An Enhanced Packet Marking and Traceback Algorithm for Ip Traceback

T. MYTHILI[1], M. KARTHICK KIRAN[2], S. NOORJAHAN[3], DR. S. MALLIGA[4]
[1,2,3,4] *Dept. of Computer Science and Engineering, Kongu Engineering College*

Abstract - Distributed Denial of Service (DDoS) attack is a major threat in today's world. Attackers hide their identity by spoofing and defending. An idea of an enhanced packet marking and trace back algorithm for ip traceback to identify an attacker that facilitates the traceback of the spoofed packet to its origin. Numerous ip traceback techniques exist, but they have limitations like the number of packets required or storage computational overheads incurred at routers. The technique proposed reduces computational time. The efficacy of the proposed scheme is compared to that of other single-packet traceback schemes in terms of computational time, storage, accuracy.

Indexed Terms -- IP Spoofing, DoS / DDoS, IP Trace back, Packet marking and logging, trace back

I. INTRODUCTION

Spoofing is the creation of TCP / IP packets with a forged source IP address. To forward packets over the Internet, routers use the destination IP address, but ignore the source address and it is never authenticated. This motivates attackers to exploit spoofing to attack Denial of Service (DoS) or Distributed DoS (DDoS). A DoS / DDoS attack is characterized by an explicit attempt by attackers to prevent the use of that service by legitimate users of a service. Because DoS / DDoS attackers use spoofing, finding the source of such attacks and defending against them is very difficult.

There are two kinds of attacks on DoS / DDoS: flooding attacks and exploits of software. It is not always necessary to flood a victim with these attacks. However, a single well - focused packet of attacks can detrimentalize a target system.

Researchers have made enormous efforts to address these attacks. Such an effort is the technique of IP trace back. It is a technique that identifies the true origin of the packet and establishes protective mechanisms to prevent spoofing attacks. IP trace back is used to identify flooding as well as single packet attacks.

IP trace back techniques can be broadly classified into two types: in-band and out-of-band approaches. In-band approaches use IP packets to enable trace back and out-of - band approaches use a separate trace packet such as an ICMP packet.

IP trace back schemes can be classified as link testing, logging, marking, or hybrid methods. Link testing and packet marking-based trace back schemes do not require router storage, but they require a huge number of packets to reconstruct the attack path. Therefore, when the size of the attack is increased, they can produce numerous false positives. Alternatively, packet logging and hybrid method-based trace back schemes may identify attackers with fewer packets, but they require considerable router memory and computation.

The proposed enhanced packet marking and traceback algorithm for the ip traceback scheme uses simple bitwise operations like XOR for logging. The novelty of the enhanced packet marking and traceback algorithm for the IP traceback scheme lies in its ability to trace back each attack packet with negligible storage and computational overheads but with greater precision and accuracy.

The contributions of this paper can be summarized as follows:

An IP traceback scheme, an enhanced packet marking and traceback algorithm for the IP traceback,

1) Which traces an attacker and attack path
2) Using a single packet-requires minimal computation during marking and logging-significantly

3) Reduces the need for router storage.

The rest of this paper is organised as follows:
1) Abstract
2) Introduction
3) Related Work
4) High Precision Single Packet IP Traceback
5) Proposed IP Traceback
6) Experimental Analysis
7) Conclusion

II. RELATED WORK

Several efforts have been made to reduce the afforded anonymity of IP spoofing. One approach, namely ingress filtering [1], blocks packets on the routers when the packets have illegitimate IP addresses. However, this method requires tremendous power and knowledge to filter each incoming packet. This works effectively on border routers, while the success rate in transit networks is dependent on other upstream Internet service providers (ISPs). In addition, legitimate user table and look-up time are increasing exponentially as the network grows, which impairs high-speed links. This method also involves complications with existing services that rely on source address spoofing, such as mobile IP(MIP) and some satellite hybrid architectures; therefore, IP trace back schemes remain necessary for attack allocation and defense.

IP trace back schemes can be classified into reactive and proactive schemes based on their mode of operation. Link testing [2-5] is a reactive mechanism that requires an attack to be alive until the trace back is completed, while logging, marking, and hybrid methods are proactive methods that do not require the attack to be alive for the trace back process to be completed. The initial approaches to trace back depend on the flow of traffic attack. They either checked the packet signatures hop-by-hop up to the attacker [2] or intentionally flooded the network to observe the drop rate of the packet[5].

Both methods required a huge amount of packets and the attack to remain alive until the completion of the traceback process. Later, the Internet control message protocol (ICMP) based trace back scheme was proposed[6] , in which an out-of-bound ICMP packet containing partial path information was generated by the routers with minimum probability; for example, 1/20,000 packets. This method required more packets to trace back the attacker. Saurabh and Sairam [7] utilised ICMP messages to trace reflector attacks. Yao et al,[8] avoided additional ICMP messages and proposed a technique that uses ICMP error messages generated by routers to locate an attacker. This technique depends on the topology of the network and can identify the attacker, but only if sufficient ICMP error messages are generated.

Router interface-based approaches [9-13] use interface router numbers rather than IP addresses to trace the attacker back. The RIM (Router Interface Marking), proposed by Chen et al.[10] is a packet marking-based approach that probabilistically marks packets; therefore, it requires more packets and leads to false positives as the number of attackers increases. Malliga and Tamilarasi [12] have proposed Modulo / Reverse Modulo Technique (MRT), a hybrid scheme that uses router interfaces. Using the router interface, MRT performs mathematical calculations and marks the resulting value. This process continues to the victim. During trace back, the reverse calculations are performed to identify the upstream links. MRT uses a 32-bit marking field and requires routers to be stored when the marking field overflows.

Malliga and Tamilarasi [11] proposed another hybrid scheme called MORE (Modulo and Reverse modulo), in which the field size of the marking is reduced to 16 bits but the number of log ta-bles is increased depending on the degree of the router. Both MRT and MORE index the log by packet digest, which requires logging every packet that passes the same path. Although MRT and MORE can trace the attacker back using a single packet, an exhaustive search is required during the trace-back process. They can also produce false positives due to collisions in the log table. M-H Yang and M-C Yang [9] recently proposed RIHT, a hybrid trace-back scheme that uses the router interface. In RIHT, the mathematical calculations performed in MRT are appropriate modified and replaced the log table with a hash table to reduce the search time. RIHT has been shown to be superior to any other hybrid scheme in terms of storage requirements, computational time and

accuracy. Although trace-back time is minimized by eliminating the exhaustive search performed by MRT and MORE, the logging time in RIHT is high due to hashing and collisions in the hash table. Kamaldeep et al. [13] proposed a trace-back scheme that minimizes RIHT logging time; however, it continues to use time-consuming double hashing.

In summary, existing schemes are exceptional in their own respects; however, they have several drawbacks. They either require / have:

1) Numerous packets for trace back
2) Abundant storage in routers or victims
3) Long computational time
4) A high FPR with an increase in the number of attackers. FPR-False Positive Rate, the packet is not spoofed by an attacker, but the packet is falsely spoofed in FPR.
5) Dependence on each router's neighbors in the attack path.

The proposed an enhanced packet marking and trace back algorithm for the Ip trace back scheme attempts to overcome these disadvantages by generating mark values using bitwise operations on router interface identifiers(IDs). The scheme achieves an enhanced packet marking and trace back algorithm for ip trace back capability with negligible storage and overhead computation. The scheme employs an efficient data structure that minimizes logging time during logging and eliminates the search time during trace back.

III. HPSIPT (HIGH PRECISION SINGLE IP TRACEBACK)

A High Precision Single Packet IP Trace back (HPSIPT) scheme that solves this tradeoff by precisely tracing back each packet with negligible storage and computational overheads. HPSIPT uses the interface of the router rather than the IP address to mark the packets. HPSIPT scheme uses simple bitwise operations such as XOR and circular shift for logging and reduces router storage requirements (less than 10 KB in most routers).

IV. PROPOSED IP TRACEBACK

The existing system, HPSIPT scheme uses simple bitwise operations such as XOR and circular shift for marking and trace back algorithm. Computational time is high because circular shift is based on number one in a bit. Only use XOR operation for marking and trace back algorithms in the proposed system, reduce computational time.

A. Packet Marking And Logging Algorithm:

$P.mark_1$ and $P.mark_{2\ are}$ marking fields.$P.mark_1$ and $P.mark_2$ are updated recursively on all routers on the attack path until they reach the victim. First, the first marking field value ($P.mark_1$) is logged in the H_i hash table, corresponding to the key x of the outer hash table in the key y of the inner hash table, where x is the value obtained from the incoming interface ID (I_{in}) through which the packet entered that particular router, and y is the XOR value of the first marking field and the second marking field. Therefore, the first marking field is updated with the value of x, whereas the second marking field is updated with the XOR value of y and the outgoing interface ID through which the packet is forwarded to the next router. This process is repeated at all routers on the attack path until the packet reaches the victim.

B. Marking Algorithm:

1. Begin

2. if Ri is a border router then

   2.1 $P.mark_1=0$

   2.2 $p.mark_2=0$

   end if

3. $x = I_{in}$

4. $y = P.mark_1 \oplus P.mark_2$

5. $H_i[x][y]=P.mark_1$

6. $P.mark_1=x$

7. $P.mark_2=y \oplus I_{out}$

8. Forward packet P to the next router

9. End

Let's assume that two 8-bit marking fields are used for illustration purposes. Assume that a packet P is received by router R1 from the attacker's LAN through the incoming interface 1(I_{in}=1) and leaves through the outgoing interface 250(I_{out}=250). P.$mark_1$ and P.$mark_2$ packet values are initialized. P.$mark_1$ is logged in the R_1 hash table H_1 at[x][y], where x is the value obtained from I_{in}((x= I_{in})=1=1), and y is the XOR value of P.$mark_1$and P.$mark_2$ (y= P.$mark_1$ ⊕ P.$mark_2$)=0 ⊕ 0=0).Therefore, in the R_1 hash table H_1,0 is logged in H_1[1][0]. P.$mark_1$ and P.$mark_2$ values are updated. P.$mark_1$ is updated to the value x (P.$mark_1$=x=1). P.$mark_2$ is updated with the XOR value of y and I_{out} (P.$mark_2$=y ⊕I_{out}=0 ⊕ 250=253). The attack packet will then be forwarded to the next R_2 router and the same process will be repeated. This process of marking is repeated until the victim reaches the packet. Table 1 shows the hop-by-hop update of the marking fields (P.$mark_1$ and P.$mark_2$) of an attack packet in the routers on the attack path and the log made in the respective router's H_i hash table during the marking process. Finally, the packet reaches the victim with the P.$mark_1$ and P.$mark_2$ values of 0 and 234 respectively.

Table 1. Illustration of marking fields' values and log updates.

I_{in}-Packet entered by interface

I_{out}-Packet left by interface

Router Visited	Iin	Iout	P.$mark_1$ (before)	P.$mark_1$ (after)	P.$mark_2$ (before)	P.$mark_2$ (after)	Update in the Hash Table H
R1	1	250	0	1	0	250	H[1][0]=0
R2	2	10	1	2	251	241	H[2][251]=
R3	7	200	2	7	243	59	H[7][243]=2
R4	0	220	7	0	60	224	H[0][60]=7
R5	0	10	0	0	224	234	H[0][224]=0

C. Trace back Algorithm:

The attack path and attacker are identified from the received packet in the trace back process. Once the victim detects an attack and intends to trace back a packet, a request for trace back is sent to the immediate upstream router. The trace back request packet contains the mark values found in the corresponding attack packet.

The values of the marking fields are restored to the premarking status at each router before forwarding the trace back request to the upstream router on the attack path; that is, the packet's old marking field values are identified. The value of the first marking field (i.e., T.$mark_1$) is retrieved from the H_i hash table stored in the respective router. The value associated with the inner hash table key y, which corresponds to the outer hash table key x, is revised to become the value of T.$mark_1$, where x denotes the current T.$mark_1$ value and y denotes the XOR value of T.$mark_2$ and I_{in}. The value of the second marking field value is obtained by applying XOR value of y and T.$mark_1$. With the revised T.$mark_1$ and T.$mark_2$ values, the traceback request packet is forwarded through the outgoing I_{out} interface to the next upstream router. This process is repeated continuously until the trace-back request packet reaches the border router; that is, until I_{out} is connected to a local network, which is the LAN of the attacker.

D. Trace back Algorithm:

1. Begin

2. I_{out} =T.$mark_1$

3. if I_{out} is connected to a router then

3.1 x=T.$mark_1$

3.2 y=T.$mark_2$ ⊕ I_{in}

3.3 T.$mark_1$=H_i[x][y]

3.4 T.$mark_2$=y ⊕ T.$mark_1$

3.5 Forward the trace-back request T via I_{out} to the upstream router on the attack path.

end if

4. R_i border router of the attacker.

5. I_{out} connected to attacker's LAN

6. End

P reaches the victim, P.mark$_1$ is 0, P.mark$_2$ is 234 (Table 1). These values are copied into the marking fields of the trace back request packet. The trace back request packet T is then sent to the upstream router R_5 with T.mark$_1$=0 and T.mark$_2$= 234. Router R_5 receives the request for trace back through its 10 (I_{in}= 10) incoming interface. Router R_5 then attempts to locate the next upstream router in the attack path by identifying the upstream 0(I_{out}) interface. Table 2 shows the hop-by-hop update of the marking fields (T.mark$_1$ and T.mark$_2$) of a traceback request packet on the routers on the attack path, the identified upstream interface (I_{out}), and the traced out upstream router during the trace back process.

The T.mark$_1$ denotes the outgoing interface connected to the upstream router on the attack path, according to the path reconstruction of the proposed traceback algorithm. The next router can be reached through the outgoing interface is 0 (I_{out}= (T.mark$_1$)=0=0) in this sample case. The next upstream router in the attack path is identified as R_4. If I_{out} is connected to a router, then the marking field values are restored to the premarking status before the request is forwarded to the next router. The value logged in hash table H_5[x][y] is copied to T.mark$_1$, where x is 0(x= T.mark$_1$=0) and y is 224 (y= T.mark$_2$⊕ I_{in}= 234 ⊕ 10= 224). Table 2 shows that the value stored at H_5[0][224] was 0. So T.mark$_1$ becomes 0. T.mark$_2$ is updated to the XOR value of y and T.mark$_1$, 224 (T.mark$_2$= (y ⊕ T.mark$_1$)= (224 ⊕ 0)= 224). Thus, T.mark$_2$ is updated from 234 to224. The traceback request is then forwarded to router R_4 via the identified upstream interface 0 (I_{out}) with T.mark$_1$=0 and T.mark$_2$= 224.

The process of trace back is continued until I_{out} connects to a local network. In the case considered, when the request for trace back reaches router R_1, T.mark$_1$ is 1 and T.mark$_2$ is 250. First, the algorithm confirms that the next outgoing interface is 1. Because the interface is connected to a local network, the process of trace back culminates in router R_1, where it identifies the LAN and R_1 of the attacker as a border router. With a single packet, the attack path, the border router of the attacker and the LAN of the attacker are identified.

Table 2: Illustration of trace back using trace back request packet "T"

I_{in}-"T" entered through I_{in}

I_{out}-"T" forwarded through I_{out}

Router Visited	Iin	Iout	T.mark$_1$ (before)	T.mark$_1$ (after)	T.mark$_2$ (before)	T.mark$_2$ (after)	Upstream Router
R5	10	0	0	T.mark$_1$ =H[0][224]=0	234	224	R4
R4	220	0	0	T.mark$_1$ =H[0][60]=7	224	60	R3
R3	200	7	7	T.mark$_1$ =H[7][243]=2	59	243	R2
R2	10	2	2	.mark$_1$= H[2][251]=1	241	251	R1
R1	250	1	1	.mark$_1$= H[1][0] =0	250	0	Attacker's LAN

V. EXPERIMENT ANALYSIS

Comparison for existing system, proposed system and percentage for values for marking and trace back.

For 10,000 packets,the time taken for marking and traceback is 0.252852s and 0.230890s for existing system and the time taken for marking and traceback is 0.045966s and 0.042994s for proposed system. The time difference between the existing and the proposed work gives the percentage gain of 81.82% for marking and 81.38% for trace back.

For 20,000 packets,the time taken for marking and traceback is 0.520726s and 0.462735s for existing system and the time taken for marking and traceback is 0.089974s and 0.079953s for proposed system. The time difference between the existing and the proposed work gives the percentage gain of 82.72% for marking and 82.72% for trace back.

For 1,00,000 packets,the time taken for marking and traceback is 2.604523s and 2.338681s for existing system and the time taken for marking and traceback is 0.463737s and 0.438771s for proposed system. The time difference between the existing and the proposed work gives the percentage gain of 82.19% for marking and 81.24% for trace back. Comparing existing and proposed system, proposed system is more efficent.

VI. CONCLUSION

An enhanced packet marking and traceback algorithm for the IP traceback scheme identifies an attacker and attack path for spoofed packets .This scheme reduce coputational time because simple bitwise operations are utilised in the marking and traceback algorithm and reduces router storage.

REFERENCES

[1] P. Ferguson, D. Senie, Network ingress filtering: defeating denial of service at-tacks which employ ip source address spoofing (BCP 38), 2000. http://tools. ietf.org/html/rfc2827 (accessed 19 February 2018).

[2] R. Stone, Centertrack: an IP overlay network for tracking DoS floods, in: Pro-ceedings of the 9th conference on USENIX Security Symposium (SSYM'00), Denver, Colorado, 2000.

[3] M. Alajeely, R. Doss, A. Ahmad, V. Mak-Hau, Catabolism attack and anabolism defense: a novel attack and traceback mechanism in opportunistic networks, Comput. Commun. 71 (2015) 111–118.

[4] P. Wanga, H.T. Linb, T.S. Wangb, An improved ant colony system algorithm for solving the IP traceback problem, Inf. Sci. 326 (2016) 172–187.

[5] H. Burch, B. Cheswick, Tracing anonymous packets to their approximate source, in: Proceedings of USENIX Security Symposium, LISA, San Diego, CA, June 2000.

[6] S. Bellovin, et al., ICMP Traceback Messages, IETF Internet Draft, Version 4, 2003 https://tools.ietf.org/html/draft-ietf-itrace-04 (accessed 19 February 2018).

[7] S. Saurabh, A.S. Sairam, ICMP based IP trace back with negligible overhead for highly distributed reflector attack using bloom filters, Comput. Commun. 42 (2014) 60–69.

[8] G. Yao, J. Bi, A.V. Vasilakos, Passive IP traceback: disclosing the locations of IP spoofers from path backscatter, IEEE Trans. Inf. Forensics Secur. 10 (2015) 471–484.

[9] M.H. Yang, M.C. Yang, RIHT: a novel hybrid IP traceback scheme, IEEE Trans. Inf. Forensics Secur. 7 (2012) 789–797.

[10] R. Chen, J.M. Park, R. Marchany, RIM: Router interface marking for IP traceback, IEEE Global Telecommunications Conference (GLOBECOM '06), San Fransisco, California, November 2006, pp 1–5.

[11] S. Malliga, A. Tamilarasi, A hybrid scheme using packet marking and logging for IP traceback, Int. J. Internet Protocol Technol. 5 (2010) 81–91.

[12] S. Malliga, A. Tamilarasi, A proposal for new marking scheme with its per-formance evaluation for IP traceback, WSEAS Trans. Comput. Res. 3 (2008) 259–272.

[13] M. Kamaldeep, M. Malik, M. Dutta, Implementation of single-packet hybrid IP traceback for IPv4 and IPv6 networks, IET Inf. Secur. 12 (2018) 1–6, doi:10. 1049/iet-ifs.2015.0483.