http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (2): 182-188, 2007
ISSN 1812-5638

A New Promising IP Traceback Approach and its Comparison with Existing Approaches

[1]V. Murali Bhaskaran, [2]A.M. Natarajan and [3]S.N. Sivanandam
[1]Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode-638052, Tamil Nadu, India
[2]Kongu Engineering College, Perundurai, Erode-638052, Tamil Nadu, India
[3]Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India

Abstract: The problem of identifying the sources of a denial of service (DoS) and distributed denial of services (DDoS) attack is hardest in the Internet security area, because attackers often use incorrect or spoofed source addresses of IP packets. IP traceback system is to identify the origin of IP packets when the source address of these packets is spoofed. In the traceback system, routers are playing vital role and also ISP (Internet Service Provider) involvement is required to trace the origin of the spoofed packets. Several approaches have been proposed to trace IP packets to their origin. Our aim is to evaluate and analyse the most promising traceback approaches with our proposed system. We have evaluated and summarized the results.

Key words: DoS, DDoS, IP traceback, spoofing

INTRODUCTION

In Denial-of-service (DoS) attacks, the packets are routed correctly but the destination becomes the target of the attackers (Kevin *et al*., 2001). DoS attacks are very easy to generate and are very difficult to detect. In a typical DoS attack, the attacker node spoofs its IP address and uses multiple intermediate nodes to overwhelm other nodes with traffic. DoS attacks are typically used to take important servers out of action for a few hours, resulting in DoS for all the users served by the server. It can also be used to disrupt the services of the intermediate routers. Generally, DoS attacks can be categorized into two main types, ordinary and distributed. In an ordinary network based denial of service attack, an attacker uses a tool to send packets to the target system. These packets are designed to disable or overwhelm the target system, often forcing a reboot. Often, the source address of these packets is spoofed, making it difficult to locate the real source of the attack.

In the Distributed DoS (DDoS) attack, there might still be a single attacker, but the effect of the attack is greatly multiplied by the use of attack servers known as agents. The attack not only disables that server but denies access to legitimate user. Routing table poisoning and packet mistreating attacks are capable of causing denial-of-service. Also, new techniques are being invented every day to create denial-of-service attacks, following are the common types of attacks:

UDP flood: User Datagram Packet (UDP) flood technology is used by the hackers to launch a DoS attack. For example, by sending UDP packets with spoofed return addresses, a hacker links one system's UDP character-generating service to another system's UDP echo service.

TCP/SYN flood: In this type of attacks (CERT, 1996), the hacker sends a large volume of SYN packets to a victim. The return addresses of the packets are spoofed. Thus, the victim queues up SYNACKs but cannot continue sending them because it never receives ACKs from the spoofed addresses.

ICMP/Smurf: In this type of attacks (CERT, 1998), the hacker broadcasts ICMP ping requests with the return address spoofed to show the ultimate victim's address, to a large group of hosts on a network. The hosts send their responses to the ultimate victim, whose system is overwhelmed and cannot provide service.

CLASSIFICATION OF TRACEBACK METHODS

Traceback methods proposed in literature for DoS and DDoS attacks, can be broadly categorized as preventive and reactive. Preventive methods take precautionary steps in preventing DoS attacks. A wide range of solutions has been proposed, however, this problem still remains as open one. The reactive methods' solutions aim at identifying the source of the attacks. This

Corresponding Author: V. Murali Bahaskaran, Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode-638052, Tamil Nadu, India

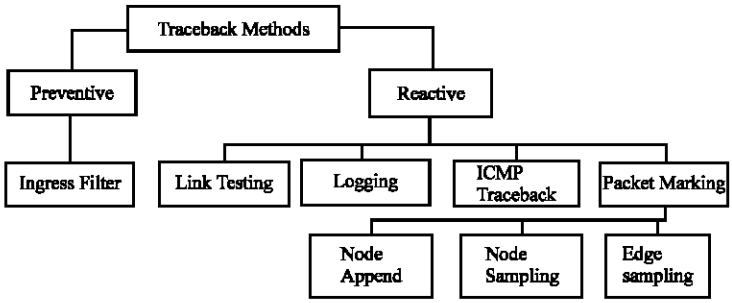

Fig. 1: Classification of traceback methods

is very important because attackers spoof their addresses, thus techniques are needed to trace back to the source of the attack. Figure 1 shows the classification of traceback methods.

Evaluation of traceback systems: This section provides an overview on current state-of-the-art approaches to IP traceback and evaluates them against the ideal system. Overview of an ideal traceback system is given below:

- Very low level of ISP involvement.
- Able to trace the attacker with a single packet.
- The effects of partial deployment can vary from inability to perform tracing altogether to producing meaningful traces limited to the range of deployment.
- Producing meaningful traces are limited to the range of deployment of the traceback system
- Minimal processing overhead during traceback
- Limited amount of additional memory required at the dedicated server and no additional memory requirements on network equipment (routers and switches)
- The ease of evasion (easy to escape if the attacker, who is aware of the scheme, can easily orchestrate an attack that will be untraceable) should be as low as possible
- High level of protection is preferred in a traceback scheme.
- Scalable and configuration of the devices involved should be totally independent of each other.
- Correctly trace back attacks consisting of packets that undergo any number of transformations of any type.

The traceback schemes discussed below fall into four general categories:

- Packet Logging-Hash-based IP traceback (Snoeren *et al*., 2002).
- ICMP traceback- iTrac (Bellovin, 2000; Mankin *et al*., 2001)
- Link Testing - Overlay network (Stone, 2000)
- Packet Marking
  - Probabilistic packet marking (Savage *et al*., 2001; Song and Perrig, 2001)
  - Proposed scheme

All these methods are evaluated based on Fig. 2 and also the merits and demerits of these methods are discussed.

Hash-based IP traceback: Hash-based approach is introduced by Snoeren *et al.* (2002). It is also called as Source Path Isolation Engine (SPIE). In hash-based traceback, every router captures partial packet information of every packet that passes through the router, to be able in the future to determine if that packet had passed through it. In this scheme such routers are called Data Generation Agents (DGAs). DGA functionality is implemented on the routers. The network is logically divided into regions. In Fig. 2, it is assumed that the routers R1, R2, R4, R5 and R7 are in region 1, the routers R3, R6, R8 and R11 are in region 2 and the routers R9, R10 and R12 are in region 3. In every region SPIE Collection And Reduction Agents (SCARs) are connected to all DGAs and are able to query them for necessary information. The SPIE traceback manager (STM) is a central management unit that communicates to IDSs of the victims and SCARs. As packets traverse the network, digests of the packets get stored in the DGAs. In this scheme, constant fields from the IP header and the first 8 bytes of the payload of each packet are hashed by several hash functions to produce several digests. Digests are stored in a space-efficient data structure called a bloom filter, which reduces storage requirements by several orders of magnitude. When a given bloom filters is about 70% full, it is archived for later querying and another one is used. The duration of using a single bloom filter is

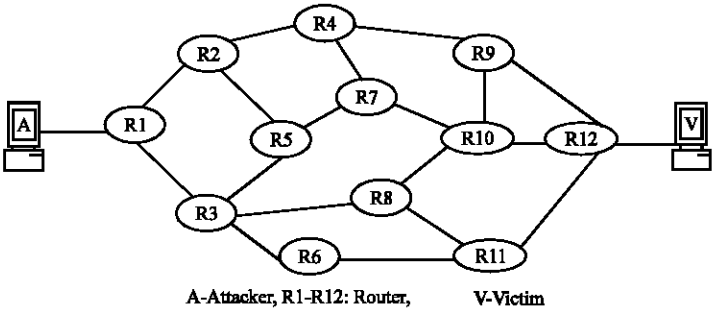

Fig. 2: Internet topology

called a time period. Hash functions also changes for different time periods. Also, a DGA is able to record any transformation (NAT, IPSec, etc.) that may affect those fields. The type of transformation and the data necessary to reconstruct it are stored in the Transform Lookup Table (TLT). Each bloom filters for a given time period has its own TLT associated with it. This scheme involves three functions that must be implemented: STM, SCAR and DGA.

Step 1: STM receives notification of an attack from a victim's IDS (Intrusion Detection System)
Step 2: STM sends the appropriate requests to SCARs
Step 3: SCARs in turn obtain copies of the digests and transformation tables from DGAs for the appropriate time period. After analyzing and correlating the tables, SCARs are able to figure out which routers in the region, if any, forwarded the packet.
Step 4: The SCAR can then reconstruct the path along which the packet traversed through the region and reports it to the STM. Based on this information, the STM is able to reconstruct the path through the network.

The limitation of the scheme is the timing issue. For high-rate interfaces, traceback must be performed within a very short period of time. The problem is magnified for the inter domain case when time synchronization cannot be expected. Also, such a strict timing constraint on traceback prohibits post-mortem traceback (i.e., long after the attack has finished). This becomes important when the victim does not realize it is being attacked, or cannot contact the STM during the attack for some reason.

ICMP traceback: This scheme determines the full path of the attack. This approach was originally introduced by Bellovin (2000). Every router in Fig. 2 is configured to pick a packet statistically (1 in every 20,000 packets recommended) and generate an ICMP traceback message or iTrace directed to the same destination as the selected packet. The iTrace message itself consists of the next and previous hop information and a time stamp. As many bytes of the traced packet as possible are also copied in the payload of iTrace (Bellovin, 2000). The time to live (TTL) field is set to 255 and is then used to identify the actual path of the attack. The routers on the path generate a new packet with an iTrace message. By assuming the victim is under (D)DoS attack and therefore the volume of packets going to it is large, the victim will eventually get all the addresses of the routers on the attack path that implement iTrace. By using TTL fields, these addresses can be sorted to reconstruct the attack path. It was shown in (Mankin *et al*., 2001) that while this approach is efficient and reasonably protected, the chance of receiving a useful iTrace is small if the victim undergoes a major DDoS attack, especially if the attack was carefully orchestrated with the goal of reducing the probability of useful iTraces.

The mechanism to resolve this statistical problem is to associate a weight or value with every iTrace generated. The value is affected by the distance from the victim, frequency of iTraces being sent to the victim and time since the attack began. Having these three contributors to the value of iTrace, the original proposal (Bellovin, 2000) was augmented by an algorithm to make a more educated choice of packet for iTrace. While introducing definite benefits, these augmentations somewhat complicate the algorithm and require a change to the forwarding table on every router implementing this scheme. To deploy the scheme, vendors need to implement two functions: iTrace and reconstruction. ICMP traceback will not be able to perform the traceback for a DDoS attack with a large number of reflectors. Therefore, the ability to handle major DDoS attacks is poor.

Overlay network: This scheme (Stone, 2000) introduces a Tracking Router (TR) in the network (R7 or R8 in Fig. 2). This TR monitors all traffic that passes through the network. In order to be able to monitor all of the traffic on the network, all packets have to be routed through this

TR. This is accomplished by building a Generic Route Encapsulation (GRE) tunnel from every edge router (In Fig. 2, R1, R2, R3, R4, R6, R9, R11 and R12 are edge routers) to this TR. Once the appropriate routing has been configured on the edge routers and TR, all traffic from an ingress edge router would travel over the GRE tunnel to the TR and then from the TR over another GRE tunnel to the egress edge router. While core routers carry the traffic, logically it is only one hop from an edge router directly to the TR. This architecture can be visualized as a star topology with the TR in the center and all of the edge routers on the network connecting to it with GRE tunnels. Since tunnels are built over the existing topology and utilize existing routing protocols, this star-like logical network is said to be an overlay network. In reality, of course, a single TR will not be able to handle the load of packets from the whole network. Therefore, it is physically a fully connected mesh of several TRs, which can still be logically thought of as a single TR. The TR will utilize signature-based intrusion detection. This is different from all the other schemes, where intrusion detection was a function of the victim. When an attack is detected, meaning a single packet or sequence of packets that constitutes an intrusive action, the origin of the attack can be identified because it is only one hop away. In order to deploy this scheme, no additional functionality needs to be developed by vendors. The scheme takes advantage of the features available on most routers today. On the other hand, ISP involvement in this scheme is large. The ISP has to perform a traceback as well as identify the attack completely on its own. Also, a number of TRs and IDS servers would have to be purchased by the ISP. ISP involvement is therefore high. By adding another edge router to the network, which results in configuring the TR to enable traceback on them. Limitation of this scheme is, it will only function well within a single administrative domain. In order for the overlay network to function well across ISPs, it would be necessary to somehow connect all of the TRs into a single system.

Probabilistic Packet Marking (PPM): Originally introduced by Savage *et al.* (2001) and it was improved by Song and Perrig (2001), with coding methods and security. This scheme is based on the idea that routers mark packets that pass through them with their addresses or a part of their addresses. Packets for marking are selected at random with some fixed probability of being selected. As the victim gets the marked packets, it can reconstruct the full path, even though the IP address of the attacker is spoofed. This scheme is aimed primarily at DoS and DDoS attacks as it needs many attack packets to reconstruct the full path. In Fig. 2, Attacker A initiates an attack to victim V. Assume that the path the packets take is R1-R2-R4-R9-R12. Each router implementing PPM accepts the stream of packets and before routing them probabilistically marks them with its partial address information (i.e., puts the router's partial address in the packet headers). Packets are marked with a marking probability p, which is suggested to be 0.04 in Savage *et al.* (2001). When the victim receives enough such packets, it can reconstruct the addresses of all the PPM-enabled routers along the attack path. Clearly, in order to reconstruct the full path the flow must contain a large number of packets. To deploy the scheme, vendors need to implement two functions: marking and reconstruction. Once the marking function is available, the software on all routers must be upgraded. Upgrade of the software on the routers is straightforward. Once the routers are upgraded, PPM needs to be enabled and that is the extent to which an ISP needs to get involved in the scheme; therefore, ISP involvement is low. Additional PPM-enabled routers can be added independently, which indicates good scalability.

The number of packets required for path reconstruction is measured in thousands for the original proposal in Savage *et al.* (2001) and decreases to just under 1000 packets for the improved scheme described in Song and Perrig (2001). For partial deployment to be effective, the victim must be aware of the network topology and routing on the network. Processing overhead in network elements is incurred for every packet. For each packet the decision is made if it should be marked or not by generating a random number. Additionally, if the packet is marked, more processing overhead is incurred associated with composing the mark and updating the ID field and Reserved Flag in that packet. The overhead associated with packet marking is minimal and should not require major upgrades to the router hardware. Major processing overhead will be incurred at the destination during reconstruction. Potentially, the victim could have to perform searches of data structures consisting of billions of entries. Reconstruction of data structures will require a large amount of memory as well.

However, as mentioned earlier, overhead and additional memory required at the potential victim is not a major setback. Bandwidth overhead for this scheme is zero since all traceback information is scrambled in the IP packet header. Both schemes described in Savage *et al.* (2001) and Song and Perrig (2001) are unable to perform traceback for a major DDoS attack with a large number of reflectors. Traceback with a PPM-like scheme is capable of tracing only a limited number of reflectors.

Proposed method: This scheme also comes under packet marking scheme. This method is based on Stone (2000). In general, there are two types of routers in an ISP domain: internal routers and external routers. Internal

routers belong to the ISP domain and external routers belong to the customers or another ISP. Internal routers are divided into two types, based on location, namely, edge routers and transit routers. Edge routers are internal routers that are adjacent to one or more external routers. Transit routers are internal routers that are only adjacent to other internal routers. In Fig. 2, it is assumed as a single ISP domain in which R1, R2, R3, R4, R6, R9 and R12 are edge routers and R5, R7, R8 and R10 are transit routers. This scheme is to prevent the attack at the nearest point to the source of attack in a single ISP domain. This system involves a Controller and Agent model. In each ISP domain, we envisage that there exists a controller, which is a trusted entity and is involved in the management of denial of service attacks. In principle, the controller can be implemented on any internal (transit or edge) router or at a dedicated host. The agents are implemented on all the edge routers in the ISP domain. Only the controller has the information about all the agents that are present in the domain. An agent has only the information of its domain controller. Both controller and agents are designed to handle the attacks on multiple victims simultaneously. It is assumed that the controller is always available and any host is able to contact the controller at any time and no internal routers are compromised. Each agent in the domain is assigned a 10 bit agent ID against its 32 bit IP address and a 6 bit controller ID that is used. So that, a maximum of 1024 (2^{10}) agent IDs only can be assigned in a single ISP domain. This information is marked in the identification field of the packet.

Controller mechanism: The controller can be implemented on a dedicated host or on any one of the transit routers in a domain. It responds to the victim's request and sends commands to its agents. However in our case, there is no need for the controller to identify the attack. Hence the implementation of the controller is much simpler. The controller can identify its agents using the unique agents' IDs and all the agents can identify its controller with the controller ID.

There are two important commands, from the controller to its agents.

- The first command is issued to all the agents in the ISP domain, when Controller receives a request from the victim to mark agent ID in the packet. This includes the 32 bit IP address of the victim, 6 bit controller ID, which is same for all the agents.
- The second command is issued when the victim identifies the attack signature based on the agent ID and requests the controller to prevent the attack at the identified agent. The controller retrieves the 32 bit IP address of the agent based on the 10 bit agent ID and commands the particular agent to prevent the traffic that is matching the attack signature.

Agent mechanism: The agents are implemented on all edge routers in a ISP domain. The functions of edge routers will not be disturbed during normal time but the special agent mechanism will be invoked when it receives a command from its controller during attack. Algorithm-1 shows the implementation of the agent mechanism. When the agent receives the command from its controller, only the victim's traffic is filtered. There are two important stages of agent mechanism.

- The first stage is invoked when the agent receives a command from its controller to mark the traffic to the victim. The command from the controller gives the information of the victim's address and controller ID. Now the agent dynamically applies a filter with the destination address of the victim and marks the packet with the controller ID and agent ID in the identification field. Packets are marked only by the agent or by the first agent that sees the traffic to the victim. Packets marked by the attacker are identified and eliminated at this stage.
- The second stage is invoked when the agent receives a command from its controller to prevent the attack traffic to the victim. In this stage, the agent drops all the packets that match the attack signature. All the packets that do not match the attack signature are again marked with the controller ID and the agent ID and are destined to the victim. The main reason for marking the packets even if it does not match the attack signature is to enable the victim to identify if there are any changes in the traffic pattern for itself and generate a quick response.

Algorithm -1

Mark_Packet(victim ID)

Step 1: Select the packet whose destination address is same as victim ID

Step 2: Check if *packet already* marked, then drop.

Step 3: else mark *packet* with controller ID and agent ID.

Block_Attack_Traffic(victim ID, Source ID)

- if *packet* matches with Attack Signature then drop
- else MarkPacket(victim address).

System operation: Controller has to maintain a database about all agent IDs. Since agents are deployed on all the edge routers, all the traffic to the victim are marked with the controller ID and agent ID in the identification field of the packet. Since the victim knows the controller ID and agent IDs, it can easily identify different attack signatures. The victim sends request for marking the packets which are reaching to it. Then controller sends command to all its agents to mark the packets which have victim address as a destination address. All agents will mark its 6 bit controller ID and its ID (10 bits) in the identification field of the packet. Once the victim identifies the attack signature based on the controller ID and agent ID, it sends request message to the controller to prevent attack traffic. After receiving the request from the victim, the controller retrieves the 32 bit IP address of the agent from its database, based on the agent ID and sends command to the agent to prevent the attack traffic. The agent who receives this command will start preventing the traffic that matches the attack signature from reaching the victim. Only the traffic that is matching with the attack signature will be dropped at the agent. As the packets are dropped at the edge router, the traffic congestion gets reduced in the domain thereby increasing traffic speed. The traffic that does not match the attack signature will be marked with the controller ID and agent ID. This is to enable the victim to easily track the changes in attack traffic. Prevention will be done until the agent receives a reset signal from its controller. However the packets will be marked for an excess amount of time. This is very useful for intermittent type of attacks, where attacking systems do not flood the victim continuously, but send attack traffic at regular intervals.

The system operates in two ways, Random Packet marking and On-demand Packet marking. In random packet marking, all the agents mark the controller ID and agent ID to all packets in regular intervals. In this method all edge routers are considerably overloaded by packet marking function. In on-demand packet marking method, all the agents mark controller ID and agent ID in the identification field of packets which send traffic to victim. When there is no attack, the functions of controller and agents are identical. Approximate source can be traced with a single packet and since all the packets are marked at ingress agent, once attack signatures are identified, prevention of attack traffic is directly near to the attacking source (ingress agent) instead of hop by hop. Each agent needs to check and prevent only the attack signature passing through it. Only one packet is enough to identify the approximate source of the packet because it is not to trace the path of the packet. In Fig. 2, router R1 only marks its agent ID and sends to destination. Victim sends request to controller to identify the edge router using agent ID. Controller will identify the nearest source of the packet by extracting IP address against the given agent ID. In this proposed method we can also block the packets against attack signature.

This proposed scheme possesses the following merits.

- It is easy to implement
- Packet marking by only one router
- No computation in the victim side to trace the packets
- Single packet is enough to trace
- It is suitable of DDoS attacks
- It has low processing and no bandwidth overhead and
- It is scalable (i.e. the amount of additional configuration on other devices needed to add a single device to the scheme)

The limitation of the proposed system is to traceback and eliminate the spoofed packets within a ISP domain only and ISP involvement is more.

RESULTS AND DISCUSSION

The proposed method can trace the source address of IP packet even with single packet and without any computation by victim. It is very simple to implement. The limitations of the proposed system are, more ISP involvement is needed and it can trace the source of IP packets in a single ISP domain. Table 1 provides a summary of the evaluation and offers a comparison of IP traceback techniques.

Table 1: Comparison of traceback systems

	ISP Involvement	No. of packets required	Memory requirements		Network processing overheads	Victim process overheads	Ability to handle DDoS attacks
			Network	Victim			
Hash-based	Fair	1	Fair	None	Low	None	Good
ICMP	Low	Thousands	Low	High	More	High	Poor
Overlay	More	1	Low	None	More	None	Good
PPM	Low	Thousands	None	High	More	High	Poor
Proposed system	More	1	Low	None	Low	None	Good

CONCLUSION

In this study, the state of the art approaches in IP traceback are discussed and the proposed method is compared with them. We know that it is very complicated to develop any system that possesses all the qualities of an ideal scheme. Solutions to a problem are rarely ideal, since each and every method has its own merits and demerits.

REFERENCES

Bellovin, S.M., 2000, ICMP traceback messages. IETF draft,http://www.research.att.com/smb/papers/draftbellovin- itrace-00.txt.

CERT, 1996, (Computer Emergency Response Team), CERT Advisory CA-92.21: TCP SYN Flooding and IP Spoofing Attacks. http://www.cert.org/advisories/CA-96.21.ping.html, 1996.

CERT, 1998, (Computer Emergency Response Team), CERT Advisory CA-98.01: Smurf IP Denial-of-Service Attack via pings. http://www.cert.org/advisories/CA-98.01.smurf.html, Jan1998.

Kevin, J.H. and G.M. Weaver, 2001, Trends in Denial of Service Attack Technology. CERT Advisory

Mankin, A. *et al*., 2001. On Design and Evaluation of Intention- Driven' ICMP Traceback. Proc. IEEE Intl. Conf. Computer Comm. and Networks, IEEE CS Press, pp: 159-165.

Savage, S. *et al*., 2001, Network Support for IP Traceback. IEEE/ACM Trans. Networking, 9: 226-237.

Snoeren, A.C. *et al*., 2002 Single-Packet IP Traceback. IEEE/ACM Trans. Networking, 10: 721-734.

Song, D.X. and A. Perrig, 2001, Advanced and Authenticated Marking Schemes for IP Traceback. Proc. IEEE INFOCOM, IEEE CS Press, 2: 878-886.

Stone, R., 2000, Center track: An IP overlay network for tracking dos floods. Proc. 9th Usenix Security Symp., Usenix Assoc., pp: 199-212.