punish them by law once we find the source of attacks. The procedure of meeting source is known as IP traceback. There are various practices trace attack source with the assistance of routers.

In computer terminology, a network forensic is basically used to detect, deflect, along with illegal network attempts at unauthorized utilization of information systems. Generally it consists of a working laptop or computer, data, or a network site that appears to get portion of a network, but is really isolated and monitored, and which seems to contain information or possibly a resource of value to attackers.

Due to ever growing line speed and Internet traffic amount, measurement of network traffic generates an enormous volume of data introducing scalability issues in both of the storage and processing. Traffic data comprises the moment and duration of a communication, the detailed shape of the communication streams, the identities of a given parties communicating, plus their location.

Packet sniffer is a program running in a network attached device that passively receives all data link layer frames passing within the device's network adapter. It is also known as Network or Protocol Analyzer or Ethernet Sniffer. The packet sniffer captures the data that really is addressed to other machines, saving it for later analysis. It could be used legitimately using a network or system administrator to monitor and troubleshoot network traffic. Using the information captured via the packet sniffer an administrator can identify erroneous packets and utilize the comprehensive data to pinpoint bottlenecks and help maintain efficient network data transmission. Packet Sniffers were never generated to hack or steal information. Had an alternative goal, in order to make things secure.

Every time a packet is received by a NIC, it first compares the MAC address of one's packet to its own. In the event the MAC address matches, it accepts the packet otherwise filters it. This is because of the network card discarding many of the packets that don t contain its own MAC address, an operation mode called non promiscuous, generally indicates that each network card is minding its own business and reading exclusively the frames led to it. In an effort to capture the packets, NIC ought to be beginning in the promiscuous mode. Packet sniffers which do sniffing by setting the NIC card singularly system to promiscuous mode, and thus receives all packets even they are not intended for

it. So, packet sniffer captures the packets by setting the NIC card into promiscuous mode.

II. LITERATURE SURVEY

S.Saurabh and SaiRam[1] proposed packet marking and IP traceback mechanism called Linear Packet Marking which needs large choice of packets, almost total choice of hops traversed from the packet. Other IP traceback algorithm requires much high large number of packets in relation to this algorithm. A large amount of them requires packets according to the scale associated with a significantly large number packets. Yet as this scheme is able to do IP traceback using a lot of packets, it could be highly scalable i.e. it might working for highly DDoS attack involving an exceptionally great many attackers distributed across network. Secondly it probably could be utilized to low rate DoS attacks that might perform attack with very less number packets. This framework is provided with the capacity to be incorporated by other traceback algorithms to scale back the volume of packets needed for path reconstruction that could improve their performance too.

In [2-3], Y. Kim et al. propose a path signature(PS) based victim-end defense system. The internal system requires all routers to flip selected bits within the IP identification field for all those incoming packets. Dependent on these marking bits, a special PS might be generated for all those packets from the same location. With the victim end, the defense system separates traffic in accordance to PS for every packet and detects DDoS attacks by monitoring anomalous changes of traffic amount from a PS. Then, a rate limit value will be set up within this traffic. However, it is hard to detect DDoS attacks if PS diversity is quite bigger than real router diversity of incoming traffic. Moreover, it is quite likely that a PS has been changed after an attack has been detected. For this situation, collateral damage regarding the legitimate traffic couldn't be avoided[2-4].

Limitations:

This procedure requires the attack to remain alive while performing traceback.Secondly IP traceback itself causes DoS attack while performing traceback.The proposed solution is not going to handle packets headers of IPV6 but generated extra traffic for traceback. Unfortunately current proposals for IP traceback mechanism has problems with various drawbacks like need for lots and lots of packets for performing traceback and the in-ability to handle highly distributed and scaled DDoS attacks. A spoofing DDoS attack could make the flow-based rate limit algorithm ineffective.

Ninglu and Yulongwang[2] proposed as Tracing the paths of IP packets returning to their origins, often known as IP traceback serves as a crucial improve defending against Denial of Service (DoS) attacks by employing IP spoofing.In log-based single-packet IP traceback, the path data is logged at routers. Packets are recorded through routers toward the path toward the destination.

Probabilistic Packet Marking:[3] It may be defined as being most widely known packet identification techniques. Within this particular methods, the packets are marked in the router's in which the packet is being transmitted. Marking the packets making use of router's address is the best possible approach compared directly onto the two alternatives provided here, where if a packet dissipates of affected with any attack, the original source router address can easily be fetched and send back into the real router. Now the router checks the packets and retransmits the packet towards the actual destination.

In an effort to react effectively against DDoS attack, all the processes for any information gathering, analysis and defense rule generation require being automated. Furthermore, dependent on these analysis results attack detection and prevention processes also have to be automated. Within this position, lots of information just might be gathered, so in the information zombie PCs, servers and agent distribution systems also need to be detected. Beyond current visualization tools, the law states that it is matured be able to show the network traffic and security status in real-time[4-6].

LIMITATIONS

Bandwidth overhead is amazingly high while tracing the attack origin.It might not trace the attack while it is over i.e attack should remain active until trace time is finished.

III. PROPOSED WORK

WinPcap can be an open source library for packet capture and network analysis for your Win32 platforms. Most networking applications access the network through widely used operating system primitives an example would be sockets. You can easily access data on the network using this approach since the operating system copes with the

low level details (protocol handling, packet reassembly, etc.) and provides a well-known interface that really is similar to the one used to construct grammar files.

Sometimes, however, the 'easy way' is not about the task, since some applications require direct access to packets upon the network. That is undoubtedly, they should get admittance to the \"raw\" data upon the network without having the interposition of protocol processing from the operating system.

The aim of WinPcap would be to give this type of access to Win32 applications; it provides facilities to: capture raw packets, both the ones destined to the machine where it's running plus the ones exchanged by other hosts (on shared media)filter the packets in accordance with user-specified rules before dispatching each of them the application transmit raw packets onto the network gather statistical particulars on the network traffic.This variety of capabilities is obtained by means of a device driver, that is installed inside the networking small portion Win32 kernels, plus a several DLLs.

For any protocol, an Info-packet contains fields that are common to every protocol and also fields that are protocol specific. Common fields in an Info-packet include the packet header size, the packet size (data + packet header). Protocol specific fields for IP, for example, include source and destination IP addresses, while for TCP, for example, they include the source and destination ports, see table . Info-packets mainly contain integers and strings. These data types are easier to manipulate and so making the task of packet processing lighter

.

Protocol: TCP
Total length.:1200
Encap. Protocol: 12
Version: FOUR
Data length.: 1500
Time To Live: 240
IP Source: 172.16.16.5
IP Destination: 172.16.16.3
Header length: 25

Algorithm to network packet analysis

Step 1: open the interface
Step 2:Get of all network interfaces in NetworkInterface[]
Step 3: Get each Network_Interface_name and its MAC addresses in the NetworkInterface[]
Step 4: Choose appropriate NetworkInterface to capture packets in promiscuous mode.
Step 5: Set number of Packets to capture. (Infinite - 1)
Step 6: Start capturing packets

- for each packet pack
  a) set filter=‘ TCP or IP’
  b) temp[]=capturesetfilter(filter)
  c) if(temp[]==‘TCP’)
  d) store pack dest port, seq, src port, syn to DB
  else
  e)store identifier(v4.0), dest port, src port, sync to DB.

Setp 7: Sort DB according to sequence number in the TCP table.
Step 8: Sort the DB according to IP addresses.
Step 9: End
Step 10: Print the packets in the console.
Step 11: End

Each system address consists of a boolean variable that most of us check with just like the system address variable that's false being a default value. On receiving a packet, a system address checks the continent of its system address variable and relates to the packet differently depending on that state.
When the system address variable is false, it generates a random floating point number w in the range [0; 1]. In case this number is below the marking probability p, probably the packet is selected for marking. Upon random selection, the system address then proceeds to examine whether this randomly selected packet has any previous system address information embedded inside it. Whether or not this will not, then the system address embeds its own identity directly into packet and forwards the packet to a higher system address. However, if the packet has previous routing information, sst address changes its own system address variable to true, and forwards the packet while not changing any one of the information within it.
In case the system address variable for sure, every received packet will surely be inspected for previous system address information. Whenever packet is present that does not contain any previous system address information, sst address identity is embedded in that packet, and to discover the system address variable is about returning to false. The system address increments every packet’s distance field unless that packet was selected for marking. In such situation, the distance field is set to 0. By avoiding overwriting, previous marking information is not lost. By marking the next available packet, the scheme means every system address can have Np marked packets. Hereby N is the total number of packets that flow through the system addresses, and p would be the marking probability of the scheme.

IP TRACEBACK MECHANISM:

Input: Nnetwork packets

Output: Constructed Attacked Graph_path with IP address
foreach Packet do
increase packet count
if (packet contains an edge e in legal graph GI) then
append legitimate sub graph Gl(v!e) to attack graph Ga
end
if (edge e is NOT contained in attacked graph Ga) then
Insert edge e to graph Ga
if (Ga is a connected graph) then
recalculate Termination Number T
reset packet count
end
end
if (Ga is a connected graph) and (packet count > Threshold T)
then
return Ga as the attack graph
end
end

IV.Experimental Results

All experiments are performed with the configurations Intel(R) Core(TM)2 CPU 2.13GHz, 2 GB RAM, and the operating system platform is Microsoft Windows XP Professional (SP2). This framework requires third party libraries like jpcap,winpcap.

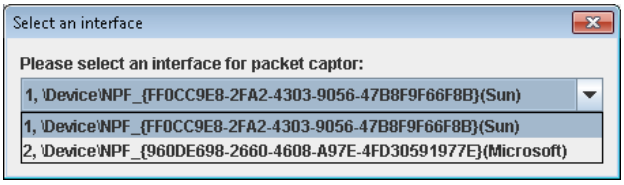

Fig1. Open an Interface

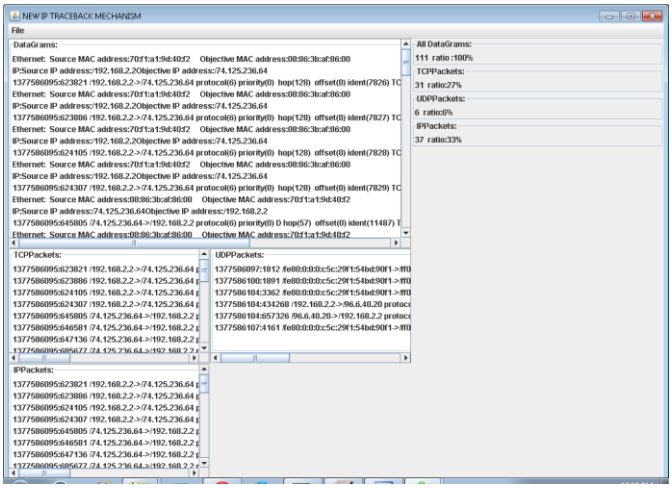

Fig.2 Capture Packets and Classification

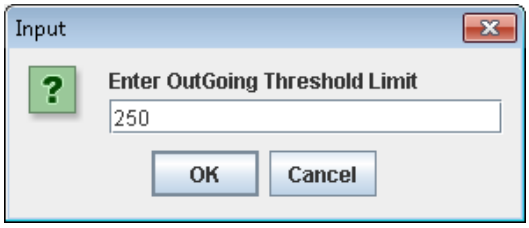

Fig3. Set threshold limit to detect attacks

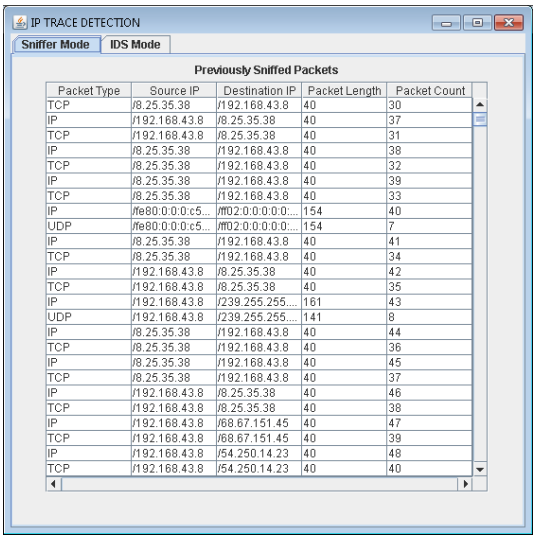

Fig.4 Load data from Database

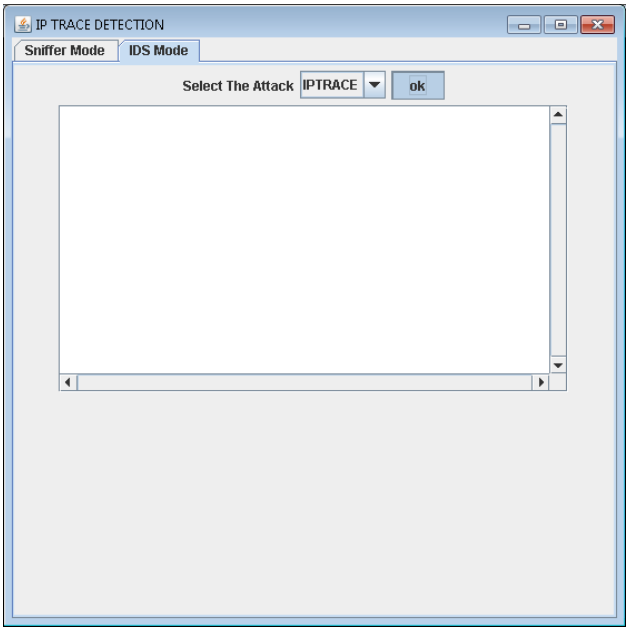

Fig.5 Iptraceback Algorithm to detect attacks

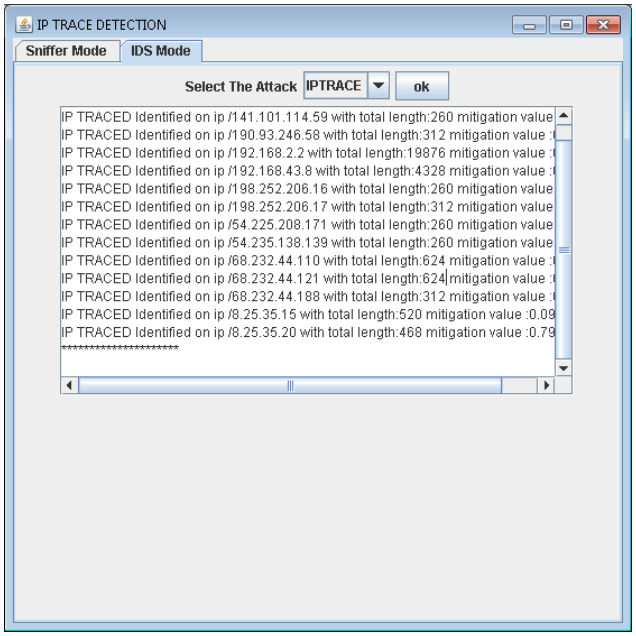

Fig.6 Iptraceback Results

IPTRACED PACKETS AFTER ATTACK:

IP TRACED Identified on ip /117.18.237.191 with total length:416 mitigation value :0.12
IP TRACED Identified on ip /141.101.114.59 with total length:260 mitigation value :0.47
IP TRACED Identified on ip /190.93.246.58 with total length:312 mitigation value :0.39
IP TRACED Identified on ip /192.168.2.2 with total length:19876 mitigation value :0.943
IP TRACED Identified on ip /192.168.43.8 with total length:4328 mitigation value :0.903
IP TRACED Identified on ip /198.252.206.16 with total length:260 mitigation value :0.676
IP TRACED Identified on ip /198.252.206.17 with total length:312 mitigation value :0.133
IP TRACED Identified on ip /54.225.208.171 with total length:260 mitigation value :0.582
IP TRACED Identified on ip /54.235.138.139 with total length:260 mitigation value :0.013
IP TRACED Identified on ip /68.232.44.110 with total length:624 mitigation value :0.255
IP TRACED Identified on ip /68.232.44.121 with total length:624 mitigation value :0.379
IP TRACED Identified on ip /68.232.44.188 with total length:312 mitigation value :0.664
IP TRACED Identified on ip /8.25.35.15 with total length:520 mitigation value :0.096
IP TRACED Identified on ip /8.25.35.20 with total length:468 mitigation value :0.795

Performance Results:

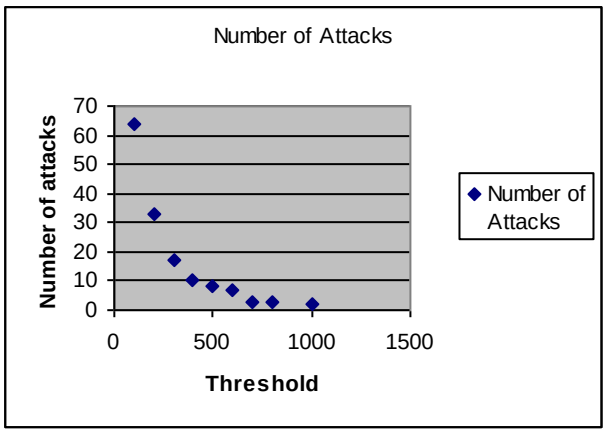

Fig 7. Number of attacks Vs Threshold

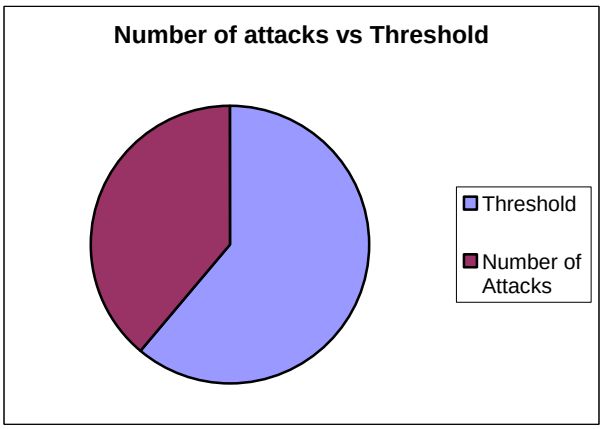

Fig 8. Number of attacks Vs Threshold Distribution

V. CONCLUSION AND FUTURE SCOPE

In this paper existing approaches and its drawbacks are identified and analyzed. In this proposed work, different network packets are analyzed from different source of neworks. This system is better suited for web as well as lan networks. DDOS attacks in web and lan network are experimented. Finally, experimental result shows that proposed approach is better suited for large networks and detection rate is high compare to traditional approaches. In future, cloud based DDOS attacks need to detect by employing advanced iptraceback mechanism.

REFERENCES

[1] Saurabh S,SaiRam,A.S Linear and Remainder Packet Marking for fast IP Traceback COSMNET, fourth international journal 2012.
[2] NingLu;Yulong wang a novel approach for single packet ip traceback based on routing path parallel and distributed systems 20 international conference 2012.
[3] Mercy Shaline and Vijayalakshmi M IP traceback system for network and application layer attacks Recent trends in Information Technology,2012.
[4] Okada M, Katsuno Y 32-BIT as number based ip traceback (IMIS)2011 fifth International conference.
[5] Khan ,Z.S;Akram N; secure single packet ip traceback mechanism to identify the source (ICITST)2010
[6] Integrated DDoS Attack Defense Infrastructure for Effective Attack Prevention, Yang-Seo Choi, Jin-Tae Oh, Jong-Soo Jang