Performance of IP Address Fragmentation Strategies for DDoS Traceback

Ihab Hamadeh
Department of Computer Science and Engineering
Pennsylvania State University
University Park, PA 16802
Email: hamadeh@cse.psu.edu

George Kesidis
Department of Computer Science and Engineering
Department of Electrical Engineering
Pennsylvania State University
University Park, PA 16802
Email: kesidis@engr.psu.edu

***Abstract*— Distributed Denial-of-service (DDoS) attacks are among the most difficult and damaging security problems that the Internet currently faces. The component problems for an end-system that is the victim of a DDoS attack are: determining which incoming packets are part of the attack (intrusion detection), tracing back to find the origins of the attack (i.e., “traceback”) and, finally, taking action to mitigate or stop the attack at the source by configuring firewalls or taking some other kind of punitive measures. The preferable solution to these problems will operate in real time so that a DDoS attack can be mitigated before the victim is seriously harmed. This paper focuses on the technique of packet marking/overloading for automated DDoS traceback which is a complex problem simply because attackers can use spoofed source IP addresses in their attacking packets. A new packet marking strategy is proposed and is shown to yield better results in terms of complexity and performance.**

I. Introduction

Distributed-Denial-of-service (DDoS) attacks are growing threats to today’s Internet. With the availability of automatic attacking tools such as Tribe Flood Network (TFN), TFK2K, Triboo and Stacheldraht, any person with substantial knowledge about networking can easily carry out a DDoS attack. Some statistics [11] show that DoS and DDoS attacks are so prevalent today that they present a great threat to e-business. A recent noticeable attack [12] was launched in October 2002 against 13 root servers which are used by the Internet’s Domain Name Servers (DNS). In the past, targets of attacks have included the most recognizable corporations, the White House and CERT itself [9]. A single DDoS attack in 2000 is believed to have cost hundreds of million of dollars [21]. Because of the damage that such attacks incur on the Internet and on the business of some online companies that profit directly or indirectly from their devoted subscribers or users (Amazon.com, Buy.com, eBay, etc.), there is an immediate need for a real-time mechanism for tracking down the sources of these attacks as part of an effort to deter future ones.

To mitigate or terminate a DDoS attack, a victim end-system must address the following component problems: determining which incoming packets are part of the attack (intrusion detection), tracing back to find the origins of the attack (traceback) and, finally, taking action to mitigate or stop the attack *at the identified source* by configuring firewalls or taking some other kind of punitive measures. Determining the sources of an attack is not, however, a simple task since attackers typically use incorrect or spoofed IP addresses. IP address spoofing can create the appearance that the attacks are being carried out by innocent end-systems. For these reasons, several solutions have recently been proposed to automatically traceback the sources of DDoS attacks and mitigate them.One family of promising techniques advocates overloading/marking of rarely used fields in IPv4 headers by trust-worthy routers with a portion of their own IP address. Example methods include probabilistic packet marking (PPM) [16], [19] and border router packet marking (BRPM) [7], [8]. A brief overview of existing techniques is given in the Section II. In Section III, we give an overview of address fragmentation techniques for the purposes of packet marking for automated traceback. A new hybrid method is then proposed and its low implementation complexity and low rate of false positives are demonstrated.

II. Related Work

This section briefly introduces several previously proposed techniques to trace back the origins of a DDoS attack [8], [4], [3]. General criteria for evaluation of traceback techniques include: false positive rates (including those maliciously caused), missed detection rates, computation and communication overhead, deployment complexity, and DoS effects of the firewalls configured as a result of traceback.

Under link testing (Input Debugging) which is based on having a victim reports an attack to its upstream router, which in response installs a debugging filter that reveals which upstream router originated the attacking traffic. While such tracing may be done manually, many ISPs have developed tools to automate tracing back of attacks across their own networks such as that proposed in [20]. Under a traceback method called link testing (controlled flooding) [2], the victim forces selected hosts to flood each incoming link of the router closest to the victim and monitors the change in the attack packet rate and determines from which link the attack is arriving. Another approach to trace back proposes to log packets at various points throughout the network and then use some extraction ("data mining") techniques to find the path packets traversed, see [15]. Snoeren et al [17], [18] proposed a modification to this approach called the Source Path Isolation Engine (SPIE). Ferguson and Senie [6] suggested the use of ingress filtering. Bellovin [1] proposed a scheme known as the ICMP Traceback Messages that was later extended by Wu et al [22]. In this scheme, routers, with low probability, generate a Traceback message that is carried in an ICMP packet and is sent along the path of the packet.

A. Probabilistic Packet Marking

The PPM scheme [16], [19], [5] requires that a router (including an interior router), with specified probability, inscribes its local path information into the packet header. The *path* of the packets is reconstructed starting from the packets received from the closest routers moving up to the ISPs' border routers.

FMS (and perhaps AMS) may not be able to traceback significant DDoS attacks in real time, i.e., while an attack is on-going. AMS requires several hundred to thousands of packets (depending on the path length and the marking probability) per attacker to be able to reconstruct the attack path (see Fig. 12 and 13 of [19]).

Also, since routers mark in a probabilistic way, the victim will receive many unmarked packets that are part of the attack. An attacker can easily take advantage of this flaw by inserting fake links distances into the overloaded packet header fields [13] to cause *malicious* false positives. To overcome this problem, Song and Perrig [19] proposed authentication of the packet marking using Message Authentication Codes (MAC).

B. Border Router Packet Marking (BRPM)

Under BRPM, only "border" routers mark packets and all packets inbound into the Internet are marked. More precisely, the marks are based on the IP addresses of the border router's input-link interface that the packet arrived on. Marking every packet in this way protects against an attacking end-system inserting fake marks into this field in an attempt to compromise traceback with malicious false positives. Traceback is made unambiguous by the deployment assumption that each packet is forwarded into the Internet by only one trustworthy marking border router. In practice, such a "border router" could be the first point-of-presence (PoP) of a trustworthy ISP, a gateway router to an unreliable autonomous system on the periphery of the Internet, or even a *secure* end-host (in which the marking process cannot be circumvented).

BRPM has the following advantages over PPM

- BRPM requires fewer marking routers. This alone implies easier deployment, fewer false positives and simplified address reconstruction.
- BRPM places firewalls as far as possible from servers, i.e., in the border routers [10].
- BRPM marks *all* packets so that *malicious* false positives are prevented.

Unlike all PPM schemes, the BRPM scheme does **not** require any router to decide whether to mark an IP datagram. Note that marking *every* packet is no more complex, from either a hardware or software perspective, than marking packets at random (where *potentially* all packets are marked).

Finally, PPM may be more secure than BRPM when hijacking of border routers is a significant threat because interior routers that are not hijacked will still be able to mark packets and, thereby, overwrite fake marks inserted by a hijacked router. However, it is not clear that this is, in fact, a significant threat and it is not clear that the marking function (a simple operation deployed in the microcode of the routers' ingress network processors) can be modified or stopped by a hijacker. Finally, this threat is clearly significantly less than that of the malicious false positive problem that PPM suffers (when attacking end-users insert fake marks).

C. Defining a border router

A "border" router can be any of the following: border/leaf routers, area border routers (ABR) and/or on gateway/boundary routers depending on the part of the Internet that needs to be secured. From the perspective of a given server that wishes to perform traceback on marked packets it determines are participating in an attack: the server could identify *all* the routers at the perimeter of its *trust region* and request that those routers mark all packets that are *destined* for the server under consideration. In a router, the marking function could be tied to the packet forwarding mechanism of the network processors resident on its input linecards. Therefore, from the perspective of any given server in this more general setting, BRPM could be deployed so that all of the routers at the perimeter of its trust region mark packets that are forwarded to it. Traceback under BRPM would, however, be most valuable if all servers had a common "maximal" trust region thereby placing firewalls as close to the true source

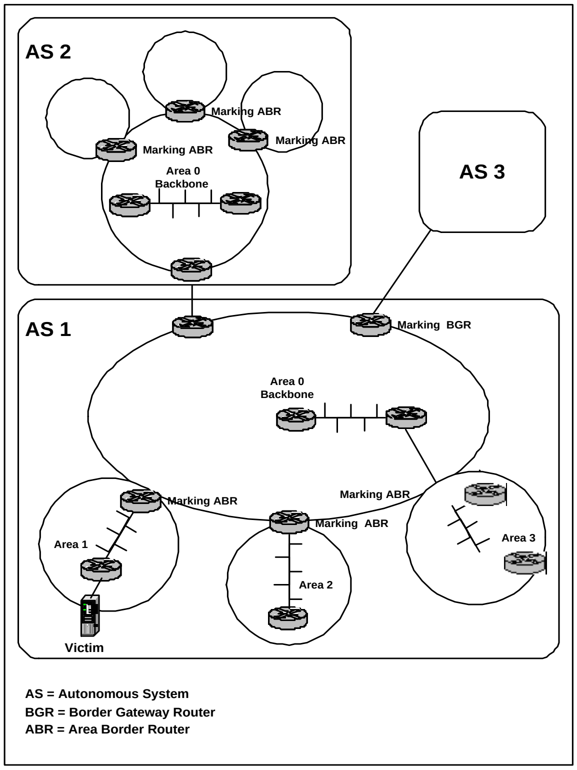

Fig. 1. BRPM deployment

of an attack as possible (resulting in the smallest DoS effect to innocent end-systems).

Figure 1 is an example indicating onto which border routers BRPM can be deployed. Note that marking ABRs mark datagrams that originate from their associated subnetworks. Referring to Figure 1, if a DDoS attack originated from any area of autonomous system $AS2$ against end systems in Area 1 of autonomous system $AS1$, the victims would be able to reconstruct the IP addresses of the area border routers from which the attack emanated. Now assume autonomous system $AS3$ does not employ BRPM due to the lack of sufficient levels of technical or political cooperation between Internet Service Providers (ISPs). If a DDoS attack originates from within autonomous system $AS3$, the marking border gateway router (BGR) marks all packets originating from $AS3$. Hence any end-systems under attack in $AS1$ and $AS2$ are able to determine that the source of the attack is coming from $AS3$.

III. Existing address fragmentation and reconstruction strategies

In this section, we briefly describe previously proposed techniques for packet marking. These methods simultaneously attempt to reduce false positives and/or address reconstruction complexity. The problem here is how to segment a 32-bit source IP address into smaller fragments suitable for overloading the IPv4's 13-bit Fragmentation Offset field plus an unused flag bit and the unused 2-bit TOS field, a total of 16 bits. The variable k will represent the total number of fragments or the number of fragments belonging to a single identified group. Also, n will represent the number of different border router interfaces through which attacking packets enter the Internet, i.e., the number of different "attacking" border routers.

A. *Packet marking strategies associated with PPM*

Two prominent varieties of packet marking have been proposed in association with PPM: Scheme (FMS) by Savage et al [16] and Advanced Marking Scheme (AMS) by Song and Perrig [19]. Under FMS, each router's IP address is bit interleaved with a "uniform hashed" version of the same address. The resulting 64-bit quantity is partitioned into k (nonoverlapping) fragments. In the FMS packet marking approach [16], fragments are collected by a victim end-system under a DDoS attack and their contents are de-interleaved to obtain an IP address and hashed-value fragments. Complete 32-bit addresses and their hash values are reconstructed by simply *concatenating* the fragments. Finally, the hash function is applied to each reconstructed address to see if the result agrees with the corresponding reconstructed hash value. This last step has the effect of reducing false positives.

Under AMS, each router's IP address is hashed into an 11-bit or 8-bit value (according to whether AMS version I or II is used) and probabilistically inscribed in forwarded IP packets. However, unlike FMS, AMS requires the knowledge of a topological map of the Internet a priori to be able to reconstruct a 32-bit router IP address from the 11-bit or 8-bit hash values.

In the case of FMS, the address reconstruction of the routers' IP address is of the order $\Theta(n^k)$. The number of false positives cannot be predicted but the number of reconstructed routers' IP addresses is $\Theta(n^k)$. False positives are eliminated by recovering the IP address and its scrambled version and find a match. Note that in FMS, by using a 32-bit "hashed" IP address, k is 8.

In the case of AMS, the address reconstruction of the routers' IP address is clearly of the order $\Theta(n)$. The number of false positives cannot be predicted but the number of reconstructed routers' IP addresses is $O(2^{32-f}n)$, where f is fragment size. False positives are eliminated by checking the reconstructed IP addresses with topological map of the Internet.

B. *Groups of overlapping fragments*

We previously associated a packet marking framework with BRPM in which a border router's IP address is fragmented (segmented) into several (k) overlapping fragments where each fragment has an identifying index (IDs

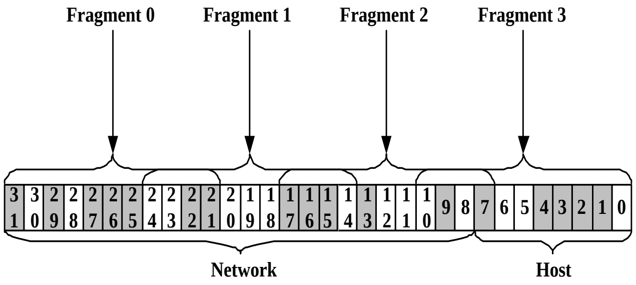

Fig. 2. Overlapping 11-bit fragments (group #1) spanning a 32-bit border router address

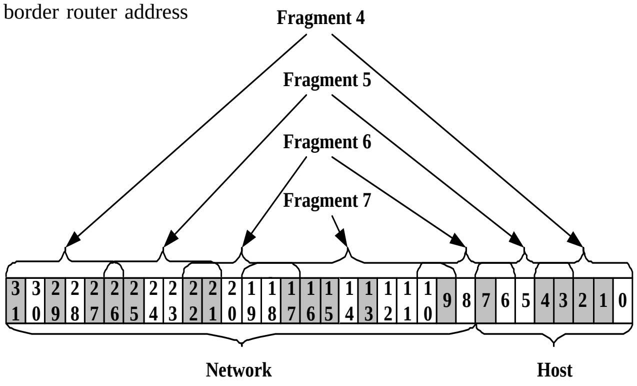

Fig. 3. Fragment group #2 overlapping in both the network and host class-C address fields

0 to $k - 1$). Border routers write into the header (of every packet they forward) a selected fragment and its identifier (ID). The number of bits needed for storing both a fragment and its ID is at most $n + \lceil \log k \rceil$ where n is the fragment size and k is the total number of fragments. Example fragmentation strategies are given in Figures 2 and 3.

The *quantity* of addresses reconstructed is reduced when overlapping fragments as the packet marks. The net effect of this fragment "redundancy" is to reduce false positives *during* the address reconstruction itself and, therefore, this process is less complex. More specifically, suppose an end-system is under a DDoS attack. Address reconstruction works as follows. The address fragments and their identifiers are extracted from the packet headers. Only pairs of fragments with *identical* (matching) overlapping fields are merged together to form a larger address "metafragment". Metafragments are then made even larger, according to this same rule, by continuing to merge them with other fragments whose overlapping fields agree with those of the metafragment.

We now specifically describe how false positives arise in the overlapping-fragments framework. Consider the simple example of two fragments ($k = 2$) of $n = 20$ bits that therefore overlap in 8 bit positions (Figure 4). For a given 32-bit border router address A, let

- $w(A)$ be this 8-vector of the overlapping bits
- $f_i(A)$ be the fragment with ID i for $i = 0, 1$
- $b_i(A)$ be the 12-vector of *non*-overlapping bits of $f_i(A)$

We therefore write $f_i(A) = b_i(A) \oplus w(A)$, i.e., the i^{th} fragment is composed of non-overlapping (unique to f_i) bits b_i and the overlapping bits w. Now consider two logged fragments with different IDs, $f_0(A_0)$ and $f_1(A_1)$, where A_0 and A_1 are the actual IP addresses of border routers that marked the corresponding packets (of course, A_0 and A_1 are not known a priori to the entity performing traceback). If the overlapping bits agree, i.e.,

$$w(A_0) \quad = \quad w(A_1) =: W, \tag{1}$$

then the following 32-bit IP address will be reconstructed given fragment instances $f_0(A_0)$ and $f_1(A_1)$:

$$\begin{aligned} D_0 &\equiv b_0(A_0) \oplus W \oplus b_1(A_1) \\ &= b_0(A_0) \oplus f_1(A_1) = f_0(A_0) \oplus b_1(A_1) \end{aligned}$$

Note that if $A_1 = A_0$ (i.e., the two fragments under consideration are taken from the same address) then $D_0 = A_0$ (i.e., the address is successfully reconstructed).

Consider two "attacking" border router IP addresses $A_0 \neq A_1$. Referring to Figure 4, false positives are generated **only** when

$$\begin{aligned} w(A_0) = w(A_1) =: W \quad &\text{and} \quad b_0(A_0) \neq b_0(A_1) \\ &\text{and} \quad b_1(A_0) \neq b_1(A_1). \end{aligned}$$

In this case, the two false positives generated are:

$$\begin{aligned} F_0 &\equiv b_0(A_0) \oplus W \oplus b_1(A_1) \\ \text{and} \quad F_1 &\equiv b_0(A_1) \oplus W \oplus b_1(A_0). \end{aligned}$$

In the following we will assume a victim server receives all fragments from all attacking routers; thus, there will be no missed detections. Assume that the approach of a single group of overlapping-fragments is used and let:

- Φ be the number of fragments
- Ω_i be the set of fragments that overlap with fragment i
- $\omega_{i,j}$ be the number of overlapping bits between fragments i and j, if $j \in \Omega_i$ (note: $\omega_{i,j} = \omega_{j,i}$)
- V be the total number of bits that are used by just a single fragment (i.e., total "nonoverlapping" bits)
- N be the number of "attacking addresses"

If we further assume that

- each fragment overlaps with exactly two other fragments
- no single bit is shared by more than two fragments
- $N \leq \min_{i,j} 2^{w_{i,j}}$

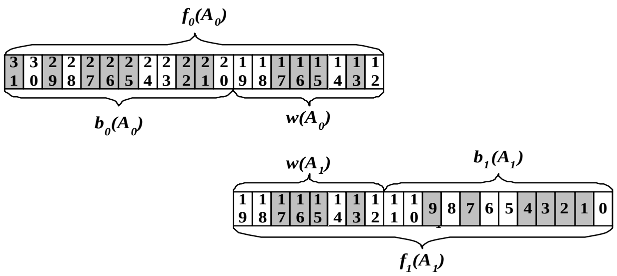

Fig. 4. Example of two fragments ($k = 2$) of $n = 20$ bits that overlap in 8 bit positions.

- The set of N attacking addresses are independently selected and follow a uniform distribution over the entire set of 32-bit IP addresses

then a simple formula for the probability of zero false positives is:

$$2^{VN}\left(\prod_{i=1}^{\Phi}\prod_{j\in\Omega_i,j>i}\frac{2^{\omega_{i,j}}!}{(2^{\omega_{i,j}}-N)!}\right)/\left(\frac{2^{32N}!}{(2^{32N}-N)!}\right)$$

where we note that

$$32 \quad = \quad V+\sum_{i=1}^{\Phi}\sum_{j\in\Omega_i,j>i}\omega_{i,j}$$

since the fragments must span the 32 different bits of an IP address. In particular, if $|\Omega_i| = 2$ and a constant $\omega = \omega_{i,j}$ for all i and all $j \in \Omega_i$, then the probability of zero false positives is:

$$2^{32V}\left(\frac{2^{\omega}!}{(2^{\omega}-N)!}\right)^{\Phi}/\left(\frac{2^{32N}!}{(2^{32N}-N)!}\right)$$

where $V + \omega\Phi = 32$.

The numbers of false positives using this overlapping technique *alone* are too high to be of practical value and requires too much address reconstruction complexity. In [7], [8], we used the technique of fragment *grouping* to further reduce address reconstruction complexity (fewer reconstructed addresses) and false positives. Consider the examples of Figures 2 and 3 where the four overlapping fragments numbered 0-3 and those numbered 4-7 are respectively *grouped* together. The fragments in each group span the entire 32-bit address. In general, the previously described address reconstruction process will lead to a set S_i of reconstructed 32-bit addresses using only those overlapping fragments whose IDs belong to the i^{th} group. Using multiple fragment groups, only those reconstructed IP addresses that are *common to all sets*, i.e., those in

$$S \equiv S_1 \cap S_2 \cap S_3 \cap \cdots \cap S_n \qquad (2)$$

are deemed to be border routers through which originated transmission to the end-system performing traceback.

In [7], [8], we generated "attacking" border router addresses in two ways:

- mutually independent and uniformly at random from within classes A, B, and C (i.e., between 0.1.0.0 and 223.255.255.0),
- weakly-correlated by fixing the leftmost 8 of the network address bits (i.e., assuming all attacks are emanating from border routers whose IP addresses fall within 200.0.0.0 and 200.255.255.255). Note that we assumed that the last 8 bits represent the host portion while the remaining 24 bits are the network portion.

After running the simulation using the approach of multiple groups of overlapping fragments, all attacking border router IP addresses were successfully reconstructed[1] along with small numbers of false positives: 2-4% for $\leq$ 400 attacking border routers and 65-70% for $\leq$ 700 attacking border routers.

IV. A NOVEL HYBRID PACKET MARKING APPROACH

We propose a hybrid marking strategy consisting of a *single* group of, say, four *overlapping* fragments spanning a 32-bit border router address.Now consider a "hash" function h that maps a 32-bit address A to an H-bit quantity $h(A)$ where $H \geq 32$ (IP addresses may be initially padded with a known fixed suffix prior to application of the hash function). This H-bit quantity is then fragmented into $k-4$ fragments. To minimize false positives, the chosen hash function h should create very dissimilar (uncorrelated) values $h(A)$ for addresses that are similar (correlated) in the domain of addresses under consideration.

Address reconstruction works as described above for the group of four overlapping fragments taken from the unmodified IP address. In addition, the hash fragments are stored. *Once* an address A is reconstructed, the hash function h is applied to it and the stored received fragments are searched to see if the fragments of $h(A)$ have all been received. If *all* $k-4$ fragments of $h(A)$ have been received, A is deemed to be an address of an "attacking" border router.

Note that, given the fragmentation strategy described above, there is a choice as to whether to attempt to reconstruct the hash quantities "$h(A)$" from their associated received fragments. Such a reconstruction process, however, will increase the complexity of address reconstruction.

Also, simply checking whether the fragments of $h(A)$ have been received (A being a reconstructed address)

[1] 0% missed detections follow from the basic assumption that all fragments are correctly received by the victim end-system from each attacking address.

could be done very efficiently in hardware using a set of content addressable memories (CAMs), one for each of the $k-4$ fragment indexes associated with the hash value.

While using non-overlapping fragments on $h(A)$ may allow more of the hash value to be represented in the case where the fragment size is smaller than $H/(k-4)$, overlapping fragments on $h(A)$ may allow more correlation between fragments. In the different simulations we conducted, nonoverlapping fragments of the H-bit quantity yielded significantly more (sometimes by as much as 50%) false positives than overlapping fragments in this case.

V. Performance study

To simplify the comparison between different address fragmentation strategies, evaluation was done in the context of the Border Router Packet Marking (BRPM) scheme advocated in [7], [8]. We used the MD5 message digest algorithm[14] as our hash function h. MD5 is a secure hash function used to verify data integrity through the creation of a 128-bit message digest from data input.

In addition to the uncorrelated and weakly correlated cases described at the end of Section III-B, we generated "attacking" border router addresses that were:

- strongly-correlated by fixing the leftmost 13 of the network address bits (i.e., assuming all attacks are emanating from border routers whose IP addresses fall within 200.0.0.0 and 200.7.255.255). Note that we assumed that the last 8 bits represent the host portion while the remaining 24 bits are the network portion.

Let f be the fragment size, k be the total number of fragments, and n be the number of "attacking" border routers. To test our proposed approach, we used the fragmentation framework of Figure 2, varied the fragment size and the number of different fragments, assumed that all fragments from each attacking address were received by the victim end-system, and assumed the victim end-system employed a perfect intrusion detection, i.e., identified all attacking packets with no false positives. We report at most $1.0\% \pm 0.2$ false positives (in the *addresses* reconstructed and deemed to be attacking) with 95% confidence for the following values of n.

Case 1: $f = 13$ and $k = 8$ (16-bit mark)

- For mutually independent IP addresses between 0.1.0.0 and 223.255.255.0, $n \leq 600$.
- For weakly-correlated independent IP addresses between 200.0.0.0 and 200.255.255.255, $n \leq 900$.
- For strongly-correlated independent IP addresses between 200.0.0.0 and 200.7.255.255, $n \leq 1500$.

Case 2: $f = 12$ and $k = 16$ (16-bit mark)

- For mutually independent IP addresses between 0.1.0.0 and 223.255.255.0, $n \leq 1500$.
- For weakly-correlated independent IP addresses between 200.0.0.0 and 200.255.255.255, $n \leq 2000$.
- For strongly-correlated independent IP addresses between 200.0.0.0 and 200.7.255.255, we report 0% false positives observed in 15 separate trials for $n \leq 2000$.

Note that within each case, the more the border IP addresses are "correlated"[2], the more the number of "attacking" border routers that can be resolved with the same false positive rate of 1%. This can be explained by taking note of the total number N of IP addresses *reconstructed*, i.e., from the group of four *overlapping* fragments of the unmodified IP address. For instance, in case 2 and for n = 1000:

- For mutually independent IP addresses between 0.1.0.0 and 223.255.255.0, $N \approx 9.5$ million IP addresses.
- For weakly-correlated independent IP addresses between 200.0.0.0 and 200.255.255.255, $N \approx 0.29$ million IP addresses.
- For strongly-correlated independent IP addresses between 200.0.0.0 and 200.7.255.255, $N \approx 0.05$ million IP addresses.

Hence, it is clear that the chance of generating a false positive is much higher in the case of mutually independent IP addresses than in the case of deterministically correlated (but otherwise independent) IP addresses.

Moreover, it is essential to note the performance of case 2 is much better than that of case 1. First, we report N for mutually independent IP addresses between 0.1.0.0 and 223.255.255.0 for case 1 and case 2 and then discuss the importance of the number of fragments k versus the fragment size f:

- Case 1: For $n = 1000$, $N \approx 0.9$ million IP addresses.
- Case 2: For $n = 1000$, $N \approx 9.5$ million IP addresses.

Though the number of reconstructed IP addresses of case 2 is approximately 10 times that of case 1 (because of case 2's smaller fragment size), case 2 yields better false positive results than case 1. This is due to the fact that in case 2, more bits are used to communicate the hashed address value. Case 1 uses only four fragments

[2] Strictly speaking, the addresses are independently selected in all cases because of the deterministic nature of the address "correlations" considered here.

to cover part of the H-bit quantity $h(A)$ (refer to section IV) while twelve fragments are used in case 2.

Also note the significantly better improvement in performance over the results quoted at the end of Section III-B for the approach using multiple groups of overlapping fragments rather than a “scrambling” hash.

The speed of execution of the algorithm identifying attacking addresses depends on the fragment size f and the degree of correlation between border routers IP addresses. A larger fragment size (and correspondingly larger numbers of overlapping bits) results in fewer reconstructed IP addresses and, hence, a smaller execution time. Let t be the execution time in seconds, where all simulations were conducted on an 800 MHz Pentium III Linux workstation with 128MB of RAM.

Case 1: $f = 13$ and $k = 8$

- For mutually independent IP addresses between 0.1.0.0 and 223.255.255.0,
  - For $n \leq 800$, $t \leq 1$
  - For $n \leq 1500$, $t \leq 9$.
- For weakly-correlated independent IP addresses between 200.0.0.0 and 200.255.255.255, $t \leq 1$ for $n \leq 1500$.
- For strongly-correlated independent IP addresses between 200.0.0.0 and 200.7.255.255, $t \leq 1$ for $n \leq 1500$.

Case 2: $f = 12$ and $k = 16$

- For mutually independent IP addresses between 0.1.0.0 and 223.255.255.0,
  - For $n \leq 400$, $t \leq 1$
  - For $n \leq 1500$, $t \leq 73$.
- For weakly-correlated independent IP addresses between 200.0.0.0 and 200.255.255.255, $t \leq 1$ for $n \leq 1400$.
- For strongly-correlated independent IP addresses between 200.0.0.0 and 200.7.255.255, $t \leq 1$ for $n \leq 1500$.

VI. Summary

We considered the traceback problem of distributed denial-of-service attacks prevalent in the Internet today. In particular, a solution to this problem based on packet marking was described. We gave an overview of existing packet marking strategies and proposed a mechanism involving overlapping fragments of the unmodified marking router IP address along with fragments of a “scrambling hash” mapping of the address. At a victim end-system, reconstruction of the unmodified address is checked against received hash-value fragments. The hash function effectively decorrelates correlated “attacking” addresses causing false positives. Our approach was shown to produce less than 1% false positives for on the order of 1000 attacking addresses. In addition, address reconstruction complexity is quite low.

Acknowledgment

The authors would like to thank Cisco Systems, Inc. for partially supporting this research by a Cisco URP grant.

References

[1] S.M. Bellovin. ICMP traceback messages. *IETF Internet Draft: draft-bellovin-itrace-00.txt*, 2000.

[2] H. Burch and B. Cheswick. Tracing anonymous packets to their approximate source. In *Proc.* USENIX LISA, Dec. 2000.

[3] H. Chung. An evaluation on defensive measures against denial-of-service attacks. Technical report, Dept of CS, USC, Fall 2002.

[4] T.E. Daniels. *Reference Models for the Concealment and Observation of Origin Identity in Store-and-forward Networks*. PhD thesis, Purdue University, Dec. 2002.

[5] T.W. Doeppner, P.N. Klein, and A. Koyfman. Using router stamping to identify the source of IP packets. In *Proceddings of ACM Conference on Computer and Communications Security*, pages 184–189, 2000.

[6] P. Ferguson and D. Senie. Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. *IETF RFC 2267*, Jan. 1998.

[7] I. Hamadeh and G. Kesidis. Border router packet marking for traceback of DDoS attacks. *CSE Dept, PSU, technical report and PSU patent disclosure*, July 12, 2001.

[8] I. Hamadeh and G. Kesidis. Packet marking for traceback of illegal content distribution and ddos attacks. In *Proc. Cross-Media Service Delivery (CMSD)*, Santorini, Greece, May 2003.

[9] R. Lemos. DoS attacks underscore net’s vulnerability. http://news.com.com/2100-1001-267709.html, Jun. 2001.

[10] Cisco Ltd. IP Source Tracker. In *http://www.cisco.com/ univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/ 120s21/ipst.htm*.

[11] G.R. Malan. Observations and experiences tracking denial-of-service attacks across a large regional ISP. Technical report, The North American Network Operators’ Group, 2001.

[12] R. Naraine. Massive DDoS attack hit DNS root servers. http://www.esecurityplanet.com/trends/article/0,,10751_148698 1,00.html, Oct. 2002.

[13] K. Park and H. Lee. On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack. In *Proc. IEEE INFOCOM*, pages 338–347, 2001.

[14] R. Rivest. The MD5 message-digest algorithm. *IETF RFC 1321*, Apr. 1992.

[15] G. Sager. Security fun with OCxmon and cflowd. In *Internet 2 Working Group Meeting*, Nov. 1998.

[16] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for IP traceback. In *Proc. ACM SIGCOMM*, pages 295–306, Stockholm, Sweden, Aug. 2000.

[17] A.C. Snoeren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio, S.T. Kent, and W.T. Strayer. Hash-based IP traceback. In *Proc. ACM SIGCOMM*, 2001.

[18] A.C. Snoeren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio, S.T. Kent, and W.T. Strayer. Single-packet ip traceback. *IEEE Trans. Networking*, pages 721–734, Dec. 2002.

[19] D.X. Song and A. Perrig. Advanced and authenticated marking schemes for IP traceback. In *Proc. IEEE INFOCOMM*, Apr. 2001.

[20] R. Stone. An IP overlay network for tracing DoS floods. In *Proc.* USENIX *Security Symposium*, Denver, Co, Jul. 2000.

[21] T. Wolverton and G. Sandoval. Leading web sites under attack. http://news.com.com/2100-1017-236683.html, Feb. 2000.

[22] S.F. Wu, L. Zhang, D. Massey, and A. Mankin. Intention driven ICMP traceback. *IETF Internet Draft: draft-wu-itrace-intention-00.txt*, Feb. 2001.