用于 IP 追踪的包标记的注记*

李德全+, 苏璞睿, 冯登国

(中国科学院 软件研究所 信息安全国家重点实验室,北京 100080)

Notes on Packet Marking for IP Traceback

LI De-Quan+, SU Pu-Rui, FENG Deng-Guo

(State Key Laboratory of Information Security, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

+Corresponding author: Phn: +86-10-62528254-801, E-mail: dequanli@ieee.org, http://www.is.iscas.ac.cn

Received 2002-11-29; Accepted 2003-07-08

Li DQ, Su PR, Feng DG. Notes on packet marking for IP traceback. *Journal of Software*, 2004,15(2):250~258.
http://www.jos.org.cn/1000-9825/15/250.htm

Abstract: Distributed Denial of Service (DDoS) attack is among the hardest network security problems to address. Recently, several countermeasures are proposed, among which, PPM (probabilistic packet marking) pioneered by Savage *et al*. is promising. In this paper, a brief review of countermeasures to DDoS is given and then an analysis on some of the packet marking schemes is provided. Some modifications are further provided. One modification to the basic PPM scheme can reduce its computation remarkably.

Key words: IP traceback; denial of service; DoS; distributed denial of service; DDoS; packet marking

摘 要: 拒绝服务攻击是一类最难对付的网络安全问题.近来,人们提出了多种对策.其中由Savage等人提出的一类基于概率的包标记方案比较有研究价值.这里先对拒绝服务攻击的对策作一简述,然后分析了几种包标记方案,指出了它们的一些缺陷,并提出了一些改进措施.其中,对基本型概率包标记方案的一个修改使得计算量大大减少.

关键词: 网络追踪;拒绝服务;DoS;分布式拒绝服务;DDoS;包标记

中图法分类号: TP309 **文献标识码**: A

1 Introduction

Recently, Internet attacks are on the rise—more than 50% increase per year during 1998-2001[1]. Why are so many attacks occurring? Studies reveal that computer attacks have similarities with many other crimes: perpetrators are motivated by many things, including greed, revenge, and peer pressure. Denial of Service (DoS) attack

* Supported by the National Outstanding Young Scientists Foundation of China under Grant No.60025205 (国家杰出青年基金); the National Grand Fundamental Research 973 Program of China under Grant No.G1999035802 (国家重点基础研究发展规划项目(973))

LI De-Quan was born in 1969. He is a Ph.D. candidate at the Institute of Software, the Chinese Academy of Sciences. His research interests are network security. **SU Pu-Rui** was born in 1976. He is a Ph.D. candidate at the Institute of Software, the Chinese Academy of Sciences. His research interests are network security. **FENG Deng-Guo** was born in 1965. He is a professor and doctoral supervisor at the Institute of Software, the Chinese Academy of Sciences. His current research areas are information security.

consumes resources associated with various network elements—e.g., web servers, routers, firewalls and end hosts—which impedes the efficient functioning and provisioning of services in accordance with their intended purposes[2]. There are two types of DoS attacks: the first one takes advantage of drawbacks of some implementation or algorithmic deficiencies in applications by one or more malformed 'killer' packets; the second takes advantage of the sheer fact that the victim is connected to the Internet by flooding a deluge of packets to the victim. Since the first type is easy to be addressed by patching up vulnerabilities or filtering out malformed packets, we focus on the latter, called flood type DoS attack. From now on, for simplicity, we refer DoS attack to flood type DoS attack. DoS attack could be more effective if several attackers at different places conspire because the effect is summed up. This is generally called distributed DoS attack (DDoS attack).

DoS attacks are among the hardest security problems to address because they are easy to launch, difficult to defend and difficult to trace[3]. Firstly, DoS attack tools are available almost everywhere on the Internet; script kiddies could download these tools and launch attacks at will. Secondly, it is very difficult to differentiate poisonous (attack aimed) requests from the legitimate ones and this renders that it's difficult to defend from DoS attack. Thirdly, unlike other types of attacks such as privilege escalation attacks, DoS attacks do not need two-way communication. Therefore the source addresses of DoS attack packets could be spoofed. This feature leaves attackers opportunities to hide their true identities. In addition, some attacks (e.g. SYN flood attack) will be more effective with source addresses forged and source addresses must be forged for some other attacks (e.g. smurf, fraggle, and the like which adopt reflection) to be effective.

The attack against 13 root servers[4] in October 2002 and the one against '.info' domain system[5] in November 2002 exemplify that the DoS attack trends[6] have shifted from targeting company networks to targeting the infrastructure of the Internet itself.

The need to defend DoS attack and to find the attackers has grown in importance. Until we are able to dedicate attention to these attacks and follow these attacks to their end, we are all vulnerable.

The rest of the paper is organized as follows. An overview on countermeasures to DDoS attack is given in Section 2. Section 3 analyzes the Packet Marking scheme in Ref.[3] and gives some modifications. In Section 4, we give a brief analysis for Advanced PPM and Authenticated PPM. We conclude the paper in the last section.

2 Countermeasures to (D)DoS Attack

There are generally two categories of countermeasures to DoS attack: one is mitigating the detrimental impact of the attack on victim; another is tracing back to the offending parts. The former approach is passive and sometimes expensive (such as hot backups), while the latter is active in that it serves as a deterrent.

2.1 Mitigating

The aim of mitigating is to alleviate the effect of DoS attacks. Two kinds of measures could be taken for this purpose. The first is to enhance the tolerance of the victim; the second is to filter out the attack flow before it arrives the victim.

2.1.1 Tolerance enhancing

The aim of this is to enhance the ability of potential victims to tolerate attacks. This is the most widely used method for end users. Syn cookies, SYN cache, shortening the timeouts of TCP handshake and enlarging the TCP half open stack are of this type. So are resource provisioning (e.g., load balancing) and hot backup utilization.

2.1.2 Filtering

(1) Ingress filtering

DDoS attack often involves address spoofing. So, intuitively, if we could eliminate the ability to forge source

addresses, DDoS problem would be solved to a large extent. One such approach, called Ingress filtering[7], is to configure routers to block packets with obviously illegitimate source addresses. For example, if a perimeter router receives a packet from its interface directed to intranet, but the source address of that packet does not belong to the same network segment, then the packet should be blocked. To be effective in DoS attack defencing, this must be widely, if not universally, deployed. Unfortunately, a significant fraction of ISPs, perhaps the majority, do not implement this service. The reasons may be the administrative burden, potential router overhead or/and that there is no financial incentive.

(2) Route-Based filtering

Route-based distributed packet filtering (DPF)[2] uses routing information to determine whether a packet arriving at a router (e.g., border router at an Autonomous Segment) is valid with respect to its inscribed source/destination addresses, given the reachability constraints imposed by routing and network topology. A single AS(Autonomous Segment) can only exert a limited effect with respect to identifying and discarding a forged IP. At the other extreme, if all autonomous systems and their routers implement router-based packet filtering, then no spoofed IP can escape (except for packets with source addresses spoofed among that belonging to the same AS with the attacker), but its ultimate effect is not much different from that achievable by perfect ingress filtering. The main strength of this method lies in the fact that with partial coverage or deployment—about 18% in Internet AS topologies—a synergistic filtering effect can be achieved, which proactively prevents spoofed IP from reaching other autonomous systems in the first place.

2.2 Tracing

Studies suggest that many intruders are deterred by the perceived risks involved. One of the intruders' greatest fears is losing their anonymity. Consequently, if we could traceback to attackers, attacks would be reduced dramatically. Even if we have not traced back to the REAL attackers, having traced back to zombies or bots in DoS attack is meaningful because we could do some filtering or throttling closer to the packet sources and thus alleviate the attack force more efficiently. We could further track the attackers from zombies too.The aim of tracing is to traceback to packet sources. In recent years, researchers propose several tracing methods and all of them have their pros and cons respectively.

2.2.1 Packet marking

The main idea of packet marking[3,8,9] is to let routers mark packets with partial path information probabilistically. Having received enough packets from attackers or zombies, the victim could reconstruct the full paths along which attack flows travel. This field is pioneered by Savage and his co-authors in Ref.[3]. In this paper, we refer the algorithm in Ref.[3] as Basic PPM.

2.2.2 Logging

The notion of this kind is to log packet information to somewhere en route to the end. In Ref.[10], all routers store digests of packets they have delivered. If a victim decides to trace an attack packet, it queries some router about whether it has relayed that packet. The latter then checks the digest of the packet against its database, if there is a match with certain confidence, then the packet has traveled through that router. The prominent advantage of this method is that we can trace a single packet and keep the privacy about packet content at the same time. The problem is that queries must be done very soon after the attack, unless routers have some way of offloading historical data to bulk storage.

2.2.3 Link testing

Starting from the router near the victim, we need to check its upstream routers one by one and see which forwards the attack flow. This type of methods must be conducted while the attack is still in progress. Besides, if

there are several attackers with distributed location, link testing would be inefficient in that the attack flow from a single upstream router may contribute little to the attack so the attack would not be disturbed remarkably while the link is being tested.

(1) Back flooding

Burch *et al*.[11] introduces a flooding mechanic to check the attack path. In this method, the victim uses large bursts of traffic to flood some links and checks whether the incoming attack has been perturbed. If it has, the incoming attack path must share some link(s) with the egress flood. With the pre-generated topology information, the victim could get some links of the attack path. By doing so recursively, the victim could construct the whole attack path. This method is problematic in that it is a DoS attack by itself.

(2) Input debugging

Many routers such as CISCO[12] routers bear a feature called input debugging. While a victim detects a flood-type DoS attack, it informs its network operator the attack and attack signature. The network operator then adopts an access list to its upstream routers one by one according to the attack signature, and observes which originates the attack flow. Afterward the network operator does this recursively until the attack source or the administrative perimeter is reached. This is often performed manually and limited to administrative perimeter. To trace across networks, timely communication and cooperation between operators of different networks are needed, which is difficult to achieve generally.

2.2.4 ICMP Traceback (iTrace)

When a packet travels along a router, the router sends an authenticated ICMP Traceback[13] (a newly defined type of ICMP packet with type TRACEBACK) message probabilistically to the packet's destination and/or its source. This message contains the source and destination IP addresses of the Generator and its peer, the timestamp, the probability used to select packets for tracing, and some contents of the traced packet--at least the IP header and the first 64 bits of the body of the traced packet. With enough traceback messages from all routers along the path, the traffic source and the path of the attack can be determined.

The drawbacks of this method are twofold. One is that extra traffic is generated. Another is that because trace message is separated from the tracing one, the two may be differentially blocked by firewalls or policy routing mechanisms.

2.2.5 Centertrack

CenterTrack[14] is an overlay network, consisting of IP tunnels or other connections, which is used to selectively reroute interesting datagrams directly from edge routers to special tracking routers. The tracking routers or associated sniffers can easily determine the ingress edge router by observing which tunnel the datagrams come from. The datagrams can be examined, then dropped or forwarded to the appropriate egress point.

The virtue of this method is that it not only enables tracing back to attackers, but also mitigates the attack itself. On the other hand, this method is vulnerable to router exploiting (attacks against using routers and other infrastructures are increasing in recent years[6]). If attacker occupies not only the attacking hosts, but also the edge routers, then these routers will not reroute his attack flow to the tracking network. Furthermore, to be effective, the overlay network should be light weighted. If all SYN packets, UDP packets among some others are all rerouted to the tracking network, it must be overloaded because all this kind of packets could be used for attack.

3 Basic PPM

3.1 Overview of basic PPM*

Appending additional data to a packet in flight is a poor choice because it is expensive and may result in fragmentation. Savage *et al*. proposed a marking scheme in which each router marks its address information into packets transiting it with a probability p. In the case of flood-type DoS attack, the victim could get many marked packets. After collecting enough marked information, the victim could reconstruct the attack path through which attack flow traveled.

Identification field of IPv4 header is seldom used because very few packets are fragmented on the fly. Thus, this field could be used to embed some tracing messages. Let's denote two connected nodes on the network as an edge. In Ref.[3], the authors define edge-ID as XOR of the two IP addresses making up of an edge and "mark" this information into packet identification field probabilistically. After receiving enough packets from attacker(s), the victim could use these edges to reconstruct the full path. For this purpose, the victim needs not only edge-IDs but also corresponding distances. Since distance seldom exceeds 30[15,16], 5 bits are enough to code it. If we rely only on the identification field, we have 11 bits left and that is not enough for 32 bit edge-ID. So edge-ID should be further fragmented into several segments. To ensure reconstructing edge correctly when combining these segments, an error detection code is needed. Reference [3] uses 32bit hash of edge-ID as error detector. Therefore, 64 bits should be marked for one edge. The authors further interleave the error detection code with the IP address bit by bit, then separate the 64 bits into 8 blocks. So 5bit for distance, 3bit for offset (indicating the place of the block in 64bit full information), and 8 bit for block are inscribed to identification field, that is 16bit.

3.2 Analysis for basic PPM

The Basic PPM is a breakthrough work in the field of tracing back to attackers. Nevertheless, it is far from perfect. There are many shortcomings with it. Park *et al*.[9] analyze the uncertainty of the Basic PPM in the face of attacker spoofing his IP address. Reference [8] shows that the Basic PPM is time consuming and with high false positive via experiments. We here point out some other shortcomings of the Basic PPM and also give some recommendations for improvement.

3.2.1 High false positive rate & intensive computation

The Basic PPM is effective while there is only one attacker. If several attackers launch attacks against a victim simultaneously (in the case of DDoS attack), the Basic PPM is computationally expensive and with a large number of false positives even without spoofing of attackers**.

If there are k nodes in the attack tree at the same distance d (to the victim), we could get $k_0,k_1,\ldots,k_7$ different blocks with the same distance and with offsets 0,1,…,7 respectively(all k_i may be equal to or little smaller than k), and we have to check $\prod_{i=0}^{7}k_i$ combinations to get k edge-IDs, with each check involving a computationally intensive hash operation. Let $V(N,n)$ denote the number of distinct values we get while sampling n times repeatedly and equi-probably, with replacement from the population $D=\{1,2,...,N\}$, then k_i has almost the same distribution as $V(256,k)$ (we use "almost" since IP address is not random). The amount of computation is exponential to the number of the coordinated attackers. To reduce the error from combination, the error check code is 32 bits. The probability for a combination of 8 random blocks of 8bits passing the check is 2^{-32}, which is reasonably low, but with k

* Interested readers are referred to Ref.[3] for detail.

** Although this is pointed out in Ref.[8], we explain it in a different and clearer way.

different nodes at the same distance, the victim has to check $\prod_{i=0}^{7} k_i$ combinations, and the average number of false positives is $2^{-32}(\prod_{i=0}^{7} k_i - k) \approx 2^{-32}\prod_{i=0}^{7} k_i$. With k=17, there will be about 1.27 false positive in average; with k=20,30,40,50,…, there will be 4.4, 97.5, 835.2, 4276.8… false positives in average!

3.2.2 Poor scalability

In the Basic PPM, the authors divide the edge information into too small fragments. Each fragment has only 8 bits, which comes to 256 different possibilities. Thus, if the number of distributed attackers approaches 256 and there are 200 coordinated attackers, it is nearly impossible for the Basic PPM to differentiate candidate edges from normal ones.

3.2.3 Unfair probability and the weakest link

For any router, the probability p of its marking process is fixed and uniform. After a router marks a packet, the packet may be remarked by downstream routers. If a router is d hops away from the victim, the probability that its marked information survives through the whole journey to the victim is $p(1-p)^d$. The larger the distance d is, the less the probability is. The link which is the nearest to the attacker is the 'weakest' one*. This means that more packets are needed to reconstruct the full path. One solution is to use an adaptive marking policy that adopts different marking probability according to the distance field. We have developed such a scheme in Ref.[17].

3.2.4 Forgeable shorter path**

It is widely accepted that attackers could spoof extra edges past the end of the true attack path. In fact, in the Basic PPM, according to the marking algorithm (see Fig.8 in Ref.[3]), an attacker could also spoof edges nearer to the victim than the attacker.

Let's see how. Suppose that the distance between the attacker and the victim is d, i.e. the attacker is d hops away from the victim. Suppose that router v_k and v_{k+1} are k and k+1 hops away from the victim respectively ($k<d$), and both are on some paths to the victim; and that router u_k and u_{k+1} are k and k+1 hops away from the victim, and both are on the attack path (Fig.1). We also suppose that the attacker knows all about this. Before the attacker sends a packet to the victim, he marks the packet with

$$R=BitInterleave(v_k \oplus v_{k+1} \oplus u_k, Hash(v_k \oplus v_{k+1} \oplus u_k)).$$

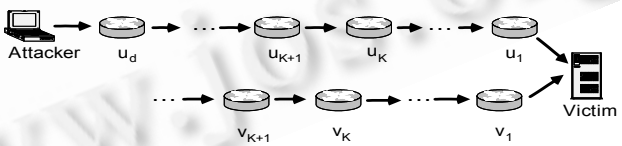

Fig.1 Attack path and forgeable path

The attacker also sets the distance to (31–d+k+1). If all the intermediate routers choose not to mark the packet (the prbability is $(1-p)^d$), then the distance will be 31 while the packet arrives at router u_{k+1}. Router u_{k+1} increments it and gets zero before relaying the packet. While the packet arrives at router u_k, it will find that the distance is zero, so it changes the marking field and then the packet looks as follows:

$$R'=BitInterleave(v_k \oplus v_{k+1} \oplus u_k \oplus u_k, Hash(v_k \oplus v_{k+1} \oplus u_k \oplus u_k))$$

* This is also pointed out by Park and Lee in Ref.[9]

** Although Savage *et al*. used "saturating addition", they did not explain it in the algorithm (Fig.8 in Ref.[3]) and did not explain how attackers spoof and how to deal with a packet for router while the distance reaches 31.

$$=BitInterleave(v_k \oplus v_{k+1}, Hash(v_k \oplus v_{k+1}))$$

except that the distance is set to 1. After the packet arrives at the victim, the victim will get an edge v_k–v_{k+1}.

The key to this is simple—add a limit checking to the distance. If a router receives a packet, it checks whether the distance is 31. If so, it marks the packet and forward it, or more drastically, drops the packet. If the distance is not 31, it marks the packet probabilistically. Few pathes exceed length 32[15,16], but if there are any, will the policy that drops packets with distance 31 serve as another DoS attack to applications with their traffic paths longer than 32? No, it won't! With marking probability p=0.04, the probability that a packet is not marked after it passes 32 routers (and then it is dropped) is $(1\text{-}0.04)^{32}\approx 0.27$, which is low. Even if the packet is dropped, the source node could retransmit it for a reliable connection. For a non-reliable connection, it's not a big problem with some packets lost.

3.2.5 Interleaving

Error checking code is interleaved into IP address bit-by-bit to get a whole 64 bit. The victim has to de-interleave after he gets all the 8 blocks and then concatenates them bit-by-bit into two 32 bit blocks. From our analysis, interleaving has no other function except combination in this circumstance*, but it complicates the operation! Using concatenation could be simpler and better. By using concatenation instead of interleaving, we could combine the fragments into $\prod_{i=0}^{7} k_i$ combinations, and then check each of them. This works similar with that while interleaving is used. We also could combine elements of the first 4 sets of fragments (with offsets 0,1,2,3) and get $\prod_{i=0}^{3} k_i$ combinations, then check them against the last 4 sets of fragments. That is: hash a combination, separate the result into 4 blocks, 8 bits each, then check whether the first block belongs to set 4 (compiled with fragments with offset 4), the second block belongs to set 5, and so on. If all answers are 'yes', we get an edge; if any of them is 'no', skip it. In this way, we check only $\prod_{i=0}^{3} k_i$ combinations, and this reduces the computation into $(\prod_{i=4}^{7} k_i)^{-1}$ that of interleaving. If k_i are about 20, this change will reduce computation to 20^{-4} of that of interleaving at this distance. Suppose there are 20 coordinated attackers with their locations widely distributed. Since the path length is about 15 in average on the Internet[15], we suppose that all attack paths are 15. We further suppose that the number of nodes increases proportionally at different distances (This is a very simplified presumption. Our purpose is to get a clue other than the exact amount of computation), that is: from 1 node(the victim itself) at distance 0 to 20 nodes at distance 15, so at distance d ($0<d<15$), there are about $(1+x)^d$ nodes (with $(1+x)^{15}=20$, that is to say, x=0.2210553). If there are k nodes at distance d_1, we may get k_i ($k_i<=k$) different fragments with offset i. Since there are false positives, we suppose there are $k_i=k$ different fragments with offset i. So the amount of computation (in the number of hash operations) while interleaving is used is

$$\sum_{d=1}^{15}((1+x)^d)^8=\sum_{d=1}^{15}(1+x)^{8d}=3.2\times 10^{10},$$

and while concatenation is used, it is

$$\sum_{d=1}^{15}((1+x)^d)^4=\sum_{d=1}^{15}(1+x)^{4d}=2.9\times 10^{5}.$$

The ratio of the two is 1.1×10^5. Reference [8] simulated Basic PPM on a 500MHz Pentium III Linux workstation and experienced more than one day with 20 attackers. With our modification applied, a couple of seconds will be

* We attempted to discuss this with the authors of Ref.[3], but got no response after sending several emails to their several addresses.

enough in the same situation.

For a combination to pass the error checking, its hash must be one of the $\prod_{i=4}^{7} k_i$ combinations and the probability is about $2^{-32}\prod_{i=4}^{7} k_i$. So the number of false positives is about

$$\prod_{i=0}^{3} k_i \cdot 2^{-32} \prod_{i=4}^{7} k_i = 2^{-32} \prod_{i=0}^{7} k_i$$

in average, which is the same as that with interleaving.

4 Advanced and Authenticated Packet Marking

Knowing that the Basic PPM is computationally intensive and with a high false positive rate, Song *et al.*[8] proposed two algorithms called advanced packet marking and authenticated packet marking scheme respectively. Both of them make use of identification field for marking.

In the advanced marking scheme, a router marks the hash of its IP address into a packet instead of the IP address itself. This scheme is both computationally efficient and scalable to highly distributed DoS attack (in the case of 1500 coordinated attackers according to the authors). Because the marked information is the hash of IP address and the hash is a one way function, this necessitates that the victim knows its upstream topology. Since an attack could come from anywhere on the Internet, this means that the victim should know virtually the topology of the whole Internet, which is difficult for all potential victims themselves since the Internet itself is changing all the time. If all potential victims or their ISPs try to keep the updated topology information by themselves, they have to probe the Internet frequently, thus a large volume of extra traffic would be caused. This problem could be solved with Topology Information Server[18] which is in the hands of some Internet Authority and stores Internet topology information. Each time the topology changs, it must be reported to the Authority to keep the information updated. A victim could fetch topology information from the server while tracing back to attacks. Several such servers may be scattered on the Internet for convenience.

The authenticated packet marking scheme is similar to the advanced one, except that while a router marks a packet, it must authenticate the marking field (i.e. the identification field of packet header). The keys used for authentication form a key chain and are time-released. The time is divided into interval. Each router then associates its key sequence with the sequence of the time intervals, one key per time interval. To counter spoofing, the key disclosure time delay must be greater than any reasonable round trip time on the Internet plus the dispersion. This eliminates attackers' ability to forge the marking field. In practice, there are some disadvantages to this scheme. For one thing, the disclosure time delay must be long enough for the difficulty of universal time synchronization; for another, this delay also delays the tracing process for which timeliness is more appreciated sometimes for countermeasures such as filtering or throttling the attack flow. Furthermore, the universal key database may be clogged because of key releasing and key fetching since every router on the Internet must disclose its newest key once in every time interval.

5 Conclusions

We give a brief overview about research in countering flood-type DoS attack. We further analyze the shortcomings of the two typical packet marking schemes and propose some modifications. One of our modifications to the Basic Probabilistic Packet Marking scheme can reduces its computation to about one 10^5th in case of 20 distributed attackers.

References:

[1] CERT. CERT Statistics. http://www.cert.org/stats/#incidents

[2] Park K, Lee H. A proactive approach to distributed DoS attack prevention using route-based packet filtering. Technical Report, CSD00-017, Department of Computer Sciences, Purdue University, 2000. http://www.cs.purdue.edu/nsl/dpf-tech.ps.gz

[3] Savage S, Wetherall D, Karlin A, Anderson T. Practical network support for IP traceback. In: Proc. of the 2000 ACM SIGCOMM Conf. Stockholm, 2000. 295-306. http://www.acm.org/sigs/sigcomm/sigcomm2000/conf/paper/sigcomm2000-8-4.ps.gz

[4] McGuire D, Krebs B. Attack on Internet called largest ever. 2002. http://www.washingtonpost.com/ac2/wp-dyn/A828- 2002Oct22?

[5] Lemos R. Attack targets info domain system. ZDNet News, 2002. http://zdnet.com.com/2100-1105-971178.html

[6] CERT. Overview of attack trends, 2002. http://www.cert.org/archive/pdf/attack_trends.pdf

[7] Ferguson P, Senie D. rfc2827, Network ingress filtering: defeating denial of service attacks which employ IP source address spoofing. IETF, May 2000. http://www.ietf.org/rfc/rfc2827.txt

[8] Song DX, Perrig A. Advanced and authenticated marking schemes for IP traceback. In: Proc. of the IEEE INFOCOM 2001. http://www.ieee-infocom.org/2001/program.html

[9] Park K, Lee H. On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack. In: Proc. of the IEEE INFOCOM 2001. 2001. 338~347. http://www.ieee-infocom.org/2001/program.html

[10] Snoeren AC, Partridge C, Sanchez LA, Jones CE, Tchakountio F, Kent ST, Strayer T. Hash-Based IP traceback. In: Proc. of the ACM SIGCOMM 2001 Conf. 2001. San Diego, 2001. 3~14. http://www.acm.org/sigs/sigcomm/sigcomm2001/p1.html

[11] Burch H, Cheswic B. Tracing anonymous packets to their approximate source. In: Usenix LISA 2000. New Orleans, 2000. 313~321. http://www.usenix.org/publications/library/proceedings/lisa2000/burch.html

[12] CISCO. Characterizing and tracing packet floods using Cisco Routers. http://www.cisco.com/warp/public/707/22.html

[13] Bellovin S, Leech M, Taylor T. ICMP traceback messages. Internet Draft, draft-ietf-itrace-04.txt, February 2003. http://www.ietf.org/internet-drafts/draft-ietf-itrace-04.txt

[14] Stone R. Centertrack: An IP overlay network for tracking DoS floods. In: Proc. of the 9th USENIX Security Symp. 2000. http://www.usenix.org/publications/library/proceedings/sec2000/stone.html

[15] Cooperative Association for Internet Data Analysis (CAIDA). The Skitter project. http://www.caida.org/tools/measurement/skitter/

[16] Theilmann W, Rothermel K. Dynamic distance maps of the Internet. In: Proc. of the 2000 IEEE INFOCOM Conf. Tel Aviv, 2000. http://www.ieee-infocom.org/2000/program.html

[17] Li DQ, Xu YD, Su PR, Feng DG. Adaptive packet marking for IP traceback. Acta Electronica Sinica, 2003 (in Chinese with English abstract), accepted.

[18] Li DQ, Su PR, Feng DG. Router numbering based packet marking. In: 2003 Int'l Workshop on Cryptology and Network Security (CANS03). Miami, 2003.

附中文参考文献:

[17] 李德全,徐一丁,苏璞睿,冯登国.IP 追踪中的自适应包标记.电子学报,2003,已录用.