International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 07 | July 2018 www.irjet.net p-ISSN: 2395-0072

Classification of IP Spoofed Request using Heuristics Fuzzy Approach in Computer Network

Dr. N. Arumugam

Lecturer (SG), Dept of ECE, Nachimuthu Polytechnic, Pollachi, Tamilnadu, South India.

Abstract— Distributed Denial of Service (DoS) attack presents a serious problem for Internet communications. This problem is aggravated when the attackers spoof their source IP addresses. As it is easy for an attacker to change the source IP address will leads to unauthorized access of network resources. Many solutions are furnished by the research community to detect this problem. This paper proposed a detection method by considers data flow as a metric. Since an attacker can change any fields in the IP packet expect hop count field, this paper presents a heuristic fuzzy logic approached detection of spoofed packets. The effective implementations of the fuzzy rule, two member functions μ_{HC} & μ_{PTT} are defined from the received IP header. Classification of spoofed packets is identified from the closeness between the changes in the hop count and packet transfer time with the fuzzy membership function by applying fuzzy triangular membership function.

Keywords: DDoS attacks; spoofed IP; hop count; packet transfer time

I. INTRODUCTION

In IP network spoofing has often been exploited by Distributed Denial of Service (DDoS) attacks to conceal flooding sources and dilute localities in flooding traffic and persuade legitimate hosts into reflectors, redirecting and amplifying flooding traffic [1]. Most DDoS attacking tools spoof IP addresses by randomizing the 32-bit source-address field in the IP header [2], which conceals attacking sources and dilutes localities in attacking traffic. The recent "backscatter" study [3], which quantifies DoS activities in the current Internet, has confirmed the widespread use of randomness in spoofing IP addresses. Moreover, some known DDoS attacks, such as smurf [4] and more recent Distributed Reflection Denial of Service (DRDoS) attacks [5], are not possible without IP spoofing. Such attacks masquerade the source IP address of each spoofed packet with the victim's IP address. In general, DDoS attacks with IP spoofing are much more difficult to defend.

To detect and prevent DDoS attacks, there are two distinct approaches: *router-based* and *host-based*. The router-based approach installs detection mechanisms inside IP routers to trace the source(s) of attack [6], or detect and block attacking traffic [7].However, these router-based solutions require not only router support, but also coordination among different routers and networks, and wide-spread deployment to reach their potential. In contrast to the router-based approach, the host-based approach can be deployed immediately. Moreover, end systems should have a much stronger incentive to deploy defense mechanisms than network service providers.

The current host-based approaches protect an Internet server either by using sophisticated resource-management schemes [9], or by significantly reducing the resource consumption of each request to withstand the flooding traffic such as SYN cookies [10] and Client Puzzle [11]. Existing host-based solutions work at the transport-layer and above, and cannot prevent the victim server from consuming CPU resource in servicing interrupts from spoofed IP traffic. At high speed, incoming IP packets generate many interrupts and can drastically slow down the victim server [12]. Therefore, the ability to detect and filter spoofed packets at the IP layer without any router support is essential to protection against DDoS attacks.

This paper proposes a trivial scheme that validates incoming IP packets at the victim internet server without using any cryptographic methodology or router support. The goal of this paper is not to achieve perfect authentication, but to screen out most bogus traffic with little collateral damage. The fundamental idea is to utilize inherent network information such as the number of hops and packet transfer time of the received packet takes to reach its destination.

The remainder of the paper is organized as follows. Section II presents the TTL-based hop-count computation, section III demonstrates computation of packet transfer time, section IV discuses the construction of mapping

tables, section V presents the correlation between hop count and packet transfer time, section VI explain fuzzy membership function based search strategy of spoofed packets, section VII presents the results and discussions and the paper conclude with future extension in section VIII.

II. HOP COUNT COMPUTATION

The number of hops a packet takes to reach its destination reflected in the Time-to-Live (TTL) field of the IP header and also each intermediate router decrements the TTL value by one before forwarding a packet to the next hop. Since hop count information is not directly stored in the IP header and it might compute from the final TTL value.

A TTL (Time to Live)

TTL indicates the time in which a packet can exist on the network [19].It is defined to prevent a packet from circling on the network and it is decremented by one when passing through one router. Hence it is possible to calculate hop count from the TTL value. TTL is an 8-bit field in the IP header, originally introduced to specify the maximum lifetime of each packet in the Internet.

B Computing hop count

There are two methods to measure the hop count from a host. One is an active measurement and the other is a passive measurement. The first method is to use ICMP ECHO packets. In most cases this gives an accurate hop count. However applying this method to thousand hosts is not realistic because sending lots of ICMP packets is not recommended as a measuring method.

The second method is simply to subtract the TTL of a received IP packet from its initial value. This can be done with out sending any sample packets and therefore is ideal of measuring the hop counts of many hosts.

(Hop count) = (initial TTL)-(TTL)

However in order to use this method the initial TTL values should be known in advance.

C Problem with Initial Values of Ttl

According to RFC 1700the recommended initial TTL value is 64. However this rule is often ignored on the real internet. Swiss Academic & Research Network (SWITCH) has researched initial TTL values of different OS (Operating Systems). As a result there are six initial TTL values: 30,32,60,64,128 and 255.

The packet whose initial TTL value 255 and the initial TTL value 128 can be distinguished from other easily. However it is more difficult to assess packets whose TTL values are less than 60 or 64.The same problem occurs to the packets with TTL value less than 30.The popular OS like Microsoft Windows, Linux and Free BSD are using 32 and 64 as initial values. Hence the following formula to convert TTL to hop count,

Hop Count =

32-TTL	TTL<=32
64-TTL	TTL<=62
128-TTL	TTL<=128
255-TTL	TTL<=255

III .PACKET TRANSFER TIME

It is also necessary to establish a method to measure packet transfer time between the measuring point and the target host. Since there is no time stamp field in IP header, it is impossible to calculate packet transfer time by simply analyzing the packet header.

The most accurate method to measure transfer time is to use measurement software which sends and receives sample packets. However, this method can be used only for specific hosts by executing a measurement program and therefore it is not suitable to collect packet transfer times of a number of hosts. Another way is to send an ICMP ECHO packet and count the time till its reply return from the host. This method seems to work well but it is very tedious to sending many ICMP packets and also the round trip time of ICMP packets differ from IP packets.

To overcome the above said problem a passive method, which is to capture and analyze TCP handshaking packets. This method makes it possible to collect roundtrip times of a large number of hosts without sending any unnecessary packets. When making a TCP connection one host send a TCP packet with a SYN flag bit on in order to request connection establishment. Immediately up on receiving the packet destination host sends back a TCP packet with ACK and SYN flags bits on to accepts the request and to establish the connection in the reverse

direction. The negotiation ends when the first host sends back an ACK packet. Logically TCP tries to make both upstream and downstream connection separately to achieve full duplex transmission. This process is called 3-way handshake [20].

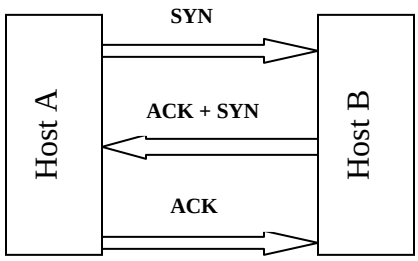

Figure 1. 3-Way handshake

The ACK packet is sent back immediately after the SYN packet is received, the ACK packet can be regarded as an ECHO packet of the SYN packet. There fore the difference between the time when the SYN packet is sent and the time when the ACK packet returns can be used as the approximation of the packet transfer time between two hosts.

IV. CONSTRUCTIONS OF MAPPING TABLE

An accurate mapping table has been constructed, which contains the set values of hop count and packet transfer time for all IPs. Building an accurate mapping table is critical thus the following methods were implemented to construct and update the mapping table:

1) Initialize the mapping table

2) Up-to-date mapping table

A Initialization of the mapping table

To construct a mapping table initially, the administrator of an Internet server should collect traces of its clients to obtain both IP addresses and the corresponding hop-count values along with packet transfer time (PTT). The initial collection period should be long enough to ensure good accuracy even at the very beginning, and the duration should depend on the amount of daily traffic the server is receiving. For a popular site, the collection period of a few days could be sufficient, while for a lightly loaded site, a few weeks might be more appropriate.

After the initial population of the mapping table and activation, it will continue adding new entries to the mapping table when requests with previously unseen legitimate IP addresses are sighted. Thus, over time, the mapping table will capture the correct mapping between IP address and hop-count along with PTT for all legitimate clients of a server. This ensures that spoofed IP traffic can be detected during a DDoS attack.

2) Updating the mapping table

The mapping must be kept up-to-date as hop-counts of existing IP addresses change. The hop-count from a client to a server could change as a result of relocation of networks, routing instability, or temporary network failures. Some of these events are transient and therefore can be omitted, but longer-term changes in hop-count must be captured.

According to [22], the hop count number is start with 4 and the maximum number of hops is 30. Hence multiple IP addresses may have the same hop count values. Unfortunately an attacker may have the same hop count values as that of spoofed IP address. It is prudent to examine hop count distributions at various locations in the internet to ensure that the limited range does not severely diminish the effective ness of HCF.

The most critical aspect in initializing and updating the mapping table is to ensure that only valid mappings are stored in the table. At the time of initialization of this table the administrator have to provide provision for new entries whenever new legitimate IP is signed. As hop count from client to server could change due to router instability or network failure the table should up to date. This update function will execute after a fixed time span and it will change the entry only after the completion of three way hand shaking of TCP connection.

V. CORRELATION BETWEEN HOP COUNT AND PACKET TRANSFER TIME

According to [21] a strong non linear relationship has been observed between hop count and packet transfer time. The following graphical scenarios confirmed the association of two metrics. Hence this paper considered these two metrics for the detection of spoofed IP.

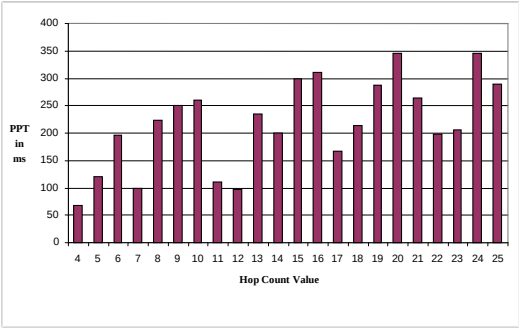

Figure 2. Correlation between Hop Count and Packet Transfer Time

VI. FUZZY MEMBERSHIP FUNCTION BASED SEARCH STRATEGY OF SPOOFED PACKETS

An algorithm based on fuzzy operation is proposed to identify the spoofed packets. The effective implementations of the fuzzy rule, two member functions are defined in the strategy for hop count and packet transfer time. Moreover, a heuristic based fuzzy set approach is built to avoid heavy numerical computing. As per the discussion in the section V the set values of HC and PTT are fixed as HC_{set} and $PTT_{set.}$ To apply heuristic fuzzy triangular membership function the received packets were segregated based on received source IP. From the sub table for every 100ms the change in HC and PTT has been calculated and it denoted as ΔHC and ΔPTT respectively. From the sub table there are two fuzzy set models are developed for identifying the spoofed packets

A Fuzzy Set Model for the Variation of Hop Count

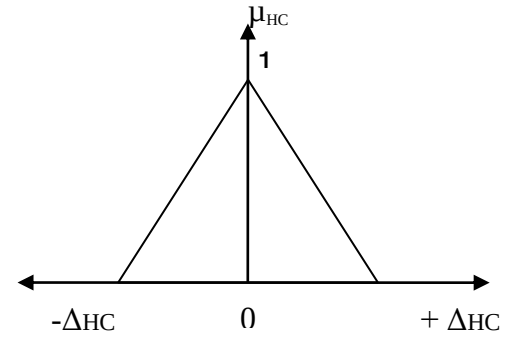

Figure 3. Membership function of HC and HC_{set}

The received hop count value can be evaluated with Fuzzy triangular member function as very close, close, or not close to the HC_{set}.The Figure 3 shows the triangular membership function of HC along with $\mu_{set.}$ A small difference between response time of packets and HC_{set} possesses a large membership value and vice versa. The linguistic terms can be formatted as a membership function with conditions and it is expressed as follows:

$$\mu_{HC} = \begin{cases} 1 - \frac{\Delta HC}{HC_{set}} & \text{for } 0 < \Delta HC < HC_{set} \\ 1 + \frac{\Delta HC}{HC_{set}} & \text{for } -HC_{set} < \Delta HC < 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{--- (2)}$$

Where, $\Delta HC = HC - HC_{set}$

B FUZZY SET MODEL FOR THE VARIATION OF PPT

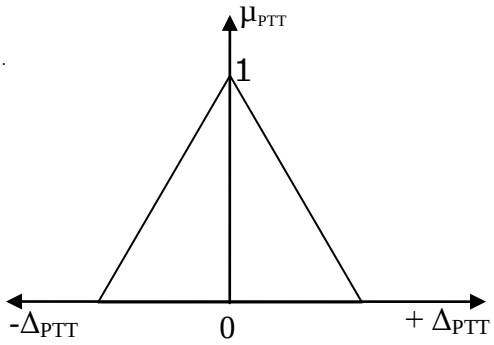

Figure 4. Membership function of PTT and PTT_{set}

The received PTT value can be evaluated with fuzzy triangular member function as very close, close, or not close to the PTT_{set}.The Figure 4 shows the triangular membership function of PTT along with $\mu_{set.}$ A small difference between response time of packets and PTT_{set} possesses a large membership value and vice versa. The relevant membership function with conditions has been formatted as:

$$\mu_{PTT} = \begin{cases} 1 - \frac{\Delta PTT}{PTT_{set}} & \text{for } 0 < \Delta PTT < PTT_{set} \\ 1 + \frac{\Delta PTT}{PTT_{set}} & \text{for } -PTT_{set} < \Delta PTT < 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{--- (1)}$$

Where, $\Delta PTT = PTT - PTT_{set}$

The intersection of μ_{HC} and μ_{PTT} finally classifies the received packets spoofing characteristics . which has been expressed as,

$$\mu_{Dn} = \min\{\mu_{HC}, \mu_{PTT}\}$$

Where, μ_{Dn} is the fuzziness of the n^{th} received packet from the host.

Furthermore, union of individual packets spoofing characteristics provides the spoofing characteristic of the host and which can be expressed as,

$$\mu_{D} = \max\{\mu_{D1}, \mu_{D2}, \mu_{D1}, \ldots\ldots \mu_{Dn}\}$$

Where, μ_{D} is the fuzziness of the received packets from the host.

VII. RESULTS AND DISCUSSIONS

The proposed scheme has been tested with the neighboring nodes at the victim server. As per the discussion in section V the initial mapping table has been constructed. Similarly at the victim server, the received packets hop count and packet transfer time from host 1are tabulated in the table I for every one second. The table II shows the details received from host 2. Since the victim server connected with all neighboring host it may receive spoofed packets also. Hence the table contains the details of legitimate packets along with spoofed packets. To identify the spoofed packets, the heuristic fuzzy logic was applied and the membership function values of μ_{HC} and μ_{PTT} were calculated using every 100 seconds and the calculated values were tabulated in the table III for the host 1. Similarly the table IV shows the details of host 2.

Table I and II shows the status of received packets details for the first 100 seconds at the victim server from the host1 and 2. In actual scenario attacker may intentionally modify the source address of the packets it sends from the compromised host.

TABLE I. PACKETS DETAILS FROM HOST1

S.No	HOP Count	PTT in ms
1.	13	305.658
2.	11	372.308
3.	16	318.133
4.	15	322.143

TABLE II. PACKETS DETAILS FROM HOST2

S.No	HOP Count	PTT in ms
1.	14	146.868
2.	16	100.778
3.	13	090.814
4.	15	110.695

From the table I and II the changes in hop count (ΔHC) and packet transfer time (ΔPPT) were calculated by using the equation (3) & (4).

$$\Delta HC = HC - HC_{set} \quad \text{-- (3)}$$

$$\Delta PTT = PTT - PTT_{set} \quad \text{-- (4)}$$

Based on the values of ΔHC and ΔPTT the corresponding membership function values of

μ_{HC} and μ_{PTT} are calculated using the equation (1) & (2) and the results tabulated in the table III and IV.

TABLE III DETAILS OF μ_{HC} and μ_{PPT} RECEIVED FROM HOST 1

S.No	μ_{HC}	μ_{PTT}	μ_{D}
1.	0.9285	0.9818	0.9285
2.	0.7857	0.8039	0.7857
3.	0.9780	0.8571	0.8571
4.	0.9285	0.9651	0.9285

TABLE IV DETAILS OF μ_{HC} and μ_{PPT} RECEIVED FROM HOST 2

S.No	μ_{HC}	μ_{PTT}	Fuzzy ness
1.	0.8235	0.7787	PS
2.	0.9411	0.8380	PS
3.	0.7647	0.7551	NS
4.	0.8823	0.9204	PS

From the table III and IV the closeness of μ_{HC} and μ_{PTT} were calculated by applying min-max computation. The host contains minimum mutual transfer function μ_{HC} & μ_{PTT} has been identified as NS (No spoofing).and the rest of the packets were identified as PS (Partially Spoofed).

VIII.CONCLUSION

The proposed spoofing detection methods is classified the received source IP packets as a NS (Not Spoofed), PS (Partially Spoofed), and FS (Fully Spoofed) based on the heuristics fuzzy triangular membership approach. The legitimate request is identified from the closeness of μ_{HC} and μ_{PTT} by applying min-max computation.

REFERENCES

[1] A. Chakrabarti and G. Manimaran, "Internet Infrastructure Security: A Taxonomy," in Proc. Conf. IEEE Network, vol.16, no.6, pp.13-21, 2002.

[2] CERT Coordinate Center, "Denial of Service Attacks," http://www.cert.org/tech tips/denial of service.html.

[3] B.Al-Duwairi and .Manimaran,"International dropping: A novel scheme for syn flooding Mitigation," in Proc. Conf. IEEE INFOCOM, April 2006.

[4] *Christos Douligeris and Aikaterini Mitrokotsa* Department of Informatics University of Piraeus, Piraeus, Greece, "Ddos Attacks *and* Defense Mechanisms: A Classification".

[5] R. R. Talpade, G. Kim and S. Khurana, 'W O W: Traffic-based Network Monitoring Framework for Anomaly Detection". Prw. 4'h EEE Symposium *on* Computers and Communications, Ted Sea, EgJpt, pp. 442451, June 1999.

[6] W. Lee and S. J. Stolfo, "'Data mining approaches for intrusion *detection",* 7' USENIX Security Symposium, San Antonio, TX, pp 79-93, January 1998.

[7] I. B. D. Cabrera et al., "Proactive Detection of Distributed Denial of Service Attacks using MIB Traflic Variables - A Feasibility SNdy", IFIPi EEE Int. Symp. On Integrated Network Management, Seattle, WA, May 2001.

[8] Y. Huang, J. M Pullen, "Countefmg Denialuf-Sewice anacks Using Congestion Triggered Packet Sampling and Filtering", Proc. IO' ICCCN, Anzona, USA, *Oct.* 2001.

[9] T.M. Gil and M Poleto, "MULTOPS: a data-structure for bandwimh attack detection", hoc. lo* USENlX Security Symposium Washington, DC, pp.23-38, Aug. 2001.

[10] Haining Wang*, Member, IEEE*, Cheng Jin, and Kang G. Shin*, Fellow, IEEE, "*Defense Against Spoofed IP Traffic Using Hop-Count Filtering *",* IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 15, NO. 1, FEBRUARY 2007.

[11] *S.* M. Bellavin, "ICMP traceback messages", Internet Draft, 2001

[12] C. Banos, "A proposal for ICMP traceback messages", Internet Draft, Sept. 2000.

[13] H. Burch and H. Cheswick '"Tracing anonymous packets to the approximate source", hoc. USENIX LISA, New Orleans, pp.319-327, Dec. 2000.

[14] R. Stone, '%enterTrack: An IP OverlayNetwork for Tracking DOS Roods", Proc. 9" USENlX Security Synpsiuq Denver, Colorado, pp 199-212, Aug. 2000.

[15] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, 'Wetwork support for IP traceback, Proc. IEWACM

Trimsaction *on* Networking vol. 9: (3), pp. 226237, June 2001.

[16] D. X. Sang and A. Penig, "Advanced and Authenticated Marking Schemes for

IP Traceback", hoc. IEEE INFOCOW Anchorage. AK,

[17] A. C, Snoeren, C. Pamidge, L. A. Sanchez *C.* E. Jones, F.Tchakountio, B. Schww and T. Strayer, "Single-Packet IP Traceback, EEE/ACM Transanions on Networking (TON), vol. 10: (ti), pp 721- 734, Dec. 2002.

[18] The Swiss Education and Research Network, "Default TTL Values in TCP/IP,"2002, http://secfr.nerim.net/docs/fingerprint/en/ttl_default.html.

[19] Information Science Institute in university of Southern California, "Internet protocol",RFC791,1981.

[20]W.Richard Stevens, "TCP/IP Illustrated, Volume 1" Addison Wesley Longman, Inc., 1994

[21]Keita Fujii and Shigeki Goto Dept of Information and Computer Science, Waseda University, Tokyo, "Correlation between Hop Count and Packet Transfer Time"

[22] http://www.opus1.com on 12th October 2009