Defeating DoS Attacks with IP Traceback[1]

Rafael P. Laufer* — Pedro B. Velloso — Otto Carlos M. B. Duarte***

**GTA – Universidade Federal do Rio de Janeiro*
P.O. Box 68504
21945-970 Rio de Janeiro, RJ
Brazil
rlaufer@gta.ufrj.br
otto@gta.ufrj.br

***LIP6 – University of Paris 6*
8 rue du Capitaine Scott
75015 Paris
France
pedro.velloso@lip6.fr

ABSTRACT: *On several denial-of-service attacks, packets with spoofed source addresses are employed in order to disguise the true origin of the attacker. A defense strategy is to trace attack packets back to their actual source in order to make the attacker accountable and isolate him from the network. To date, several traceback systems have been proposed and evaluated. In this paper, we present some basic anonymous denial-of-service attacks and summarize existing solutions to the IP traceback problem. Additionally, we briefly introduce our proposed IP traceback scheme designed to perform the traceback from a single attack packet and without state storage in the network infrastructure.*

KEY WORDS: *management, security, denial of service (DoS), IP traceback.*

1. Introduction

The current routing infrastructure is vulnerable to anonymous denial-of-service (DoS) attacks (CERT, 1996a), (CERT, 1996b), (CERT, 1998), (CERT, 2000b), (CERT, 2003). Such attacks are specially designed to conceal the true identity of the attacker in addition to making the services provided by the victim inaccessible to

[1] This work has been supported by CNPq, CAPES, FAPERJ, FINEP, RNP, and FUNTTEL.

users. These attacks are generally conducted by sending packets to the victim at a higher rate than they can be served, causing the denial of legitimate service requests. In distributed denial-of-service (DDoS) attacks, the aggregate traffic from several different sources is responsible for disabling the services provided by the victim. Recently, the number of distributed attacks against famous websites is alarming and digital plagues have been specifically developed for that malicious purpose (CERT, 1999), (CERT, 2000a), (CERT, 2003), (Symantec, 2004), (Gibson, 2001). Although less common, denial-of-service attacks constituted by a single packet also exist and are much easier to be conducted (CERT, 1996c), (CERT, 1997). In both cases, the results are financially devastating (Garber, 2000) and a solution that identifies the true origin of attack packets becomes necessary.

Due to the datagram technique employed in the IP protocol, the attacker can easily inject packets with spoofed source addresses into the network and remain anonymous throughout the attack. In fact, there is no entity or mechanism responsible for verifying the authenticity of the source. Once the routing infrastructure is exclusively based on the destination address, packets with spoofed source addresses generally reach the victim without difficulty. Denial-of-service attacks can also become anonymous due to the stateless nature of IP routing. Currently, no information about forwarded packets is stored in routers for future queries and, as consequence, it is not possible to deduce the path route traversed by a spoofed attack packet.

Indirect attacks can also be conducted to hide the tracks of actual perpetrators. Such attacks employ one or more intermediate machines between the attacker and the victim. The role of these machines in the attack can be classified into active or passive, depending on their behavior. In active mode, intermediate machines are first compromised by viruses, e-mail worms or attackers themselves. Those machines then act as “zombies”, being remotely controlled by a master machine responsible for coordinating the attack. Once received the order, the zombies send a steady stream of traffic to the targeted server (Roberts, 2002). In distributed attacks, hierarchical networks constituted by several zombies may be formed (Dittrich, 1999). On the other hand, in passive mode, those machines just act as “reflectors” of the attack traffic. The attacker sends false requests on behalf of the victim to these reflectors who, without noticing that the source address

is spoofed, innocently respond towards the victim (Chang, 2002). In addition, if the size of the response is bigger than the size of the request, the reflectors also act as "amplifiers" of the attack traffic.

A complete solution for every kind of denial-of-service attack is somewhat complex. Hence, more restricted solutions were proposed. Their purpose is only to identify the machines that directly generate attack traffic and the respective network path traversed by such traffic, regarded as the IP traceback problem (Savage, 2001). The limitation of such solutions, however, is that identified machines might not be the actual attack sources. In fact, only zombies may be recognized and, therefore, more sophisticated schemes are required to locate the true origin of the attack. Nevertheless, even without a perfect identification of the attacker, partial information about the traversed network path is still useful to control the unnecessary traffic at a node closer to the source.

In this paper, we present the current state of art in IP traceback, regarding the different approaches proposed so far. We classify each traceback scheme according to its well-known characteristics and similarities. Basically, we split traceback schemes into two large groups: stateless and auditing-based IP traceback. The latter group is subsequently divided according to end-host or network infrastructure storage. We then explain how a few denial-of-service attacks are conducted against targeted servers and why current preventive measures are not enough in defeating DoS attacks. Additionally, we outline our proposed IP traceback scheme designed to perform the traceback from a single attack packet without requiring state storage in the network infrastructure.

The rest of the paper is structured in the following way. First, in Section 2, we present basic denial-of-service attacks. Preventive measures against DoS attacks are presented in Section 3. We then explain the IP traceback problem in Section 4. In Section 5, we present the different approaches to the IP traceback problem that have been already proposed. Section 6 briefly presents our approach to the IP traceback problem. Finally, conclusions about this work are presented in Section 7.

2. Anonymous Denial-of-Service Attacks

A few well-known anonymous DoS attacks are launched using packets with spoofed source addresses. For instance, SYN flooding attacks are conducted by sending a steady stream of TCP SYN packets with unreachable source addresses towards the targeted server or victim. For each new connection request, the victim innocently reserves local resources, responds with a SYN/ACK packet and waits for a nevercoming response. Eventually, the operating-system table becomes filled with half-opened connections and no more connection requests are accepted. Therefore, service is denied to legitimate users.

It is also possible to launch denial-of-service attacks through UDP flooding. This kind of attack can disable two victims by making them continuously send traffic to each other. One way of doing this is to send a packet apparently coming from the `echo` service of victim V_1 to the `chargen` service of victim V_2, as depicted in Figure 1. Victim V_2 then responds back to V_1, who again responds to V_2, creating a loop. Consequently, the attack packet will never leave the network. By sending a lot of these everlasting packets to one of the end-host systems, the attacker may exhaust bandwidth resources or even overload the victims.

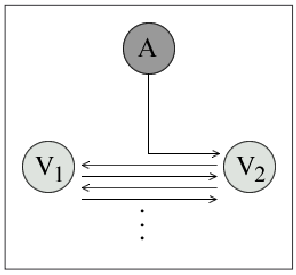

Figure 1. *UDP flooding: the attacker* A *sends packets to the victim* V_2 *on behalf of victim* V_1 *to provoke an endless exchange of traffic between both victims.*

Another well-known denial-of-service condition is achieved through ICMP. This attack is conducted by sending an ICMP `ping` packet on behalf of the victim to the broadcast IP address of a

"reflector" network. Each host of this network then responds with an ICMP response message addressed to the victim. By sending a lot of these attack packets to reflector networks, the attacker causes the victim to be flooded by ICMP messages. Clearly, the reflector network is an "amplifier" of the attack traffic since, for each packet sent by the attacker, several packets are sent towards the victim. Figure 2 briefly demonstrates this concept. Attacker *A* sends a single ICMP `ping` packet on behalf of the victim *V* to the broadcast IP address of a reflector network composed of five hosts H_1, H_2, H_3, H_4, and H_5. Each of the hosts then replies with another ICMP message directed to the victim.

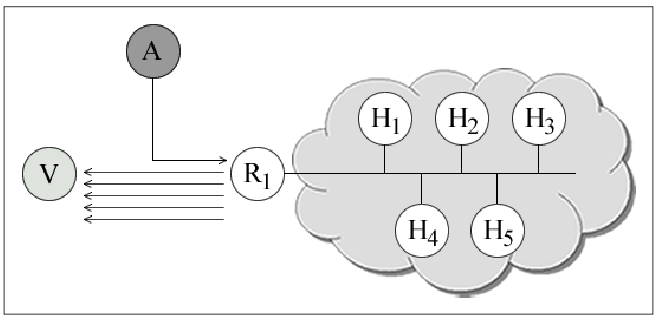

Figure 2. *ICMP flooding: the attacker* A *sends* `ping` *packets to the reflector network on behalf of victim* V *to provoke a flooding addressed to the victim.*

3. Preventive Measures

The ingress filtering technique (Ferguson and Senie, 2000) was introduced to avoid the traffic of packets with spoofed source addresses in the network. Implemented in some routers, ingress filtering restricts the routing of traffic that originates from a downstream network to only well-known and advertised prefixes. Equivalently, a router must drop any packet whose source address does not belong to one of such advertised networks. Figure 3 depicts a simple network where ingress filtering is used against source-address spoofing. For convenience, only private IP addresses are used. With

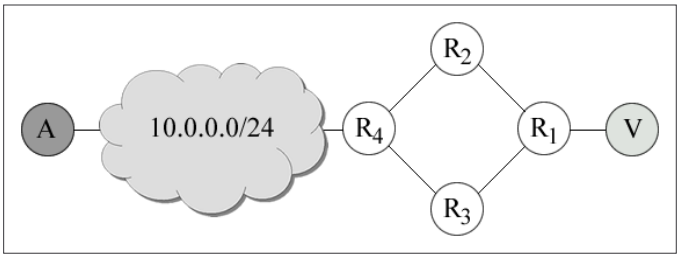

Figure 3. *Ingress filtering is used at router* R_4 *to prohibit the attacker from using a source IP address residing outside the* `10.0.0.0/24` *prefix.*

ingress filtering, router R_4 drops any packet coming from subnetwork spoofed source addresses to the victim V. The spoofed source address, however, must reside inside the `10.0.0.0/24` prefix. For instance, the IP address of a neighbor machine could be used as the source address of attack packets. In addition, there is an undesirable dependency between security of end hosts and universal deployment of this technique. Since the filtering directly affects the routing process, inspecting the source address of every packet may also require additional resources from routers. Further, some technologies, such as Mobile IP (Perkins, 2002), legitimately employ spoofed source addresses and could also be affected.

A protection scheme has also been proposed to protect a server from SYN flooding attacks (Belenky and Ansari, 2003). Basically, the scheme keeps track of half-opened TCP connections at a particular server. The tracking is not necessarily implemented on end servers; it can also be implemented on routers and firewalls, for instance. When the number of these connections exceeds a threshold, either new connection requests are blocked, or old half-opened connections are closed in order to make room for new connections. This scheme, however, is specifically designed for this kind of attack and does not provide any information about real perpetrators.

4. IP Traceback

In this section, we briefly explain the definition of the IP traceback problem as it was introduced by (Savage *et al.*, 2001). Figure 4 shows

an example network as seen from a victim V. Every attacker A_i is a leaf node that generates attack traffic towards the victim. The dotted line denotes the network path of the attack traffic sent by attacker A_2. Internal nodes R_i represent routers along a path between the victim V and some attacker A_i. The set of routers R_i is also referred as the *upstream routers* from V throughout this paper. An *attack path* is an ordered list of routers between an attacker A_i and the victim V. For example, in Figure 4, the attack path (R_5, R_2, R_1) is depicted by the dotted line. By grafting together the attack paths of every attacker, an *attack graph* is composed. Additionally, a router is named a *false positive* if it is in the reconstructed attack graph but it is not in the real attack graph. Similarly, a router is named a *false negative* if it is not in the reconstructed attack graph but it is in the real attack graph.

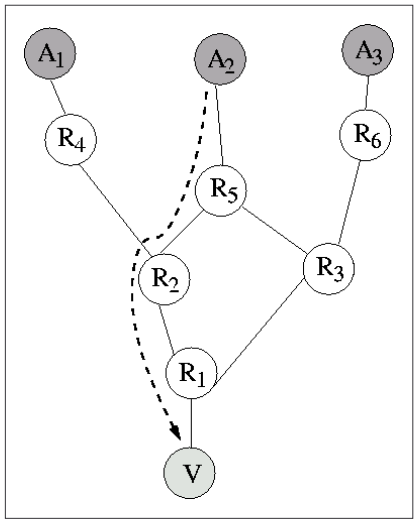

Figure 4. *Network as seen from the victim* V.

According to the definition, the exact traceback problem is to correctly determine the attack graph. Nevertheless, on most IP traceback schemes, it is very hard to restrict the interference of the attacker in the traceback procedure. Therefore, a more restricted problem is postulated. The approximate traceback problem is defined as finding an attack path that contains the real attack path as a suffix.

For instance, (R_6, R_5, R_2, R_1) is a valid solution to the approximate traceback problem since it contains the actual attack path (R_5, R_2, R_1) as a suffix. Further, a solution is named robust if the attacker cannot prevent the victim from finding out an attack path with the actual attack path as a suffix.

5. IP Traceback Schemes

There are two basic approaches to the IP traceback problem. The first one tries to determine the actual perpetrators during an ongoing attack in a stateless way. The second one collects audit trails during the attack to enable backtracing after the attack is finished, among other advantages. We now present proposed IP traceback schemes, according to this classification.

5.1. *Stateless IP Traceback*

Input debugging (Stone, 2000) is an intuitive strategy to trace an ongoing attack back to its source. First, the victim must identify an attack "signature", which is some common characteristic observed in every attack packet. From the signature, network operators can determine the interface from which the attack traffic comes in the nearest router to the victim. By reversely resolving the MAC address of the received packets, the adjacent upstream router is identified. This process, recursively repeated on every upstream router, reveals the network path to the attacker or to the border of another ISP. In the latter case, this second ISP must be contacted to continue the traceback process. Although simple, this technique is highly dependent on communication and cooperation of ISPs to successfully identify the attacker. In addition, the attack must be long enough in order to complete the traceback process, once it cannot be done after the attack is finished. A system called CenterTrack employs this technique by using tunnels (Stone, 2000). After detecting the attack, all the victim traffic is redirected to a center traceback router, with which other routers already hold previous established tunnels. From the center router, input debugging is used to discover from which tunnel the packet came. Beyond the already mentioned disadvantages,

source routing techniques (Postel, 1981) can also be used to prevent packets from traversing the center router.

Under another perspective, Burch and Cheswick (Burch and Cheswick, 2000) developed a traceback system based on link testing. Its operation is based on observing the attack traffic received by the victim while flooding each link with short bursts of traffic. First, every incoming link of the nearest router to the victim is tested, as depicted in router R_1 of Figure 5. When a link is flooded and the attack traffic received by the victim decreases, this link is considered a component of the attack path. In the example of Figure 5, when the link between R_1 and R_3 is flooded, the attack traffic rate drops at the victim. Therefore, this link is considered to be in the attack path and router R_3 is the next one to be tested. The process is then recursively repeated in each upstream router until the attack path (R_{12}, R_{10}, R_8, R_3, R_1) is identified.

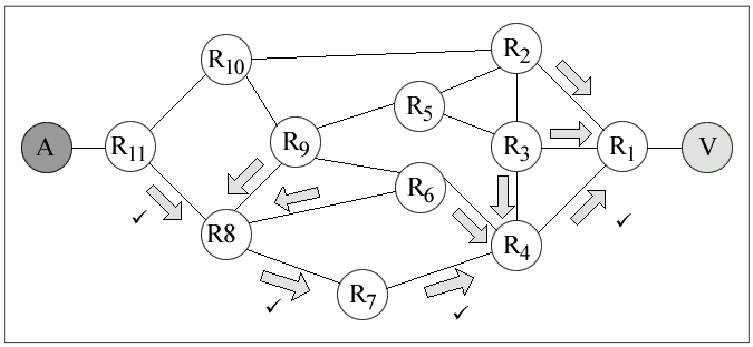

Figure 5. *Each router floods its links with short bursts of traffic expecting to interfere with the attack traffic at the victim.*

Although it is not necessary any intervention from network operators, this technique requires an accurate map of upstream routers as well as nodes willing to be regularly flooded. In addition, the entire system is based on the `chargen` service, which is disabled by default (Belenky and Ansari, 2003). As in the previous techniques, it is possible that the identification of the attackers is compromised if the attack suddenly stops while performing the traceback. Further, it is assumed that during the attack the links are heavily loaded. In fact, the

variation observed in the victim for distributed or short-flow attacks may be totally imperceptible.

5.2. *Auditing-based IP Traceback*

The advantage of the previous approaches is the lack of state. Nevertheless, as a consequence, it is impossible to trace the origin of an attack after it has completed. Therefore, another approach considers creating auditing information as packets traverse the network. Such information can be later used to reconstruct the attack path. Different proposals suggest that auditing information be placed either in the victim or in the network routers.

5.2.1. *End-host Storage*

A simple scheme for backtracing the true origin of a packet is to include the IP address of every traversed router in the packet. The idea is pretty much similar to the IP Record Route option (Postel, 1981). As a consequence, every packet received by the victim contains the entire attack path. This scheme is able to trace the origin of an attack with a single packet, providing the high scalability required in distributed attacks. Nevertheless, it cannot be qualified as an ideal IP traceback scheme (Savage *et al.*, 2001). First, appending data to the packets in flight implies significant additional processing overhead. Furthermore, by increasing the packet size on every hop, unnecessary fragmentations may happen and routers may be overloaded.

Savage *et al.* introduce an auditing-based system, where the required information to perform backtracing is located at the victim (Savage *et al.*, 2001). In summary, routers overload rarely used IP header fields to notify the victim of their presence in the attack path. Every router probabilistically inserts partial information about itself in the packets routed to the victim, as depicted in Figure 6. With probability p, router R_n inserts information about itself in the packet, as denoted by the dark packet. After receiving enough attack packets, the victim can reconstruct the entire attack path. To reduce router overhead and required per-packet space, sampling and encoding techniques are used. Although innovative, this proposal requires high computational effort during reconstruction and generates several false positives, even for small-scale distributed attacks (Song and Perrig,

2001). Analysis of Park *et al.* also show the vulnerability of the system to marking-field spoofing (Park and Lee, 2001). For instance, considering the marking probability of 0.04 suggested by Savage *et al.* and an attack path of 10 routers, the probability that the victim receives a packet with the marking field spoofed by the attacker is approximately $(1\text{-}0.04)^{10} \approx 66\%$.

A similar traceback scheme is proposed by (Bellovin *et al.*, 2003). Whenever routing a packet, routers probabilistically send to the victim an ICMP packet with information about themselves, their adjacent

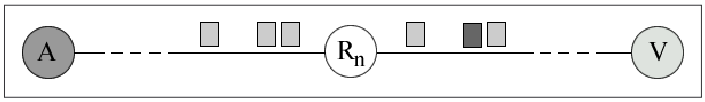

Figure 6. *Routers probabilistically mark routed packets with information about themselves to notify the victim of their presence in the attack path.*

routers and the sampled packet. The dark packet of Figure 7 represents the extra packet carrying traceback information. For a long packet flow, the victim can use the received data to reconstruct the attack path. Nevertheless, since auditing information is sent in additional router-generated packets, messages must be authenticated to avoid forgery. Therefore, the adoption of a public-key infrastructure between every potential victim and candidate routers is mandatory. Mankin *et al.* extend the work of Bellovin by introducing some new concepts, such as "utility" and "value" of traceback messages, as well as a proposal to improve them (Mankin *et al.*, 2001).

Another auditing-based system proposed by Dean *et al*. considers probabilistic and algebraic techniques (Dean *et al.*, 2002). Their basic idea is that each packet carries a representation of the attack path as a result of a well-known polynomial, whose unknown variables are the router IP addresses. If the victim receives enough packets of the same route, an equation system can be derived and solved with unique solution. Nevertheless, as the proposed algorithms are based on

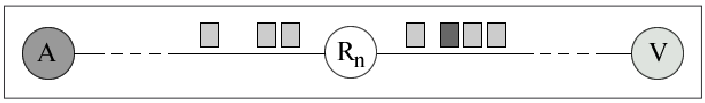

Figure 7. *Routers probabilistically generate an extra packet with information about themselves and adjacent routers and send it to the victim.*

probabilistic marking, attackers can violate the system in the same way mentioned before (Park and Lee, 2001).

Another auditing-based system proposed by Dean *et al*. considers probabilistic and algebraic techniques (Dean *et al.*, 2002). Their basic idea is that each packet carries a representation of the attack path as a result of a well-known polynomial, whose unknown variables are the router IP addresses. If the victim receives enough packets of the same route, an equation system can be derived and solved with unique solution. Nevertheless, as the proposed algorithms are based on probabilistic marking, attackers can violate the system in the same way mentioned before (Park and Lee, 2001).

5.2.2. Network Infrastructure Storage

Under the perspective of storing auditing information in the network infrastructure, the simplest way to collect auditing traces is by every router logging all packets that traverse it (Stone, 2000). Although quite simple, excessive resources are required for data storage and data mining. For instance, storing every packet of a saturated OC-24 link (1.244 Gbps) would require the exorbitant amount of 9.3 GB per minute or, equivalently, 13.4 TB per day. In addition, a compromised router may cause privacy problems once it contains information about every routed packet. Instead, an alternative logging procedure that stores only some bytes of the packet header could be used to avoid privacy problems.

An alternative to reduce the amount of stored information is to use Bloom filters (Bloom, 1970). Recently, these filters have been widely used in computer networks (Broder and Mitzenmacher, 2002). Snoeren *et al.* propose a mechanism that traces a single IP packet without storing all routed traffic (Snoeren *et al.*, 2002). Instead, only packet digests are stored in Bloom filters located at router-connected

devices, as depicted in Figure 8. The digests of every packet routed by R_n are stored in a Bloom filter to save space. Periodically, saturated filters are stored for future queries and replaced by new empty filters. To later determine if a packet traversed the router, its filter is simply checked. A recursive process is executed by each router to reconstruct the attack path to its true origin. Nevertheless, even with Bloom filters, such system demands high-capacity storage devices depending on the capacity of the links. Improvements proposed by Li *et al.* drastically reduce the space required for data storage even though the capability of tracing a single packet is compromised (Li *et al.*, 2004).

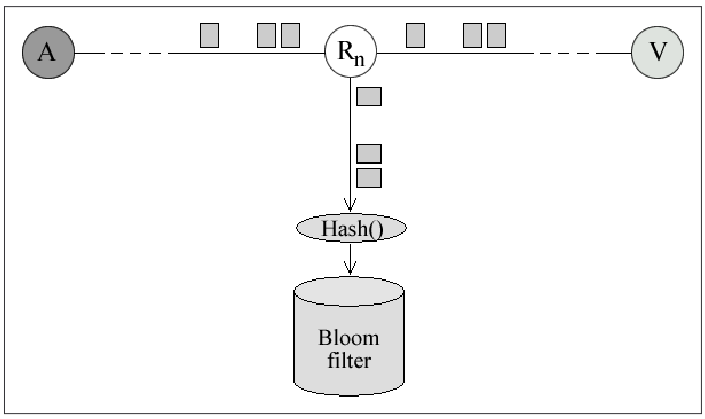

Figure 8. *Each router stores packet digests in Bloom filters for future queries.*

6. A Stateless Single-Packet IP Traceback Scheme

In this section, we briefly present our IP traceback scheme designed to trace the source of each individual packet. The proposal is based on the packet-marking approach to avoid storing state at routers. Instead of inserting its entire IP address into the packet, each node only inserts the digests of its IP address to indicate its presence on the path. In order to reduce the required space on each packet and to avoid the processing cost of appending data to packets, the attack route is stored in a built-in Bloom filter (Bloom, 1970) integrated into the packet. The filter is employed to avoid packet fragmentation by reducing and limiting the size of information inserted into the packet.

In addition, we propose the use of a generalized Bloom filter to prevent digest forgery by the attacker and therefore backtracing failures.

The marking algorithm for this case is quite simple. Just before forwarding a packet, the router inserts the IP address of its output interface into the filter of the packet. Figure 9 depicts a router inserting its output-interface IP address into the forwarded packets. Upon receiving an attack packet, the victim disposes of a filter whose elements are the routers that compose the attack path. To reconstruct the attack path, the following procedure is used. Initially, the victim checks for the presence of all neighbor routers in the Bloom filter of the received attack packet. The router that is recognized as an element of the filter is identified as the upstream router and is therefore integrated into the attack path. Figure 10 depicts this process. Afterwards, this selected router receives from the victim a request to continue the reconstruction procedure along with the respective Bloom filter. It then verifies its authenticity and checks which neighbor router is also recognized as an element of the filter, identifying the next upstream router. This process is recursively repeated on each upstream router to reconstruct the actual path traversed by the packet. When a router does not recognize any neighbor router as an element of the filter, the process stops and this router may be considered the source of the attack.

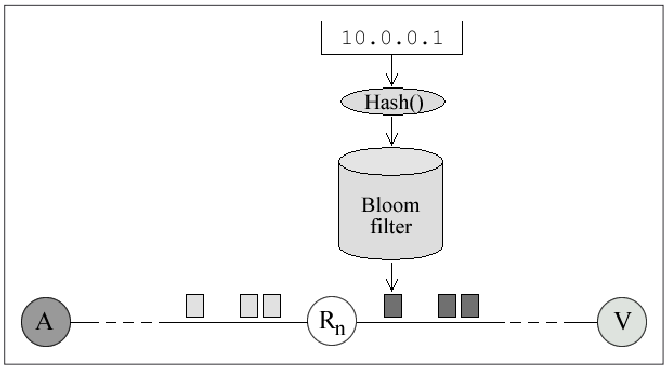

Figure 9. *Router* R_n *inserts the IP address* `10.0.0.1` *of its output interface into the Bloom filter of routed packets that leave this interface. The victim then receives a filter containing the IP addresses of every router in the attack path.*

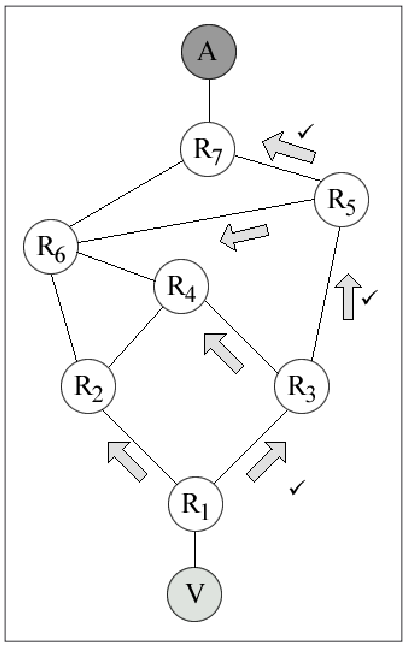

Figure 10. *At the reconstruction procedure, routers check if adjacent routers are also inserted into the Bloom filter of the victim. The ones recognized as elements of the filter continue the traceback procedure.*

Some advantages come from the adoption of this approach. First, the complete route of each packet can be individually determined. Such behavior is idealized by every IP traceback system since it permits the identification of each source of a distributed attack, even if it contributed with only one packet. By enabling backtracing of a single packet, the system becomes as scalable as it can be. Besides, no information needs to be stored in the network infrastructure. All traceback data is stored at the victim, who chooses to hold it or not according to the local security policy. Another advantage is the ability of tracing an attack long after it is over and without any help from network operators.

On the other hand, additional processing overhead is introduced during each packet routing. Moreover, the adoption of a Bloom filter introduces false positives into the attack path. During the reconstruction procedure, a false positive implies the incorrect

integration of a router into the attack path. If this probability is small enough, the occurrence of false positives does not significantly impact on the reconstruction. There would be some concurrent routes for the same packet but the set of possible attackers would still be reduced. Nevertheless, since the attacker controls the initial content of the packet, he can fill all the filter bits with 1. By saturating the filter, every router is integrated into the attack path during the reconstruction procedure, making impractical to distinguish the real path.

In order to minimize misleading techniques and to make the system less dependent of the initial state of the filter, we propose a generalization of the Bloom filter. With the generalized Bloom filter, the false-positive probability is reduced and it does not depend so much on the initial condition of the filter. On the other hand, false negatives, which do not exist in standard Bloom filters, are introduced with this generalization.

7. Conclusion

Denial-of-services attacks can cause significant damage to web service providers. Currently, the Internet routing infrastructure does not provide means of locating the attacker nor avoiding such attacks. The rapid growth of denial-of-services attacks led to a great number of proposed solutions. One popular solution is known as IP traceback, which detects the origin of the attack by tracing packets back to their actual source.

This paper presents the main proposed solutions to the IP traceback problem and analyzes the advantages and drawbacks of each different approach. Furthermore, we introduce a new approach to packet-marking IP traceback. Probabilistically, the proposed system is able to trace an attack back to its source by analyzing a single packet. When traversing the network, packets are marked with node digests instead of full IP addresses. Upon receiving a packet, the victim disposes of a representation of the attack path. A built-in Bloom filter is employed in the packet header in order to store node digests in a compact and fixed-size form. Nevertheless, the performance of Bloom filters is highly dependable on their initial condition, which is in control of the attacker. Given the current context, we believe that a lot of effort can

still be done to considerably reduce the interference of the attacker in IP traceback.

8. References

A. Belenky and N. Ansari. On IP Traceback. *IEEE Communications Magazine*, vol. 41, no. 7, p. 142─153, Jul. 2003.

S. M. Bellovin, M. D. Leech, and T. Taylor. ICMP Traceback Messages. Internet Draft: draft-ietf-itrace-04.txt, Aug. 2003.

B. H. Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Communications of the ACM*, vol. 7, no. 13, p. 442─426, Jul. 1970.

A. Broder and M. Mitzenmacher. Network Applications of Bloom Filters: A Survey. Technical report, Harvard University, 2002.

H. Burch and B. Cheswick. Tracing Anonymous Packets to their Approximate Source. In USENIX LISA'00, p. 319─327, New Orleans, Lousiana, EUA, Dec. 2000.

CERT. CERT Advisory CA-1996-01 UDP Port Denial-of-Service Attack, Feb. 1996a. Available: http://www.cert.org/advisories/CA-1996-01.html.

CERT. CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks, Sept. 1996b. Available: http://www.cert.org/advisories/CA-1996-21.html.

CERT. CERT Advisory CA-1996-26 Denial-of-Service Attack via ping, Dec. 1996c. Available: http://www.cert.org/advisories/CA-1996-26.html.

CERT. CERT Advisory CA-1997-28 IP Denial-of-Service Attacks, Dec. 1997. Available: http://www.cert.org/advisories/CA-1997-28.html.

CERT. CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks, Jan. 1998. Available: http://www.cert.org/advisories/CA-1998-01.html.

CERT. CERT Advisory CA-1999-17 Denial-of-Service Tools, Dec. 1999. Available: http://www.cert.org/advisories/CA-1999-17.html.

CERT. CERT Advisory CA-2000-01 Denial-of-Service Developments, Jan. 2000a. Available: http://www.cert.org/advisories/CA-2000-01.html.

CERT. CERT Advisory CA-2000-21 Denial-of-Service Vulnerabilities in TCP/IP Stacks, Nov. 2000b. Available: http://www.cert.org/advisories/CA-2000-21.html.

CERT. CERT Advisory CA-2003-20 W32/Blaster worm, Aug. 2003. Available: http://www.cert.org/advisories/CA-2003-20.html.

R. K. C. Chang. “Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial”. *IEEE Communications Magazine*, vol. 40, no. 10, p. 42─51, Oct. 2002.

D. Dean, M. Franklin, and A. Stubbleeld. “An Algebraic Approach to IP Traceback”. *ACM Transactions on Information and System Security*, vol. 5, no. 2, p. 119–137, May 2002.

D. Dittrich. “The DoS Project's ‘trinoo’ distributed denial-of-service attack tool”, Oct. 1999.

P. Ferguson and D. Senie. “Network Ingress Filtering:Defeating Denial of Service Attacks which Employ IP Source Address Spoofing”. RFC 2827, May 2000.

L. Garber, “Denial-of-Service Attacks Rip the Internet”, *IEEE Computer*, vol. 4, no. 33, p. 12–17, Apr. 2000.

S. Gibson. “The Strange Tale of the Attacks Against GRC.COM”. Gibson Research Corporation, Feb. 2001. Available: http://www.grc.com/dos/grcdos.htm.

J. Li, M. Sung, J. Xu, and L. Li. “Large-Scale IP Traceback in High-Speed Internet: Practical Techniques and Theoretical Foundation”. In Proceedings of the 25th IEEE Symposium on Security and Privacy, Oakland, California, EUA, May 2004.

A. Mankin, D. Massey, C. L. Wu, S. F. Wu, and L. Zhang. “On Design and Evaluation of ‘Intention-Driven’ ICMP Traceback”. In Proceedings of the IEEE ICCCN 2001 Conference, Scottsdale, Arizona, EUA, Oct. 2001.

K. Park and H. Lee. “On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack”. In Proceedings of the IEEE INFOCOM 2001 Conference, Anchorage, Alaska, EUA, Apr. 2001.

C. E. Perkins. “IP Mobility Support for IPv4”. RFC 3220, Jan. 2002.

J. Postel. “Internet Protocol”. RFC 791, Sept. 1981.

P. Roberts. “FBI Finds Source of Internet Attacks”. *PCWorld.com*, Nov. 2002.

S. Savage, D. Wetherall, A. Karlin, and T. Anderson. “Network Support for IP Traceback”. *IEEE/ACM Transactions on Networking*, vol. 9, no. 3, p. 226–237, Jun. 2001.

A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, and W. T. Strayer. “Single-Packet IP Traceback”. *IEEE/ACM Transactions on Networking*, vol. 10, no. 6, p. 721–734, Dec. 2002.

D. X. Song and A. Perrig. “Advanced and Authenticated Marking Schemes for IP Traceback”. In Proceedings of the IEEE INFOCOM 2001 Conference, Anchorage, Alaska, EUA, Apr. 2001.

R. Stone. “CenterTrack: An IP Overlay Network for Tracking DoS Floods”. In 9th USENIX Security Symposium, p. 199–212, Denver, Colorado, EUA, Aug. 2000.

Symantec Security Response. “W32.Mydoom.F@mm”, Feb. 2004. Available: http://securityresponse.symantec.com/avcenter/venc/data/w32.mydoom.f@mm.html.