Available online at www.sciencedirect.com
SCIENCE DIRECT
Computer Communications 29 (2006) 2269–2280
computer communications
www.elsevier.com/locate/comcom

Distributed packet pairing for reflector based DDoS attack mitigation

Basheer Al-Duwairi *, G. Manimaran

Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA

Received 8 December 2004; received in revised form 21 February 2006; accepted 1 March 2006
Available online 31 March 2006

Abstract

Reflector based DDoS attacks are feasible in variety of request/reply based protocols including TCP, UDP, ICMP, and DNS. To mitigate these attacks, we advocate the concept of victim assistance and use it in the context of a novel scheme called pairing based filtering (PF). The main idea of the PF scheme is to validate incoming *reply* packets by pairing them, in a distributed manner, with the corresponding *request* packets. This pairing is performed at the edge routers of the ISP perimeter that contains the victim rather than at the edge router to which the victim is directly connected, leading to protection from bandwidth exhaustion attacks in addition to the protection from victim's resource exhaustion attacks. We evaluate the proposed scheme through analytical studies using two performance metrics, namely, the probability of allowing an attack packet into the ISP network, and the probability of filtering a legitimate packet. Our analysis shows that the proposed scheme offers a high filtering rate for attack traffic, while causing negligible collateral damage to legitimate traffic.

Keywords: Network security; DDoS attacks

1. Introduction

Current Internet infrastructure is exposed to many serious threats that can affect the availability of important Internet services. Denial of Service (DoS) attacks [21,32,35] and a more complicated version known as Distributed DoS (DDoS) [23,37] are the most common. These attacks deny regular Internet services from being accessed by legitimate users either by blocking service completely or by disturbing it such that users become not interested in the service anymore (for example causing significant delay in accessing an airline reservation web site). Today's Internet has witnessed several incidents that confirm the devastating effect of such attacks. For example, in October 2002, eight out of the thirteen root DNS servers were affected as a result of severe flooding denial of service attack [20].

In DDoS attacks, the attacker's objective is to overpower the victim while keeping his identity unknown. DDoS attacks are broadly classified as *direct* and *reflector* based attacks. The basic idea behind direct DDoS attacks is to compromise one or more machines (the masters), which will in turn instruct previously compromised innocent machines (the slaves) to aggressively overwhelm the victim by high volume streams of flooding packets with faked IP source addresses, leaving the victim with no clue about the true sources of these packets. Several research efforts (e.g., [8,10,2,3,14,15,24,30,31]) have been made to trace such attacks.

Reflector based DDoS attacks [13,26] employ different strategy. In such attacks, each of the compromised machines is instructed to continuously send request packets to a set of Internet reflectors (an Internet reflector is an IP host that will reply to any request packet). The source address of each of these request packets is spoofed to be the same as the address of the targeted site. As a result, the reflectors send their replies to the given address causing packet flooding at that

* Corresponding author. Present address: Faculty of Computer & Information Technology, Department of Computer Engineering, Jordan University of Science & Technology, Irbid 22110, Jordan. Tel.: +962 2 7201000x22503; fax: +962 2 7095046.
E-mail addresses: basheer@just.edu.jo (B. Al-Duwairi), gmani@iastate.edu (G. Manimaran).

0140-3664/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2006.03.007

site. Using Internet reflectors complicates the problem of DDoS attacks. Researchers are more concerned about these attacks because attack packets (reply packets originated from the reflectors themselves) carry legitimate IP source addresses making it useless to trace such attacks. Also, because these attacks are usually characterized by an amplification factor that increases their intensity. For example, in Smurf attacks [6], the attacker sends ICMP echo requests (pings) to the broadcast address of a network, so the victim is hit by many more packets. The Fraggle attack [7] uses UDP echo packets in the same fashion as the ICMP echo packets. In TCP-based reflector DDoS attack [13], the attacker sends SYN packets to many reflectors. Each corresponding SYN-ACK packet is then sent to the victim. Attack amplification is achieved through the multiple retransmission of SYN-ACK packets after each time out as explained in Section 2.

In this paper, we advocate the concept of victim assistance and propose a novel scheme to mitigate TCP-based reflector DDoS attacks. The proposed scheme is *general* in the sense that it can handle different types of reflector attacks. However, our discussion will be focused on TCP-based reflector attacks due to the following reasons:

- TCP carries 95% of today's Internet traffic and 80% of the total number of flows in the Internet [22]. Therefore, attackers prefer using TCP-based traffic to avoid detection.
- Any general purpose TCP connection-accepting Internet server could be used as a packet reflection server. This provides attackers with large pool of servers to be used as reflectors.
- Several incidents of TCP-based reflector attacks has been reported (e.g., in addition to the reported attacks against GRC.com [13], many TCP-based reflector attacks were captured at Los Nettos ISP network [16]).
- Different than other types of reflector DDoS attacks, TCP-based reflector attack cannot be mitigated by blind filtering of attack packets, because such solution would prevent the victim itself from establishing any TCP connection.

The proposed scheme, called pairing based filtering (PF), is based on informing edge routers of an ISP network about TCP connection establishment requests initiated by the victim, such that incoming reply packets can be paired with them, while replies that represent attack packets can be filtered directly. It is important to mention that the basic idea of request/reply pairing is not new as it has been used earlier for DoS attack detection and mitigation (e.g., SYN-dog [35], D-WARD project [23], and TCP flooding blocking [12]). However, different than earlier work, request/reply pairing in our scheme is performed in a distributed manner at the edge routers of the ISP network that contains the victim, which is a challenging problem given the possibility of routing asymmetry, rather than at the edge router to which the victim is connected. *This feature has the advantage of filtering attack packets far away from the victim* (*i.e., at the ISP perimeter*), *providing protection from bandwidth exhaustion attacks in addition to the protection from victim's resources exhaustion attacks*. The proposed scheme ensures the filtering of most of the attack traffic, while minimizing the collateral damage to legitimate traffic. In this paper, we discuss the design and implementation of the PF scheme and we characterize its behavior. Also, we perform preliminary statistical analysis to study the effects of traffic unpredictability (represented by routing instability and asymmetry), and other design parameters on the performance of the PF scheme. We show that the probability of filtering a legitimate packet is less than 0.001 when suitable parameters are chosen.

The rest of this paper is organized as follows: Section 2 provides detailed prospective about TCP-based reflector DDoS attacks. Section 3 highlights the main motivations and objectives of this paper. Section 4 is devoted for describing the proposed scheme. Section 5 reviews the relevant work. Finally, conclusions are drawn in Section 6.

2. Reflector based DDoS attacks

The analysis presented in [26] show that reflector based attacks are feasible in variety of request/reply based protocols including TCP, UDP, ICMP, and DNS. After determining the type of reflected attack packets, ISP edge routers can be configured to filter packets that share the same attack packet attributes, such as protocol type, destination IP address, and destination port number. While ICMP and DNS based reflector DDoS attacks can be handled by installing such filters at edge routers, *TCP-based reflector DDoS attacks represent a real challenge when protecting victims who are in critical need to establish connections with other systems outside the ISP perimeter*. This is due to the fact that filtering SYN-ACK packets blindly will prevent legitimate SYN-ACK packets from reaching their destination, and consequently, it leads to connection establishment failure.

In TCP protocol, the connection is initially established via the well known three way handshaking procedure. The source sends a SYN packet specifying the port number of the destination that it wants to connect to, and its initial sequence number (ISN). When the destination receives the source's SYN packet, it typically allocates memory buffers for sending and receiving the connection's data, and it records the various details of the connection including the source's remote IP and connection port number. The destination responds with its own SYN packet containing its initial sequence number. It also acknowledges the source's SYN by sending an acknowledgment packet that holds the source's ISN plus one. In this way, the destination will be prepared to accept the source's final connection opening ACK packet. Also, if the source's ACK packet should fail to arrive, the destination will be able to resend its SYN-ACK packet, presuming that it might have been lost or dropped by an intermediate Internet router.

Attackers may take advantage of the availability and connectivity of large number of Internet reflectors to

coordinate a highly distributed DoS attack. This can be done by abusing the TCP protocol in the following way. An attacker, A, selects a set of Internet reflectors. It then sends low rate faked SYN packets to each of these reflectors with a spoofed source address equals to that of the final target V. For each received SYN packet, the reflectors reply with a SYN-ACK packet to the given address V. Therefore overwhelming the victim site, V, by high aggregate rate SYN-ACK packets.

The normal reaction of the victim, V, is to respond by sending RST packet for each received unexpected SYN-ACK packet. From the prospective of a reflector, this can be an indication of ongoing DDoS attack against V. However, this cannot be confirmed because of the low rate of originally received SYN packets. In some cases, the victim becomes disabled and cannot respond to the received SYN-ACK packets coming from the reflectors. Therefore, these reflectors assume transmission failure and decide to retransmit their SYN-ACK packets (multiple times) leading to *attack amplification*.

3. Motivation and objectives

When considering TCP-based reflector DDoS attacks, the following questions come to mind: Is it a real threat? and if so, what type of targets need protection? where to deploy the defense mechanism? and, what are the main objectives we need to achieve in defending against these attacks? The following subsections are devoted to answer these questions.

3.1. The need for protection from TCP-based reflector attacks

In TCP-based reflector DDoS attacks, the target can be a web server that accepts connections from clients throughout the Internet. In most cases, a web server does not initiate TCP connections by itself. Therefore, TCP-based reflector DDoS attacks against such systems can be mitigated by filtering any incoming SYN-ACK packet. However, there exist several scenarios where the targeted system requires to initiate TCP connections with other systems across the Internet. Protection of such system from reflected attack SYN-ACK packets is not trivial because it is difficult to distinguish these packets from legitimate SYN-ACK packets that are necessary for connection establishment. Here, we list few examples of servers that initiate TCP connections which could become *targets* of TCP-based reflector attacks.

3.1.1. FTP server [28]

In active mode FTP, the client connects from a random unprivileged port (Port Number $PN > 1024$) to the FTP server's command port, port 21. Then, the client starts listening to port $PN + 1$ and sends the FTP command "PORT $PN + 1$" to the FTP server. The server will then connect back to the client's specified data port from its local data port, which is port 20.

3.1.2. Proxy server [34]

A proxy server receives a request for an Internet service (such as a Web page request) from a user. If it passes filtering requirements, the proxy server looks in its local cache of previously downloaded Web pages. If it finds the page, it returns it to the user without needing to forward the request to the Internet. If the page is not in the cache, the proxy server, acting as a client on behalf of the user, uses its own IP address to request the page from the server out on the Internet. When the page is returned, the proxy server relates it to the original request and forwards it on to the user. Although using a proxy server in some cases is optional and can be avoided if it becomes under attack, some ISP's make all their users use a proxy servers to block sites with unsuitable content. Therefore, a lot of management overhead would be incurred if ISP customers were asked to reconfigure their web browsers to avoid connecting to the Internet via a proxy server under attack.

3.1.3. SOCKS server [18]

It is a function that is used to manage the connections between clients/servers on a secure internal network and clients/servers on an untrusted network such as the Internet. The SOCKS server sits between the trusted and untrusted systems. The server regulates which connections are allowed, logs information regarding the connections, and hides the internal network information (such as internal IP addresses). Hosts outside the secured network perceive the SOCKS server as the source of the communication. The server resends requests and responses between the trusted and untrusted systems.

3.2. Where to deploy the defense scheme

Home users, business firms, academic institutions, and medical centers usually obtain access to the Internet via a local Internet Service Provider (ISP). The main task of an ISP network is to route Internet traffic between its customer networks, and to provide access to the rest of the Internet by connecting to other ISPs. Fig. 1 shows an ISP network that contains a targeted system in one of its customer networks. Any traffic going to the targeted system (the victim) must enter the ISP network first by passing through an edge router. Hence, the perimeter is the earliest location for defense such that attack packets can be identified and dropped before reaching the victim, leading to protection from bandwidth exhaustion attacks.

3.3. Main objectives

We believe that a successful attack mitigation scheme should achieve the following objectives:

- *Accuracy:* the ability to filter attack traffic with minimal collateral damage to legitimate traffic going to the victim. This requires an efficient method for packet classification.

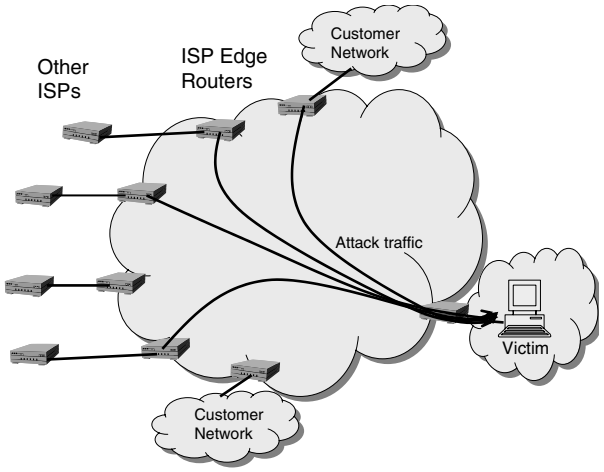

Fig. 1. ISP network. Attack traffic targeting the victim can originate from several directions. However, it must enter the ISP network first by passing through an edge router.

- *Low overhead:* the overhead imposed by the scheme should be minimal. Only simple state information may be maintained at edge routers for the purpose of packet classification.
- *Early filtering:* attack traffic must be filtered before it enters the ISP network that contains the victim in order to avoid congestion inside the network.
- *Fruitful deployment:* the direct benefit of deploying an attack mitigation scheme should be felt by the victim in order to convince the administrators of the ISP network that contains the victim to deploy the scheme.
- *Ease of deployment:* to simplify the deployment in terms of management and operation, the scheme should be deployed in a single administrative domain such as an ISP network.

4. Pairing based filtering (PF)

4.1. Assumptions

Our methodology in mitigating the effect of TCP-based reflector DDoS attacks is based on inherent features of the attack itself and the deterministic nature of the TCP protocol, and more importantly, it depends on the stability and symmetry of Internet routing. Here, *routing stability* refers to the situation of having the same ingress router for all packets sent from a given source to a destination located in the ISP network, while *routing symmetry* refers to the situation of having identical egress and ingress edge router for a SYN packet and its corresponding SYN-ACK packet.

We assume routing stability[1] based on earlier studies [25]. This assumption leads us to conclude that there is a mapping between the reflectors used by an attacker and the edge routers of the ISP network which contains the targeted system. By this mapping we mean that each subset of reflectors used by the attacker have to forward their attack packets through the same edge router during the attack period.

We do not necessarily assume routing symmetry. Instead, we assume that a mechanism such as loose source routing [34] can be used to force a SYN packet generated by the victim to pass through the edge router at which the corresponding SYN-ACK packet is expected to arrive. In fact, having routing symmetry would facilitate the pairing of SYN packets generated by the victim and their corresponding SYN-ACK packets (this is the bases of our proposed scheme).

4.2. The PF scheme

Packet Pairing based Filtering (PF) scheme is a SYN-ACK packet classification and filtering scheme to be implemented at the edge routers of the ISP network that contains the targeted system, and should be activated after detecting TCP-based reflector DDoS attack. The main idea of the proposed PF scheme is as follows: since the attacker keeps using the same set of reflectors continuously during the attack, subsequent SYN-ACK packets from the same reflectors can be filtered at their ingress points to the ISP network unless they correspond to previously sent SYN packets by the victim.

The basic architecture of PF scheme can be viewed as a two level filter. Generally, SYN-ACK packets going to the victim can be classified as legitimate, attack, or suspicious. The filter is designed such that legitimate packets are passed directly, attack packets are dropped directly, and suspicious packets are marked and passed. Initially, all packets are assumed to be suspicious. However, by taking advantage of the nature of reflector DDoS attack and the deterministic behavior of the TCP protocol itself, subsequent packets can be classified with high accuracy. Therefore, the two levels of the filter are:

- Level 1: All legitimate packets are passed directly. Therefore, only suspicious packets and attack packets should be inspected at level 2 of the filter.
- Level 2: All suspicious packets are passed. However, any attack packet will be filtered.

The following data structures are required at each edge router to implement the classification and filtering functions of the PF scheme:

- *Legitimate packet list (LPL):* LPL contains list of SYN-ACK packets that are expected to arrive at the edge router. We use the combination of source IP address and ACK number to uniquely identify certain packet. LPL is updated frequently by forcing SYN packets generated by the victim for a given destination to exit the ISP perimeter at the same edge router at which the

[1] This assumption is relaxed in Section 4.4.

corresponding SYN-ACK packet is expected to arrive[2] (this is achievable using loose source routing [34]). The knowledge about expected SYN-ACK ingress point is achieved by keeping up to date table at the victim, in which the ingress point of each received suspicious SYN-ACK packet is recorded. This can be achieved by marking these packets (at their ingress points) by a code that uniquely specifies the edge router through which the packet has been forwarded (i.e., the ingress point).

- *Source filtering list (SFL):* The SFL contains the list of reflectors that are being continuously used by the attacker. It is built gradually while passing suspicious packets deterministically. When a suspicious packet is passed, its source address is inserted into the SFL, such that subsequent packets from the same source are filtered unless they are found to be in the LPL.

The PF scheme is initiated by the victim upon detection of TCP-based reflector DDoS attack. All ISP's edge routers are informed (through an authentic multicast message) to activate the PF scheme. It is important to mention that the same procedure is performed at all edge routers of a given ISP, and this procedure is applied only to SYN-ACK packets going to the victim.

Fig. 2 shows the basic operation of the PF scheme at a given edge router. When the edge router receives a SYN-ACK packet going to the victim, this SYN-ACK packet must correspond to previously sent SYN packet to be considered as legitimate. The router can determine the legitimacy of the given packet by inspecting the LPL. If packet legitimacy cannot be established, then it has to go through the second level of filtering, in which the packet is dropped directly if its source address is found in the SFL (i.e., attack packet). Otherwise (i.e., suspicious packet), the packet is marked, and passed after inserting its source address in the SFL.

4.3. Implementation of LPL and SFL

4.3.1. Bloom filters

For efficient packet processing and storage, LPL and SFL are implemented using Bloom filters[3] can be implemented using any other data structure [4]. What follows is a description of Bloom filters which is adopted originally from [5]. A Bloom filter is a data structure for representing a set of n elements (also called keys) to support membership queries. The idea is to allocate a vector R of m bits, initially all set to 0, and then choose k independent hash functions, each with range $\{1,\ldots,m\}$. For each element, A, the bits at positions $H_1(A)$, $H_2(A),\ldots,H_k(A)$ in R are set to 1. (A particular bit might be set to 1 multiple times.) Given a query for B, we check the bits at positions $H_1(B)$, $H_2(B),\ldots,H_k(B)$. If any of them is 0, then certainly B is not inserted in the filter. Otherwise we conjecture that B is inserted in the filter although there is a certain probability that we are wrong. This is called a "false positive". The parameters k and m should be chosen such that the probability of a false positive is acceptable. It has been shown in [4] that the false positive rate of a bloom filters is given by the following equation:

$$p_f = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \approx (1 - e^{-\frac{kn}{m}})^k. \quad (1)$$

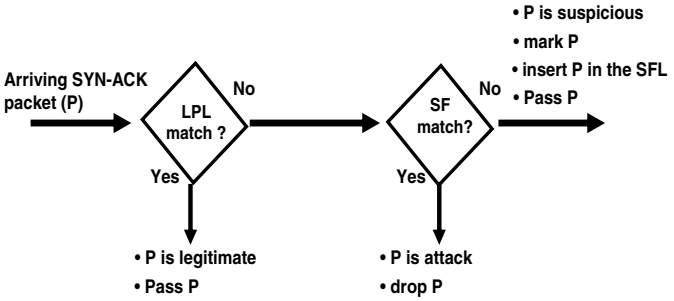

Fig. 2. Basic operation of the PF scheme.

4.3.2. The PF algorithm

We incorporate the idea of using Bloom filters in our scheme to support efficient packet classification and filtering at high Internet speeds. Fig. 3 shows the PF algorithm at edge router ER_x. An incoming SYN-ACK packet P is inspected for legitimacy (step 1). This is done by inspecting the LPL bits indexed by $H_1(P.\text{source}, P.\text{ack-number})$, $H_2(P.\text{source}, P.\text{ack-number}),\ldots,H_k(P.\text{source}, P.\text{ack-number})$. This combination (i.e., P.source and P.ack-number)

PF Algorithm at Edge Router ER_x

- for each SYN-ACK packet, P, received at edge router ER_x and going to victim V
  1. if (Packet P is in the LPL)
     a. pass P
     b. erase P from the LPL
  2. else if (P.source is in the SFL) drop P
  3. else
     a. insert P.source in the SFL
     b. mark P
     c. pass P

Fig. 3. PF algorithm at edge router ER_x.

[2] If the victim does not have knowledge about the expected ingress point of the corresponding SYN-ACK packet, then the SYN packet is sent normally.

[3] We use Bloom filters as an example for implementing LPL and SFL and study the scheme's effectiveness accordingly. In practice any other data structure could be used.

should be the same as the combination of destination IP address and (SYN number + 1) of the corresponding SYN packet that is supposed to be already inserted in the LPL. If the packet is found to be legitimate, then it is passed (step 1.a), and its entries are erased from the LPL table (step 1.b).

The packet is dropped if its legitimacy cannot be established and its address is found to be in the SFL (step 2). SFL membership test (i.e., to check if certain source is in the SFL) is done by inspecting the bits indexed by $H_1(P.\text{source})$, $H_2(P.\text{source}),\ldots,H_k(P.\text{source})$. Otherwise, the packet is considered suspicious (step 3) and its address is inserted in the SFL by setting the SFL bits indexed by the same values used for membership test such that future SYN-ACK packets from the same source are classified as attack packets after failing the legitimacy test. The packet is also marked by setting a 1-bit flag that indicates that the packet is suspicious, and by augmenting it by a code that uniquely represents the edge router through which the packet is passed. For an ISP network with N edge routers, $\lceil \log_2 N \rceil$ bits would be required to represent each edge router. Table 1 shows the number of edge routers in different ISP networks. It can be seen that the number of bits required for edge router encoding is ISP dependent, and in most cases, it is less than 13 bits. Such code can be written in the 16 bit ID of the IP packet header.

4.3.3. Packet marking

The purpose of marking suspicious packets is two fold. By receiving a suspicious packet, *SP*, the victim knows that the source address of the given packet has been inserted in the SFL of the edge router represented by the code obtained in the marked packet. If the packet is found to be legitimate, the victim should send an erase request to that edge router to remove the source address of the packet from its SFL. This is done by resetting the SFL bits indexed by $H_1(SP.\text{source})$, $H_2(SP.\text{source})$, $\ldots,H_k(SP.\text{source})$. Failing to do so may result in setting all SFL bits as a result of continuous insertions of newly seen sources, which leads to complete blocking of legitimate packets.

The marking at the edge router helps the victim in updating its information about the mapping between different packet sources and ISP edge routers. This information is used to update LPLs whenever a TCP connection is established by the victim. Fig. 4 shows the LPL update algorithm. Whenever the edge router receives SYN packet from the victim itself to establish a connection with certain system outside the ISP perimeter, the combination of the packet destination and (SYN number + 1) is inserted in the LPL by setting the bits indexed by $H_1(P.\text{destination}, P.\text{SYN-number} + 1)$, $H_2(P.\text{destination}, P.\text{SYN-number} + 1)$, $\ldots,H_k(P.\text{destination}, P.\text{SYN-number} + 1)$.

Table 1
Number of edge routers in various ISP networks

ISP	Number of edge routers (N)	$\lceil \log_2 N \rceil$
AT&T (US)	8044	13
Ebone (Europe)	108	7
Exodus (US)	49	6
Level 3 (US)	875	10
Sprintlink (US)	5990	12
Telstra (Australia)	2249	12
Tiscali (Europe)	182	8
Verio (US)	2846	12
VSNL (India)	10	4

This information was extracted from the Rocketfuel [33] ISP topology raw traces.

LPL Update Algorithm at Edge Router ER_x

- for each SYN packet P received from victim V at edge ER_x
  - insert P in the LPL

Fig. 4. LPL update algorithm at edge router ER_x.

4.4. Theoretical analysis of the PF scheme

Ideally, PF scheme should filter all attack packets without causing any collateral damage to legitimate packets. However, due to design and implementation factors of the scheme, and due to Internet traffic unpredictability, ideal operation of the scheme cannot be achieved. The following properties characterize the behavior of the PF scheme:

- PF scheme does not provide instant filtering of attack packets at the moment the scheme is activated. Meaning that the edge routers will experience some delay before being able to classify incoming SYN-ACK packets with high accuracy. This delay is due to the gradual process of building the SFL at each edge router.
- Ideally, all legitimate SYN-ACK packets going to the victim are allowed to pass the ISP perimeter. The validity of this property depends on the knowledge of the victim about the expected ingress point for each SYN-ACK packet, and on the *symmetry* of egress/ingress of request/reply packets from/to the victim. This is because failing to inform the correct edge router about an incoming legitimate SYN-ACK packet may lead to filtering of that packet.
- Ideally, only one packet from each attack source is allowed to pass the ISP perimeter. This is because the source address of the first attack packet from a given source would be inserted in the SFL causing subsequent packets from the same source to be filtered. This property does not hold always due to *routing instability* which allows packets coming from the same source to change their ingress point to the ISP network. If an attack packet from certain source enters the ISP perimeter at an edge router different than the one at which previous packets from the same

source arrived, then such packet would be allowed to pass since its source address is not in the SFL of the current edge router.[4]

Our aim is to evaluate the PF scheme in terms of the following performance metrics which map to the previous properties:

- Transient defenseless period (TDP): the period since the scheme is activated until all distinct attack sources are identified.
- P_l: probability of filtering a given legitimate SYN-ACK packet.
- P_a: probability of allowing more than one packet from a given attack source (i.e., reflector) to pass the ISP perimeter.

4.4.1. TDP analysis

It is imperative to mitigate the effect of DDoS attack within short period of time. Complete mitigation[5] of the attack is achieved only by identifying all distinct attack sources (i.e., reflectors being used by the attacker) because attack packets will be subject to deterministic dropping if their sources are in the SFL. There is a tradeoff from the perspective of the attacker, between available resources (e.g., available bandwidth), detection avoidance by the reflectors themselves (e.g., high rate SYN packets from certain source may be considered as an indication for ongoing reflector attack), and amount of damage desired at the targeted system.

Assuming that edge router ER_x receives attack packets from the n different reflectors represented by the set $X = \{x_1, x_2, \ldots, x_n\}$ with an average rate of b packets/second for each reflector, then we need to find the average number of attack packets, M, required to insert all attack reflector addresses in the SFL. This problem is an instant of the well known *coupon collector problem*. It has been shown that $M = n\left(1 + \cdots + \frac{1}{n-1} + \frac{1}{n}\right)$ [29]. Therefore, the average time to collect all distinct attack reflector addresses can be expressed as $\frac{M}{nb}$. This value represents the average transient defenseless period at router ER_x. In practice, the average time would be larger due to the ramp up behavior observed in DDoS attacks in general [16].

4.4.2. P_l analysis

The probability of dropping a legitimate SYN-ACK packet depends on whether the packet source is *new* to the victim (i.e., the first time to be contacted by the victim since the scheme is activated), or it is *old* (i.e., it has been contacted earlier by the victim after the scheme was activated). The following analysis focuses on estimating the percentage of legitimate traffic that may get dropped. This analysis apply for packets that arrive after the transient defenseless period.

The importance of an address being new or old reflects the knowledge of the mapping between the address and the ingress point to the ISP network that contains the victim. SYN-ACK packets coming from new source addresses will be blocked with probability given by

$$P_{\text{bnew}} = P_{\text{asymm}} P_f, \tag{2}$$

where P_{asymm} represents the probability of egress/ingress asymmetry (i.e., the probability that a given SYN packet exits the ISP perimeter at certain edge router, while the corresponding SYN-ACK packets enters the ISP perimeter at different edge router), and P_f represents the false positive rate of the bloom filter that represents the SFL. On the other hand, SYN-ACK packets coming from old source addresses are expected to be blocked with probability given by

$$P_{\text{bold}} = P_{\text{inst}} P_f, \tag{3}$$

where P_{inst} represents the probability of routing instability. Let P_{new} be the probability that the source of a legitimate SYN-ACK packet is new to the victim. From equations 2 and 3, the probability of blocking legitimate SYN-ACK packet is given by

$$P_l = P_{\text{new}} P_{\text{asymm}} P_f + (1 - P_{\text{new}}) P_{\text{inst}} P_f. \tag{4}$$

It is important to realize that P_{new} is not fixed. Initially, all legitimate SYN-ACK sources are new to the victim. However, as the time proceeds, most of the legitimate SYN-ACK sources become old.

To study the effect of routing instability and routing asymmetry separately, we fix one of them and vary the other along the possible values of P_{old} (the same as $1 - P_{\text{new}}$). The SFL parameters m, k, and $n_{\max}$ were set to $128k$ bits, 4, and 5000, respectively. This setting corresponds to a P_f of 0.0004. Fig. 5 shows the effect of routing instability on P_l. It can be seen that the blocking probability of legitimate packets increases when the routing instability increases, which can be explained by recalling that a legitimate SYN-ACK packet coming from old sources cannot be paired with the corresponding SYN packet if the it arrives at an edge router different than the expected one.

Fig. 5 shows the effect of routing asymmetry on P_l. It can be seen that P_l increases when the routing asymmetry increases, which can be explained by recalling that a legitimate SYN-ACK packet coming from new sources cannot be paired with the corresponding SYN packet if it arrives at an edge router different than the one at which the SYN packet departed the ISP perimeter. Overall, the small values of P_l, which is in the range of 10^{-3}, indicates that collateral damage under PF scheme is minimal.

4.4.3. P_a analysis

The probability of allowing more than one packet from the same attack source can be expressed as

[4] An exception to this is when a false positive occurs to the packet's source address. In this case the packet would be filtered.

[5] This is in ideal situation.

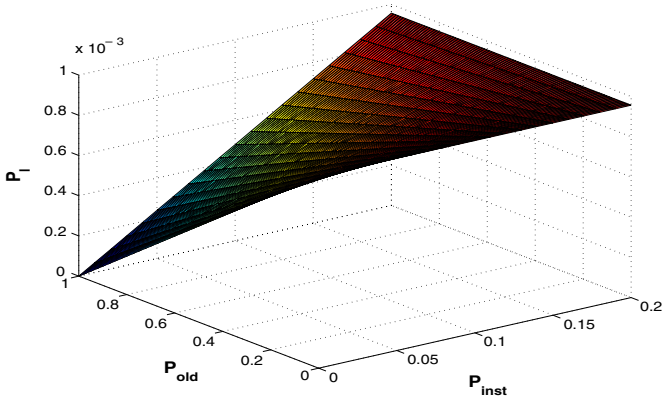

Fig. 5. Probability of filtering a legitimate SYN-ACK packet (P_l). P_{asymm} and P_f were fixed to 0.2 and 0.0004, respectively.

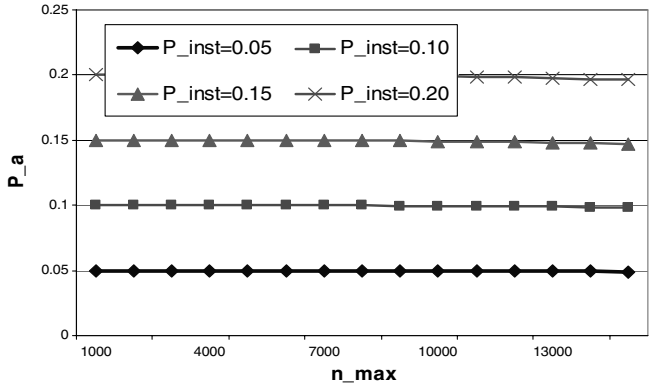

Fig. 7. Probability of allowing more than one packet from the same attack source (i.e., the same reflector).

$$P_a = P_{\text{inst}}(1 - P_f). \tag{5}$$

It can be seen from equation 1 that the effective false positive rate in an individual SFL depends on different design parameters including its size, m, the number of hash functions used, k, and the number of source addresses, n, inserted in it. Obviously, the number of reflectors that are being used by the attacker which map to certain edge router, ER_x, depends on the actual network topology, routing protocol, and attacker's choice. Therefore, parameter n is expected to be different for different edge routers. We can, however, make some simplifying assumptions in order to derive an upper bound on the false positive rate of the SFL at each edge router. We assume that up to $n_{\max}$ reflectors out of those used by the attacker map to any edge router. Fig. 6

For the purpose of discussion, we will consider using a SFL of size 128K bits with four hash functions. We assume different values for the probability of routing instability ranging from 0.05 to 0.2. Fig. 7 shows the value of P_a as a function of $n_{\max}$. It is clear that stable routing (i.e., low values for P_{inst}) reduces the chances for more than one attack packet per source to pass the ISP perimeter. We, also, observe that P_a decreases slightly by increasing $n_{\max}$. In fact, increasing $n_{\max}$ beyond the range shown in the figure leads to sharp decrease in P_a. This can be interpreted by recalling that increasing $n_{\max}$ corresponds to higher values of P_f. This represents one of the constraints imposed on the attacker by the PF scheme which prevents the use of very large number of reflectors, because this would lead to higher P_f and consequently lower P_a. The effect of $n_{\max}$ on P_l can be seen in Fig. 8 which plots P_l under the same conditions. It is clear that P_l increases slightly (observe that P_l remains in the 10^{-3} range) by increasing $n_{\max}$, which is due to the corresponding increase of P_f. By looking at Figs. 7 and 8 together, one can infer the tradeoff regarding $n_{\max}$ from attacker's viewpoint.

4.5. TCP-based reflector DDoS attacks against GRC.com: A case study

As a case study, we consider the reflector-based DDoS attacks launched against Gibson Research Corporation (GRC) in January 2002. In particular, we describe GRC's connection to the Internet, provide a brief description of the attack, then we evaluate the proposed scheme (i.e., the PF scheme) in the context of this attack

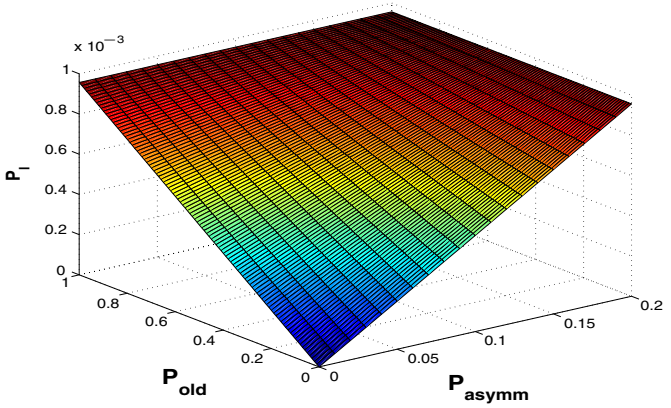

Fig. 6. Probability of filtering a legitimate SYN-ACK packet (P_l). P_{inst} and P_f were fixed to 0.2 and 0.04, respectively.

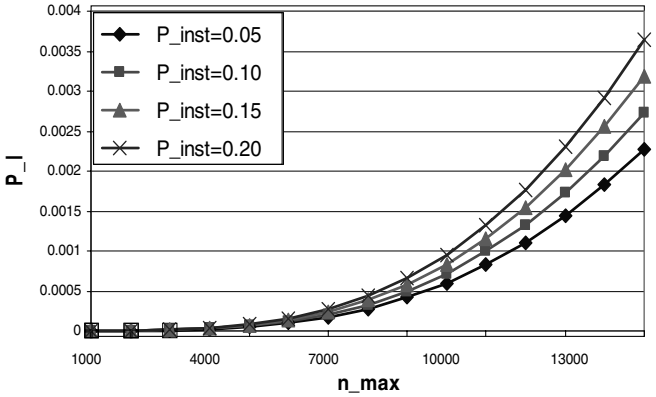

Fig. 8. Probability of filtering a legitimate SYN-ACK packet (P_l). P_{new} and P_{asymm} were fixed to 0.5 and 0.2, respectively.

to see how efficient it would be if it had been deployed. Also this study aims at providing an insight on the typical values of the parameters associated with the PF scheme.

As described in [13], GRC is connected to the Internet via a pair of T1 trunks through a Cisco router that belongs to Verio network (one of the major ISPs in the united states). They provide a total of 3.08 megabits of bandwidth in each direction (1.54 megabits each), which is ample for their daily needs. The Verio router that supplies GRC's T1 trunks enjoys two massive 100 megabit connections to the Internet. But from there all of the traffic bound for GRC must be funnelled through the two T1 trunks.

The attack against GRC as described in [13] was TCP-based reflector DDoS attack. An analysis of the attack traffic showed that GRC's T1 trunks were flooded by SYN/ACK packets originating from hundreds of machines that belongs to Verio, Qwest, yahoo.com, Above.net and many other networks. GRC estimated the number of distinct IP addresses from which reflected SYN-ACK packets originated to be in the range of hundreds. For the purpose of this study we roughly consider the number of attack machines to be 500. Farther more, we assume that these machines are evenly distributed among 10 ISP networks, which means that 50 attack machines map to one of the 10 of the 2846 Verio's edge routers.[6]

GRC reported that the number of attack packets exceeded one billion by the time the attack ended. However, they did not specify the aggregate attack rate. A study conducted by A. Hussain [16] revealed that aggregate rate of reflected attack traffic is typically in the range of 340–13,000 packets/s.[7] As a worst case scenario, we assume that the attack against GRC was as intense as 13,000 packets/s, which corresponds to an average rate of 26 packets/s for each attack machine. Therefore, the average value of TDP under these settings would be around 0.11262 s which is a very small value that indicates a very fast response of the proposed scheme. The small number of attack machines that maps to a given edge router (around 50) implies that an SFL of size 16K bits would be sufficient to mitigate this particular attack with false positive rate of $P_f = 0.000012$. It would be difficult to evaluate the values of P_l and P_a without having measurements of the routing instability and routing symmetry probabilities. However, both values (i.e., P_f and P_l) are expected to be very low due to the limited number of reflectors involved in the attack and due to the fact that these reflectors belong to small number of ISP networks.

[6] The number of edge routers within Verio's network is found to be 2846 [33].

[7] This number is based on the attack traffic captured at Los Nettos edge router which is connected to the Internet via an edge router that belongs to Verio.

4.6. Practical considerations

In this subsection,we discuss the practicality of the PF scheme from security and implementation aspects:

4.6.1. Generality of the PF scheme

It is important to emphasize that the PF scheme is not specific to TCP-based reflector attacks since it has the ability to filter incoming reply packets of any protocol type (assuming that it would be activated according to the attack packets type). In practice, an attacker can just generate random packets from his zombies and flood the target's link. If the attacker particularly likes sending TCP SYN-ACK packets, for some reason, he can fabricate them and send them directly to the target without the help of reflectors. Although this is generally true, it is not useful for the attacker because:

- The amplification effect[8] cannot be achieved this way.
- The zombies under attacker's control can be located if a traceback scheme is employed.
- The PF scheme can still filter attack SYN-ACK packets.

4.6.2. TCP header inspection

The proposed scheme requires that all edge routers of the ISP network that contains the victim to inspect each packet closely enough to determine if it is a TCP SYN-ACK packet, so the mechanism can more closely analyze the packet to see if it is a legitimate SYN-ACK. This requires, at least for all TCP packets destined to the victim, examining the TCP header fields. Existing edge routers may not examine TCP header fields at high speeds, so installing this mechanism at those routers would slow them down. To reduce the overhead imposed on edge routers, a lightweight algorithm similar to the one proposed in [36] could be used to distinguish TCP control packets from TCP data packets. Based on that algorithm, a router can tell TCP control packets from data packets without accessing the TCP header by checking the "total length field" in the IP header. If the total length of an IP packet is 40, then it is most probably a TCP control packet (given that its protocol type is TCP and its fragmentation offset is zero). By following this approach, among the IP packets destined to the victim, only TCP control packets undergo TCP header inspection by edge routers. Once a TCP control packet is identified, the edge router has to inspect the corresponding flags in the TCP header to determine whether it is a SYN-ACK packet or not. This implementation based modification reduces the overhead of TCP header inspection. The actual TCP header inspection can be

[8] Recall that amplification effect in TCP-based reflector attacks is due to multiple packet retransmission.

done by the router itself or by a special device attached to it.

5. Relevant work

Reflector based DDoS attacks can be defeated either by filtering the reflected attack packets (which hold valid IP source addresses), or by solving the origins of the problem (i.e., filtering IP packets with spoofed source addresses). In this section, we discuss the main research efforts that addressed the reflector based DDoS attacks explicitly, and we review some of the research efforts that targeted the filtering of spoofed IP packet.

5.1. Detection and prevention of reflector based DDoS attacks

In [26], filtering of different types of reply attack packets that share certain attributes, such as the destination port number and IP destination address, was considered. However, the issue of collateral damage (i.e., filtering legitimate replies as well) was not addressed. In [27], a distributed approach for detecting reflector attacks was proposed. The approach is based on sharing beliefs among potential reflectors if any abnormal traffic is observed, such that the reflectors become aware that they are being used in a reflector based DDoS attack, and consequently start ignoring incoming request packets that have source address equals to the victim's address. Clearly, this approach cannot be deployed in practice because there is no way by which certain reflector knows the group of reflectors participating in ongoing attack, such that it can share its belief with them. Even if attack detection is possible among set of reflectors, there is no mechanism by which reflectors can distinguish attack packets from legitimate packets. Moreover, it is possible for the attackers to abuse the scheme by sharing their own beliefs with many other innocent reflectors in order to drop legitimate traffic passing through them.

5.2. Filtering of spoofed IP packets

Generally, filtering of spoofed IP packets can be done at the source network, intermediate routers, or at the destination network. For example, in ingress filtering [11], routers are configured to block packets that arrive at the edge router of the source network with illegitimate source addresses. This may violate some existing setups and protocols such as Mobile IP and multi-homing. It is also difficult to convince ISP administrators to support ingress filtering because the benefit is not felt directly by the deploying ISP. Our scheme is different in this aspect because fruitful deployment is one of its main objectives. Another example is the SAVE protocol [19], which is designed to provide routers with the information needed for source address validation. The main problem of this protocol is that legitimate packets may be filtered even in the absence of an attack. This is due to routing instability which leads to errors in the source address validation tables maintained at the SAVE enabled routers. In general, such schemes require large scale deployment to prevent IP source address spoofing efficiently.

Hop count filtering [17] is a simple approach to drop spoofed packets at the destination network. It is based on observing that the distance traveled by a spoofed packet is usually different than that traveled by a packet originated from the actual spoofed source. Therefore, attack packets can be distinguished and dropped directly. The main drawback of this approach is the need to keep up to date database of source addresses and their distances. This might be difficult due to route changes. Also smart attackers may spoof IP addresses that never communicated with the given reflector such that it cannot judge about the validity of these packets.

6. Conclusions

Reflective DDoS attacks are feasible in variety of request/reply based protocols. In this paper, we proposed a victim-assisted based scheme to mitigate such attacks. The proposed scheme, called Pairing Based Filtering (PF), is based on the idea of pairing request packets and their corresponding reply packets in a distributed manner.[9] The pairing is performed at the edge routers of the ISP network that contains the victim, such that attack reply packets can be identified and filtered directly, leading to protection from bandwidth exhaustion. Through analytical studies, we showed that PF scheme offers protection for legitimate packets going to the targeted system during a reflective DDoS attack, while filtering attack packets. Our analysis shows that the probability that a legitimate packet being dropped under the PF scheme is less than 0.001, when suitable parameters are chosen.

Obtaining victim's assistance represents the basis in defending against reflective DDoS attacks in the proposed scheme. The importance of this approach is reflected in the fact that new source of information (i.e., the victim) is now available to make distinction between attack and legitimate packets. This approach opens new directions of research in designing efficient countermeasures for DDoS attacks. For example, (1) using the proposed scheme in such a way to provide a hybrid service of attack prevention and mitigation; (2) investigating the viability of victim's assistance in defending against direct DDoS attacks.

[9] The focus of the paper was on TCP-based reflector DDoS attacks due to the difficulty of handling this particular type of attacks.

References

[2] B. Al-Duwairi, A. Chakrabarti, G. Manimaran, An efficient packet marking scheme for IP traceback, in: Proc. third IFIP-TC6 Networking Conference, Athens, Greece, May 2004.

[3] B. Al-Duwairi, G. Manimaran, A novel packet marking scheme for IP Traceback, in: Proc. of 10th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2004), Newport Beach, California, USA, July 2004.

[4] B.H. Bloom, Space/time trade-offs in hash coding with allowable errors, in: Communications of ACM 13, July 1970, pp. 422–426.

[5] A description of Bloom filters is available at: <http://www.cs.wisc.edu/~cao/papers/summary-cache/node8.html>.

[6] CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks. <http://www.cert.org/advisories/CA-1998-01.html>.

[7] CERT, Trends in Denial of Service Attack Technology. <http://www.cert.org/archive/pdf/DoS_trends.pdf>.

[8] H. Burch, B. Cheswick, Tracing anonymous packets to their approximate source, in: Proc. 2000 USENIX LISA Conference, December 2000, pp. 319–327.

[10] D. Dean, M. Franklin, A. Stubblefield, An algebraic approach to IP traceback, in: Network and Distributed System Security Symposium (NDSS '01), February 2001.

[11] P. Ferguson, D. Senie, Network ingress filtering: defeating denial-of-service attacks which employ IP source address spoofing, RFC 2827, 2000.

[12] Y. Kim, J.-Y. Jo, H.J. Chao, F.L. Merat, High-speed router filter for blocking TCP flooding under distributed denial of service attack, in: International Performance, Computing and Communications Conference, Phoenix, AZ, April 2003.

[13] S. Gibson, Distributed Reflection Denial of Service, February 22nd, 2002. <http://grc.com/dos/drdos.htm>.

[14] M.T. Goodrich, Efficient packet marking for large-scale IP traceback, in: Proc. of ACM CCS 2002, November 2002.

[15] F. Hsu, T. Chiueh, A path information cashing and aggregation approach to traffic source identification, in: Proc. of IEEE International Conference on Distributed Computing Systems (ICDCS), May 2003.

[16] A. Hussain, J. Heidemann, C. Papadopoulos, A framework for classifying denial of service attacks, in: Proc. of ACM SIGCOMM 2003, Germany.

[17] C. Jin, H. Wang, Kang G. Shin, Hop-count filtering: an effective defense against spoofed DDoS traffic, in: Proc. ACM Conference on Computer and Communications Security (CCS)'2003, Washington, DC, October 2003.

[18] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, L. Jones, SOCKS Protocol Version 5, RFC (1928).

[19] J. Li, J. Mirkovic, M. Wang, P. Reiher, L. Zhang, SAVE: source address validity enforcement protocol, in: Proc. of IEEE INFOCOMM 2002, April 2002.

[20] D. McGuire, B. Krebs, Attack on internet called largest ever, < www.washingtonpost.com>, October 2002. <http://www.washingtonpost.com/wp-dyn/articles/A828-2002Oct22.html>.

[21] C. Meadows, A formal framework and evaluation method for network denial of service, in: Proc. IEEE Computer Security Foundations Workshop, June 1999, pp. 4–13.

[22] M. Mellia, I. Stoica, H. Zhang, TCP model for short lived flows, in: Proc. IEEE Communication Letters, February 2002.

[23] J. Merkovic, G. Prier, P. Reiher, Attacking DDoS at the Source, in: Proc. of ICNP 2002.

[24] K. Park, H. Lee, On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack, in: Proc. of IEEE INFOCOM 2001, March 2001.

[25] V. Paxson, End-to-end routing behavior in the internet, in: Proc. IEEE/ACM Transactions on Networking, vol. 5(5): pp. 601–615, October 1997.

[26] V. Paxson, An analysis of using reflectors for distributed denial-of-service attacks, in: Proc. Computer Communication Review vol. 31(3), July 2001.

[27] T. Peng, C. Leckie, R. Kotagiri, Detecting reflector attacks by sharing beliefs, in: Proc. IEEE Global Communications Conference, October 2003.

[28] J. Postel, J. Reynolds, File Transfer Protocol (FTP), RFC 959.

[29] S. Ross, A First Course in Probability, fourth ed., Prentice-Hall, Englewood Cliffs, NJ, 1994.

[30] S. Savage, D. Wetherall, A. Karlin, T. Anderson, Practical network support for IP traceback, in: Proc. of ACM SIGCOMM, August 2000, pp. 295–306.

[31] A.C. Snoeren, C. Partiridge, L.A. Sanchez, C.E. Jones, F. Tchhakountio, S.T. Kent, W.T. Strayer, Hash-Based IP TraceBack, in: Proc. of ACM SIGCOMM, August 2001.

[32] O. Spatscheck, L. Peterson, Defending against denial of service attacks in Scout, in: Proc. USENIX/ACM Symp. Operating System Design and Implementation, February 1999, pp. 59–72.

[33] N. Spring, R. Mahajan, D. Wetherall, Measuring ISP topologies with rocketfuel, in: Proc. of ACM SIGCOMM, August 2002.

[34] A.S. Tanenbaum, Computer Networks, fourth ed., Printice Hall, 2003.

[35] H. Wang, D. Zhang, K.G. Shin, Detecting SYN flooding attacks, in: Proc. of IEEE INFOCOMM 2002, April 2002.

[36] H. Wang, K.G. Shin, Transport-aware IP routers: a built-in protection mechanism to counter DDoS attacks, in: Proc. of IEEE Transactions on Paralle and Distributed Systems, vol. 14, no. 9, Septembr 2003.

[37] D.K.Y. Yau, J.C.S. Lui, F. Liang, Defending against distributed denial of service attacks with max-min fair server-centric router throuttles, in: Proc. of IEEE International Workshop on QoS, May 2002.

Basheer Al-Duwairi received the PhD and MS degrees in computer engineering from Iowa State University in Spring 2005 and Spring 2002, respectively. Prior to this, he received the BS degree in electrical and computer engineering from Jordan University of Science and Technology (JUST) Irbid, Jordan in 1999. He joined the JUST as a faculty member in Fall 2005. The focus of his PhD work was on designing and analyzing practical schemes for mitigating and tracing-back DDoS attacks in the Internet. He has coauthored several research papers in these fields. His research interests are in the areas of Internet security and real-time systems. http://www.just.edu.jo/~basheer.

Manimaran Govindarasu received the PhD degree in computer science and engineering from IIT Madras, India, in 1998. He has been an associate professor in the Department of Electrical and Computer Engineering at Iowa State University since fall 2005; prior to this, he was an assistant professor in the same department from Spring 1999 to Spring 2005. His research expertise are in the areas of trusted Internet encompassing QoS, infrastructure security, and fault tolerance focusing on routing, multicasting, and DDoS issues; and resource management in real-time systems. He has coauthored approximately 100 peer-reviewed research publications, of which two conference/workshop papers received the best paper awards. He is a coauthor of the text Resource Management in Real-Time Systems and Networks (MIT Press, 2001). He has served as guest coeditor for for the IEEE Network special issue on multicasting: an enabling technology, January/February 2003, the Journal of High Speed Networks special issue on trusted Internet, 2005, the Journal of Systems and Software special issue on parallel and distributed real-time systems, July 2005. He is a founding cochair of the Trusted Internet Workshop (TIW) held in conjunction with HiPC. He has given tutorials at reputed conferences and in IEEE ComSoc Tutorials Now, served as a member of technical program committee and session chair in many IEEE conferences. He is a member of the IEEE, IEEE Computer Society, IEEE Communication Society, and ACM. http://www.ee.iastate.edu/~gmani.