IP Traceback Marking Scheme Based DDoS Defense

Ping Yan

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Computer Science and Engineering

Supervised by

Prof. MoonChuen Lee

Abstract of thesis entitled:

IP Traceback Marking Scheme Based DDoS Defense

Submitted by Ping Yan

for the degree of Master of Philosophy

at The Chinese University of Hong Kong in December 2004

This thesis presents an effective means to defend against DDoS attacks by first locating the actual attack source(s) and then removing the attack traffic.

The proposed adaptive packet marking scheme for IP traceback supports two types of marking: router *id* marking and probabilistic domain *id* marking. After collecting sufficient packets, the victim reconstructs the attack graph, with each node in the graph viewed as a domain. Meanwhile, based on the reconstructed attack graph, we defend against DoS attacks with two alternative schemes: packet filtering at victim-end and attack traffic rate-limiting at the sources. At the victim side, we propose to let a number of filtering agents inspect the markings inscribed in the received packets and filter the attack packets according to the attack signatures. At the side of attack sources, we propose to inform the detected source domains to rate-limit the attack traffic.

The simulation results show that the proposed IP traceback scheme outperforms other IP traceback methods as it requires fewer packets for attack paths reconstruction, and can handle large number of attack sources effectively; the success rate of attack sources identification is high and the false positives produced are relatively low. Further, it does not generate additional traffic. The packet filtering mechanism can remove around 80% attack traffic. With the source-end rate-limiting system, the throughput of legitimate traffic is significantly improved from 10% to 90%. Therefore, the throughput of legitimate traffic could be significantly improved, so that the DDoS attacks could be ultimately thwarted.

Keywords: DDoS attacks, DDoS defenses, IP traceback, probabilistic packets marking, inter-domain marking, source router marking, attack graph reconstruction, packet filtering, traffic rate limiting.

基於 IP 反追蹤算法的分散式拒絕服務攻擊防禦

碩士研究生論文摘要

拒絕服務攻擊(DOS)是穩定的 Internet 網路的主要威脅. 阻擋 DDoS 攻擊的難點在於攻擊資料包通常都具有不正確的或“僞裝”的 IP 源位址（IP Spoofing）藉以逃避反向追蹤. 論文給出一個有效的阻擋 DDoS 攻擊的解決方案, 它首先以包標記算法反向追蹤定位攻擊源, 然後清除攻擊數據流.

本文提出的“適應性包標記算法”支援兩種不同的數據包標記: 域(domain) ID 標記和路由器 ID 標記. 當足夠數量的有標記的數據包集合在被攻擊主機，就可以恢復出攻擊所經過的路徑. 試驗結果顯示我們的方法相較於其他 IP Traceback 方法需要更少的數據包重建攻擊路徑, 而且可以以更少的錯誤率處理更多的攻擊源, 同時, 它不會帶來額外的網路負載.

同時, 我們實現了基於適應性包標記算法的 DDoS 防禦和阻擋方案, 它包括被攻擊主機的包過濾機制和源位置的數據流限制機制. 一方面, 包過濾機制由一些過濾點 (filtering agents) 檢查包內標記, 檢測出攻擊包並丟掉它們, 從而 80%左右的攻擊包被清除; 另一方面, 在追蹤到的攻擊源處對攻擊數據流進行限制可以有效的防止攻擊數據流進入網路消耗網路帶寬. 算法有效的減少到達被攻擊主機的攻擊數據流, 從而保證了被攻擊主機的服務可用性.

關鍵字: DDoS 攻擊, DDoS 攻擊防禦, 包標記算法, 域 ID 標記, 路由器 ID 標記, 攻擊路徑恢復, 數據包過濾機制, 攻擊數據流限制機制

Acknowledgement

I would like to express my deepest appreciation to my supervisor, Professor M.C. Lee, who guided me through my research. He provided me the most priceless advice and support on both technical and editorial issues. I can not achieve anything without his patience and step-by-step guidance—from how to read papers, how to propose ideas to how to polish the writings. Both my interest and confidence in doing research also come from his generous encouragement and positive evaluations.

I would also like to thank Professor Irwin King and Professor John Lui on my thesis committee. They actively participated my term talks and have put a great deal of effort on strengthening and improving my research and thesis. Additional gratitude is to the Department of Computer Science and the institution – the Chinese University of Hong Kong that provide me the most convenient and comfortable environment to conduct my research.

Here are my most sincere thanks to all of my friends and colleagues, who are so trustable and dependable when there were technical troubles or other frustrations. They are Wei dan, Zhou lin, Phoebe, Wang ying, Ma bin, Sun haibin, Jiang wenjie, Fan bin, Lushi, Ah Heng, Ah Tai, Paul, Ma tianbai. So many, many people have been there whenever I need help; most of them may not have their names listed here, but I

thank them from the bottom of my heart. I'm obliged to their warm friendship, care and encouragement. I did enjoy a lot being around with them.

I would also like to mention the people in the Skitter Project. They provided me the testing data, which is essential for the simulations of the research.

Needless to say, that I am grateful to all the faculty and staff of the Computer Science department. They are knowledgeable and nice persons, from whom I learned a lot.

This work is dedicated to my parents and elder brother, who love me the most in the world.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 The Problem

Denial of service (DoS) attacks (as shown in Fig. 1.1) — and more commonly observed, the distributed ones (DDoS) attacks (as shown in Fig. 1.2) — have become a major threat to the stability of the Internet. Since the several high-profile Distributed Denial of Service (DDoS) attacks launched on Yahoo! [9], eBay and Amazon.com in 2000, DDoS attack technologies have continued to evolve and continues to be used to impair Internet infrastructures [3][31].

The first obvious consequence of DDoS attack is service disruption, and not only a majority of Internet users are annoyed by the failures of Internet service access, but also the commercial entities on the Internet are suffering from the bad publicity and possible loss of customers [12]. A real-time online transaction may directly suffer money loss due to service disruption caused by a DDoS attack. Computer Economics [11] estimated that the total economic impact of Code Red was $2.6 billion, and Sircam cost another $1.3 billion. A recent attack via SQL Slammer caused an estimated $1 billion in damage during the first five days as it rapidly spread around the globe [36]. A 2003 CSI/FBI computer crime & security survey reveals 55% of respondents reported DoS attacks with $65 million loss. Even worse, DDoS attacks are quite simple to commit. Quite a lot of tools and malicious source codes are available on the Internet. The well-known ones include TFN2K, Stacheldraht and Trin00 as relatively old version. Therefore, dealing with DDoS is a major security concern of the Internet.

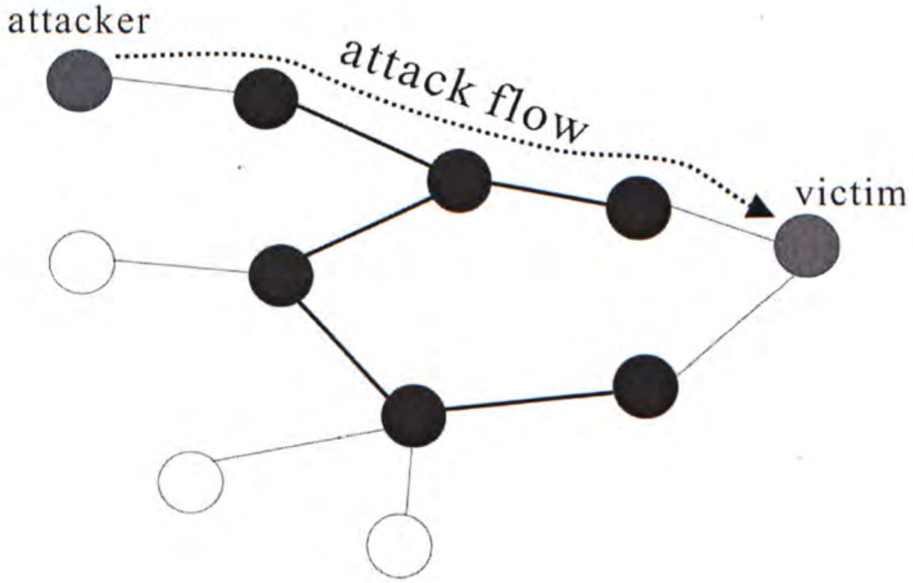


Figure 1.1: Denial of Service attack.

Although great efforts has been involved in attack detection and prevention, there is still a lack of effective and efficient solutions to intercept ongoing attack in a timely fashion.

However, because of the anonymous nature of Internet, the attackers can arbitrarily spoof the source IP addresses in the attack packets. The attacker will use a forged IP address in the malicious packets, which is addressed as *IP Spoofing*. First, this can avoid being traced

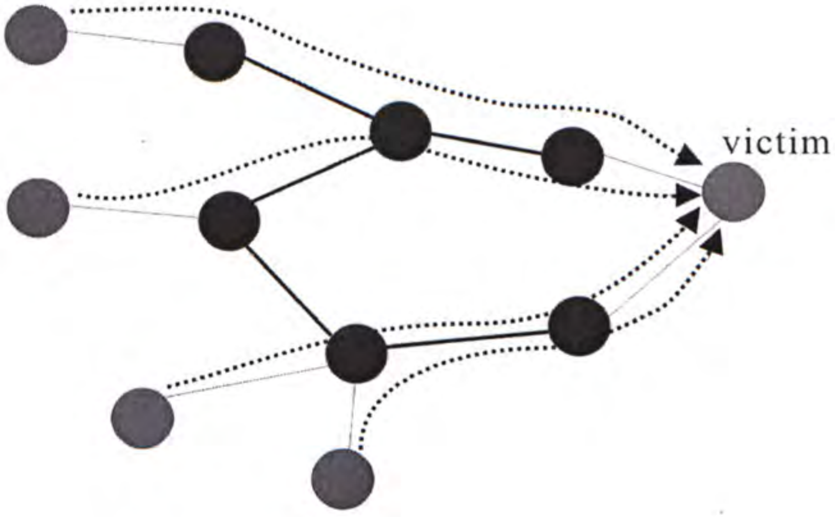


Figure 1.2: Distributed Denial of Service attack.

back and possible law punishment, and second, the source addresses being scattered across a large amount of different spoofed addresses makes some detection tools fail to identify the traffic anomalies. IP spoofing [16] is a quite simple technique prevailing in MITM (Man In the Middle) attacks, DoS attacks and other most network crimes. In such instances, conventional methods of determining the location of the system with a given IP address on the Internet (e.g., traceroute) no longer work because the source address used for tracing can be spoofed. Therefore, more advanced methods of identifying the source of attacking packets are needed.

1.2 Research Motivations and Objectives

The most primary practice against IP spoofing [26] is *IP traceback*, which aims to find the actual sources of the attack, where we define the *source* as the router directly connected to the system from which the flow of packets, constituting the attack, was initiated [20][16]. IP traceback approaches are for the destination under attack to reconstruct the full paths traversed by the attack packets, so as to identify the origin(s) of the attack traffic.

IP traceback approaches are extensively proposed in the literature. Savage et al. [45] proposed to let each router probabilistically inscribe a piece of partial path information in the IP header of the packet it forwards. In the phase of attack paths reconstruction, the pieces of traceback information extracted from received packets will be put together hop by hop from near the victim. In the ICMP message traceback scheme [19], Bellovin proposed to write the traceback information in a separate ICMP message. In Advanced and Authenticated Marking Schemes (AMS), Song and Perrig [49] proposed to hash the traceback message to fit in the relatively short marking field instead of fragmenting these router messages. They also proposed an authenticated marking scheme using a 5 bits Hash Message Authentication Code (HMAC) [34] appending on the marked message. However, it seems unpractical due to the daunting computation overhead.

All these solutions represent certain effectiveness on solving the IP traceback problem. However, they also show weakness in different ways or show certain incapability to trace back the attackers efficiently for large scale DDoS attacks. For instance, the proposed probabilistic packet marking schemes or ICMP traceback messages schemes rely on observing a high volume of spoofed traffic comprised of thousands or millions of packets, so the attacker can undermine the traceback by spreading the attack traffic across many attacking hosts (also referred to as agents, slaves, or reflectors in a reflector DDoS attack [21]), greatly increasing the amount of time required by the traceback scheme to gather sufficient packets to analyze. Therefore, an effective traceback scheme should use as few packets as possible to reveal an attack path.

In addition, some people are challenging the necessity of the full-path traceback solution [17]; identifying all the intermediate routers that the attack packets traversed, may be unattractive to the victims and ineffective for DoS (DDoS) countermeasures. First, the full-path traceback is as good as the address of an ingress point in terms of identifying the attacker. Second, each packet in a datagram network is individually routed so packets may take different routes even if their source and destination are identical. Third, the addressing within ISPs' networks is not necessarily understandable to the public since ISP may use private addressing plans within their own networks [17]. Therefore, we propose a domain based IP packet marking scheme to identify the intermediate domains instead of the individual routers, except the one serving as the attack source.

Therefore, one of this research's objectives is to extensively discuss how our IP traceback proposal addresses the problems with the existing solutions. The proposed packet marking scheme is intended to serve as an IP traceback scheme effective in the common performance metrics for IP Traceback schemes, namely, Minimum Number of Packets, Computation Overhead, Robustness and Deployment Overhead[35].

In addition to the packet marking scheme for IP traceback which is used to solve the IP traceback problem, we propose two alternative attack traffic regulation schemes to protect the victim under the DDoS attacks.

In general, IP traceback and attack source identification has limited capability of defending against DDoS attacks with regard to helping mitigation of DDoS attack traffic and guaranteeing the service availability [15]. First, current IP traceback solutions are not always able to trace packets' origins (e.g., those behind firewalls and NAT (network address translators). Second, even if the attack sources can be identified successfully, it is a rather difficult task to stop them at the sources, especially when they are scattered in various AS's. It is of no help on preserving the service availability by IP traceback techniques alone. While what concerns the network security community is how to defend against DDoS attacks, reducing the damages caused by such notorious attacks.

There are quite a number of measures that are used to either prevent the happening of DoS/DDoS attacks or mitigate the effect of it. They may need the modification of the Internet infrastructure or configuration of the network elements and some need to call for cooperation of network operators.

Ingress/Egress filtering [27] and DPF (route-based Distributed Packet Filtering) technique [41] are two distributed filtering techniques relying on the Internet-wide deployment. They are used to detect those messages with invalid sources and discard them. However, it is still a question whether this "global security" would be practical in the near future since they require global cooperation; furthermore, messages whose claimed sources are modified between adjacent routers in the middle of the network will not be detected by ingress filtering.

A second category of techniques is defenses right before victim; congestion control based mechanisms are such representatives. As most DDoS attacks incur network congestion while flooding the victims by large traffic, a congestion control mechanism should help alleviate the resources consumption suffered by legitimate traffic. Random Early Detect (RED) [28] is an approach to identify and preferentially drop the flows that do not obey TCP-friendly end-to-end congestion control. Some other works, such as Fair Queueing [25] and Class-Based Queueing [29] try to allocate certain available bandwidth to each flow so that they can be all served. The drawback with these methods is that packets or flows are unidentifiable concerning whether they are attack traffic or legitimate traffic.

J. Ioannidis and S. M. Bellovin proposed an approach of the Pushback mechanism [33]. The main drawback with the hop-by-hop tracing methods is that, in large-scale DDoS attacks, they have limited capabilities to separate the legitimate packets from attack packets in a pattern-based way. At the same time, some researchers propose that DDoS attacks should be stopped as close to the sources as possible. In the work of Jelena Mirkovioć et al. [1] a DDoS defense system called D-WARD is proposed, which is deployed at the source-end networks (stub networks or ISP networks) and prevents the machines from participating in DDoS attacks. However this approach requires that many routers at different network entry points each independently run the system, since each D-WARD router only polices data flows originating from its own network. Further more, the DDoS defense capability is confined by the limited attack traffic detection.

Therefore, the other objective is to remove the attack traffic to defend against the DDoS attacks; addressing the problems discussed above meanwhile. We would propose a method leveraging on the IP traceback, with attack sources identified.

1.3 The Rationale

Our proposed DDoS attacks defense mechanism involves first finding the attack sources and then using the attack signatures to perform packets filtering at the victim end or rate-limiting at the source end.

The IP traceback scheme keeps track of the intermediate domains instead of what routers a packet has traversed within a domain and identifies the source routers serving as ingress points of attack traffic at the same time. As the proposed marking scheme needs only a small number of packets to reconstruct the attach graph, it facilitates s a real time practice of stopping the on-going DDoS attacks, which is the focus of this work.

The packets filtering process take place on participating routers located around the potential victims (routers are supposed to have been equipped with tools for packet filtering. Netfilter, iptable are such available tools implemented due to different router OS). First, the IP Traceback mechanism is used to reconstruct the attack graph with the collected marked packets from the attacking sources or other sources whatsoever. The reconstructed attack graph is encoded as signature bitmaps, consisting of markings of detected source routers or domains on attack paths. The signature bitmaps would be sent to the filtering agents, and by inspecting each received packet, the filtering agents would catch an attack packet whose marking match with a filtering signature and drop it with certain probability.

However, we note that not all attack packets could be removed at the victim-end. And the network bandwidth is not protected since attack traffic still marches the whole way till near the victim. Therefore, we propose to let source routers identified by the underlying IP traceback scheme rate-limit the attack traffic from the sources. The traffic limiting rates are calculated such that the bandwidth of the victim is well utilized, thus the throughput of legitimate traffic is optimized.

1.4 Thesis Organization

This research examines in details a practical DDoS defense approach based on a packet marking scheme for IP traceback. Chapter 2 surveys the previous works on DDoS defense problem and addresses the problems with the existing solutions. Chapter 3 extensively presents the proposed IP traceback scheme and states about the effectiveness of this method, which is quite related to the effectiveness of the overall performance of the defenses scheme. In the fourth chapter, we would articulate the operation of the packet filtering mechanism and source-end rate-limiting system and present simulation results. Chapter 5 concludes our research work, listing the contributions of our work and discussing future work.

□ **End of chapter.**

Chapter 2

BACKGROUND STUDY

In this chapter, we present a detailed and formal definition of Distributed Denial of Service Attacks and IP traceback problem. We would also highlight the desired features of an IP traceback system [23]. we would survey the previous works on DDoS defense problem and discuss the problems with the existing solutions [57].

2.1 Distributed Denial of Service Attacks

A Denial of Service (DoS) attack is characterized by an explicit attempt by an attacker to prevent legitimate users of a service from using the desired resources [32]. Examples of denial of service attacks include:

- attempts to "flood" a network, thereby preventing legitimate network traffic
- attempts to disrupt connections between two machines, thereby preventing access to a service
- attempts to prevent a particular individual from accessing a ser-

vice

- attempts to disrupt service to a specific system or person.

In a distributed denial of service (DDOS) attack, the attacker compromises a number of daemons and installs flooding servers on them, later contacting the set of servers to combine their transmission power in an orchestrated flooding attack. The distributed format adds the "many to one" dimension that makes these attacks more difficult to prevent [37]. The use of a large number of daemons both augments the power of the attack and complicates defending against it: the dilution of locality in the flooding stream makes it more difficult for the victim to isolate the attack traffic in order to block it, and also undermines the potential effectiveness of traceback techniques for locating the source of streams of packets with spoofed source addresses.

A distributed denial of service attack is composed of four elements, as shown in Fig. 2.1.

- First, it involves a *victim*, i.e., the target host that has been chosen to receive the brunt of the attack.

- Second, it involves the presence of the attack *daemon agents*. These are agent programs that actually conduct the attack on the target victim. Attack daemons are usually deployed in host computers. These daemons affect both the target and the host computers.

- The third component of a distributed denial of service attack is the *control master program*. Its task is to coordinate the attack.

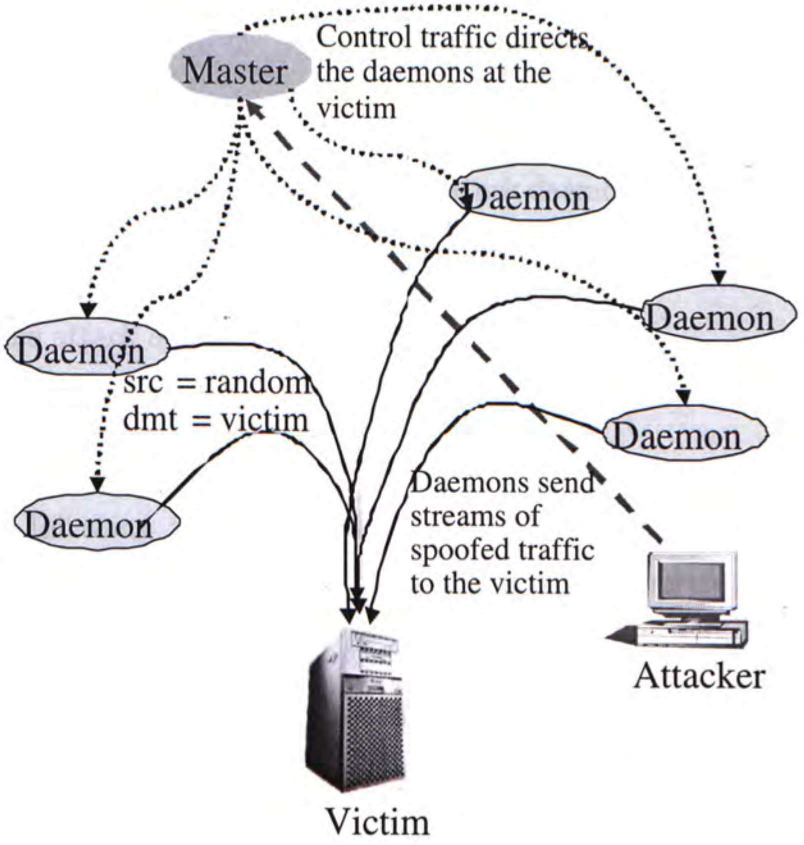

Master
Control traffic directs the daemons at the victim
Daemon
Daemon
Daemon
Daemon
src = random
dmt = victim
Daemon
Daemons send streams of spoofed traffic to the victim
Attacker
Victim

Figure 2.1: Architecture of a distributed denial-of-service (DDOS) attack.

- Finally, there is the *real attacker*, the mastermind behind the attack. By using a control master program, the real attacker can stay behind the scenes of the attack.

The following steps take place during a distributed DoS attack:

1. The real attacker sends an "execute" message to the control master program.
2. The control master program receives the "execute" message and propagates the command to the attack daemons under its control.
3. Upon receiving the attack command, the attack daemons begin the attack on the victim.

2.1.1 Taxonomy of DoS and DDoS Attacks

In the following text, we review some well-known attack methods: Smurf, SYN Flood, and User Datagram Protocol (UDP) Flood and the current distributed denial of service methods: Trinoo, Tribe Flood Network, Stacheldraht, Shaft and TFN2K.

Smurf Attack - When a perpetrator sends a large number of ICMP echo (ping) traffic at IP broadcast addresses, using a fake source address. The source address will be flooded with simultaneous replies (See CERT Advisory: CA-1998-01 [2]).

Smurf attack involves an attacker sending a large amount of Internet Control Message Protocol (ICMP) echo traffic to a set of Internet Protocol (IP) broadcast addresses. The ICMP echo packets are specified with a source address of the target victim (spoofed address). Most hosts on an IP network will accept ICMP echo requests and reply to them with an echo reply to the source address, in this case, the target victim. This multiplies the traffic by the number of responding hosts. On a broadcast network, there could potentially be hundreds of machines to reply to each ICMP packet. The process of using a network to elicit many responses to a single packet has been labeled as an "amplifier". There are two parties who are hurt by this type of attack: the intermediate broadcast devices (amplifiers) and the spoofed source address target (the victim). The victim is the target of a large amount of traffic that the amplifiers generate. This attack has the potential to overload an entire network.

User Datagram Protocol (UDP) Flood [1] - When a connection is established between two UDP services, each of which produces output, these two services can produce a very high number of packets that can lead to a denial of service on the machine(s) where the services are offered. Anyone with network connectivity can launch an attack; no account access is needed.

For example, by connecting a host's chargen service to the echo service on the same or another machine, all affected machines may be effectively taken out of service because of the excessively high number of packets produced. In addition, if two or more hosts are so connected, the intervening network may also become congested and deny service to all hosts whose traffic traverses that network.

SYN Attack - When an attacker sends a series of SYN requests to a target (victim). The target sends a SYN/ACK in response and waits for an ACK to come back to complete the session set up. Since the source address was fake, the response never comes, filling the victim's memory buffers so that it can no longer accept legitimate session requests.

SYN Flooding attack is also known as the Transmission Control Protocol (TCP) SYN attack, and is based on exploiting the standard TCP three-way handshake. The TCP three-way handshake requires a three-packet exchange to be performed before a client can officially use the service. A server, upon receiving an initial SYN (synchronize/start) request from a client, sends back a SYN/ACK (synchronize/acknowledge) packet and waits for the client to send the final ACK (acknowledge). However, it is possible to send a barrage of initial SYN's without sending the corresponding ACK's, essentially leaving the server waiting for the non-existent ACK's. Considering that the server only has a limited buffer queue for new connections, SYN Flood results in the server being unable to process other incoming connections as the queue gets overloaded.

In the following text, we describe the distributed denial of service methods employed by an attacker. These techniques help an attacker coordinate and execute the attack. These types of attacks plagued the Internet in February 2000. However, these distributed attack techniques still rely on the previously described attack methods to carry out the attacks. The techniques are listed in chronological order. It can be observed that as time has passed, the distributed techniques: Trinoo, TFN, Stacheldraht, Shaft, and TFN2K have become technically more advanced and, hence, more difficult to detect.

Trinoo uses TCP to communicate between the attacker and the control master program. The master program communicates with the attack daemons using UDP packets. Trinoo's attack daemons implement UDP Flood attacks against the target victim [4].

Tribe Flood Network (TFN) uses a command line interface to communicate between the attacker and the control master program. Communication between the control master and attack daemons is done via ICMP echo reply packets. TFN's attack daemons implement Smurf, SYN Flood, UDP Flood, and ICMP Flood attacks [6].

Stacheldraht(German term for "barbed wire") is based on the TFN attack. However, unlike TFN, Stacheldraht uses an encrypted TCP connection for communication between the attacker and master control program. Communication between the master control program and attack daemons is conducted using TCP and ICMP, and involves an automatic update technique for the attack daemons. The attack daemons for Stacheldraht implement Smurf, SYN Flood, UDP Flood, and ICMP Flood attacks [5].

Shaft is modeled after Trinoo. Communication between the control master program and attack daemons is achieved using UDP packets. The control master program and the attacker communicate via a simple TCP telnet connection. A distinctive feature of Shaft is the ability to switch control master servers and ports in real time, hence making detection by intrusion detection tools difficult [7].

TFN2K uses TCP, UDP, ICMP, or all three to communicate between the control master program and the attack daemons. Communication between the real attacker and control master is encrypted using a key-based CAST-256 algorithm [14]. In addition, TFN2K conducts covert exercises to hide itself from intrusion detection systems [8].

2.2 IP Traceback

Criminals have long employed the tactic of masking their true identity, from disguises to aliases. It should come as no surprise then, that criminals who conduct their nefarious activities on networks and computers should employ such techniques. Due to the anonymous nature of the IP protocol, *IP spoofing* is prevailing in most different form DDoS attacks. The network routing infrastructure is stateless and based largely on destination addresses; no entity in an IP network is officially responsible for ensuring the source address is correct. In IP spoofing, an attacker spoof the IP address, especially source IP address of his malicious message [54].

Relatively advanced spoofing beyond simple floods are used in very specific instances such as evasion and connection hijacking as well as network scanning and probes. The sequence and acknowledgement numbers are commonly manipulated in such instances, but we would focus on its presence in DDoS attacks in this work. Examining the IP header, two bytes among various other information about the packet contains the source and destination IP addresses. Using one of several tools, an attacker can easily modify these addresses — specifically the "source address" field. In order to prolong the effectiveness of the attack, they spoof source IP addresses to make tracing and stopping the DoS as difficult as possible. IP spoofing is almost always used in denial of service attacks. Since crackers are concerned only with consuming bandwidth and resources, they need not worry about properly completing handshakes and transactions. Rather, they wish to flood the victim with as many packets as possible in a short amount of time.

When multiple compromised hosts are participating in the attack, all sending spoofed traffic, it is very challenging to quickly block traffic.

Being unable to accurately identify the true source of an IP datagram if the source wishes to conceal it is the primary difficulty in dealing with such attacks. Therefore, we are interested in finding out the origins of the attack packets. The accountability of DDoS attacks also require the accurate identification of attacking sources.

The ability to trace back the source of a particular IP packet is what we want by IP Traceback [22]. We would define IP traceback by describing the assumptions an IP traceback system must carry and its model.

2.2.1 Assumptions

There are several important assumptions that a traceback system should make about a network and the traffic it carries [48].

- Packets may be addressed to more than one physical host. IP packets may contain a multicast or broadcast address as their destination.

- Duplicate packets may exist in the network. An attacker can inject multiple identical packets itself, possibly at multiple locations.

- Routers may be subverted, but not often. An attacker may gain access to routers along (or adjacent to) the path from attacker to victim by a variety of means.

- Attackers are aware they are being traced. A sophisticated attacker is aware of the characteristics of the network, including the possibility that the network is capable of tracing an attack.

- The routing behavior of the network may be unstable. Two packets sent by the same host to the same destination may traverse wildly different paths.
- The packet size should not grow as a result of tracing. It is a main cause of fragmentation on the Internet.
- End hosts may be resource constrained. We assume that an end host, and in particular the victim of an attack, may be resource poor and unable to maintain substantial additional administrative state regarding the routing state or the packets it has previously received.
- Traceback is an infrequent operation.

The first two assumptions are simply characteristics of the Internet Protocol. A tracing system must be prepared for a situation where there are multiple sources of the same (identical) packet, or a single source of multiple (also typically identical) packets. The next two assumptions speak to the capabilities of the attacker(s). The traceback system must not be confounded by a motivated attacker who subverts a router with the intent to subvert the tracing system. The instability of Internet routing is well known [42] and its implications for tracing are important. As a result, any system that seeks to determine origins using multipacket analysis techniques must be prepared to make sense of divergent path information. The final assumption that traceback queries are infrequent has important design implications. It implies queries can be handled by a router's control path, and need not be dealt with on the forwarding path at line speed. While there may be auditing tasks associated with packet forwarding to support traceback that must be executed while forwarding, the processing of the audit trails is infrequent with respect to their generation.

The above assumptions are largely borrowed from the article of Savage on IP traceback. One assumption related to deployment issue with IP traceback schemes we need to clarify is that an IP traceback scheme is viewed as effective if it works without problem for the networks that participate, which means, we do not make universal deployment assumption for IP traceback schemes. If a network (or router, in the context of router-based traceback) does not participate, obviously, whether the non-participating network is on the attack path can not be identified by the IP traceback schemes.

2.2.2 Problem Model and Performance Metrics

In order to remain consistent with the terminology in the literature, we will consider a packet of interest to be nefarious, and term it an *attack packet*; similarly, the destination of the packet is the *victim*. Ideally, a traceback system should be able to identify the source of any piece of data sent across the network. As with any other auditing system, a traceback system can only be effective in networks in which it has been deployed [43]. Hence, we consider the source of a packet to be one of the following:

- the ingress point to the traceback-enabled network;
- the actual host or network of origin;
- one or more compromised routers within the enabled network.

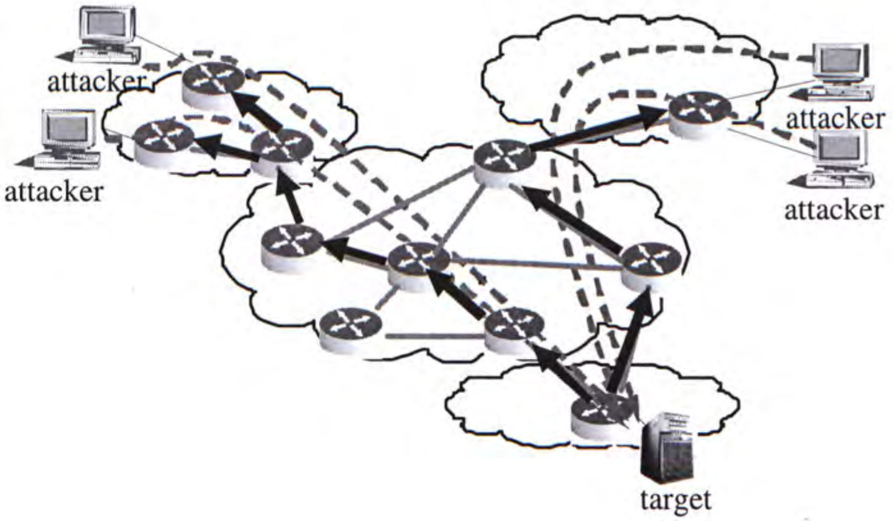


Figure 2.2: IP traceback.

In the literature, DDOS attacks are commonly considered as an attack propagating in a tree T, where the root of the tree T is the victim, V, each internal node in T corresponds to a router X on the Internet, and each leaf in T is an (possibly compromised) attack host [30]. Thus, T is a subtree of the Internet, where we are modeling only the inflow of packets to V. In fact, from the perspective of V, the tree T is a subtree of a much larger universal tree U that consists of the union of all routes to V in the Internet. For any internal node X in T, other than the root, we therefore sometimes refer to the parent of X as X's *downstream* neighbor. Likewise, the children of a node X in T are sometimes called X's *upstream* neighbors.

Our goal in the traceback problem is to identify the internal nodes of the tree T. That is, we wish to identify the internal nodes in the universal tree U that correspond to routers unwittingly serving in the attack tree T to send attack packets to the victim V. In addition, we specifically want to exclude from T any routers that are not part of the attack. Moreover, so as to traceback large-scale DDoS attacks, we desire solutions that allow for efficient traceback even if T contains hundreds of routers.

In a largely similar way, we would model the attack with a number of coordinated attackers attacking a single victim as an undirected graph with each node representing a *domain* (See Fig. 2.2). Domain is a logical subnet[1]on the Internet; a campus or internal corporate network is an example of a domain. Data exchange between campus and corporate domains is facilitated by one or more ISP domains, which offer, as a service, transmission and switching facilities for data exchange between their customers. The IP packets thus flow though different network domains, from regional ISP network to international ISP network and finally get to the destination. In general, to model the attack, a network domain can be thought of as a cloud, which connects to other domains at the peering points, with clients attaching on border routers. In our solution, the reconstructed attack graph would incorporate attack paths and the source router(s) identified, with each node on the paths can be viewed as a domain. Note that, Each internal node X in our attack model is a domain instead of a router, the leaf nodes still representing attack host (possibly compromised ones) though.

In the literature, the performance of IP traceback approaches is commonly measured by several parameters, namely, Minimum Number of Packets, Computation Overhead, Robustness and Deployment Overhead [35].

[1] Subnet is a portion of a network, which may be a physically independent network segment, which shares a network address with other portions of the network and is distinguished by a subnet number [55].

- *Minimum number of packets* is the number of packets required for attack graph reconstruction; it is desired to be minimized to achieve a fast response to an attack and diminish the damage [45][35].

- *Computation Overhead* represents estimated time required for reconstructing the path back to the attackers. An efficient traceback approach should feature a relatively low computation complexity and incremental deployment into the current Internet structure, at low cost [35].

- *Robustness* of an IP traceback method is evaluated by two characteristics, namely number of false positives and number of false negatives. A *false positive* is a router that is actually not on an attack path but is reconstructed to an attack graph by a traceback mechanism, and a *false negative* is a router that is missed in the reconstructed attack graph [45][35]. An IP traceback approach is said to be robust if it gives a relatively low rate of false positives and false negatives, and if the rate does not grow rapidly as the increase of number of attackers.

- *Deployment* - We need to employ a number of routers on the Internet to implement the traceback mechanism. This parameter can not be measured or calculated directly but can be evaluated by means of common sense.

2.3 IP Traceback Proposals

2.3.1 Probabilistic Packet Marking (PPM)

Probabilistic packet marking was originally introduced by Savage et al. [45]. In his approach (as known as Fragmentation Marking Scheme, or FMS), there are two phases namely marking procedure and reconstruction procedure. In the first phase, each router X performs, for each packet it processes, an information injection event that occurs with a set probability p (e.g., $p = 1/20$). The information injection involves using b bits in the IP header that are typically not used or changed by routers; 5 bits of this for a hop count, which helps their reconstruction algorithm. The remaining bits are used for message MX which is the identification information of that router. If that message is too big, they break it into fragments and use some bits of usable IP header to store a fragment offset and its data fragment.

Their algorithm is quite interesting, as it introduces the packet marking framework, and does not require a priori knowledge of the universal internet topology. But their algorithm, unfortunately, is not practical for large distributed denial-of-service attacks. In particular, their algorithm for reconstructing a message MX from a router at distance d from the victim requires n_d^l checksum tests, where n_d is the number of routers in T at distance d from V and l is the number of fragments messages have been divided into (in most cases, 8 fragments are required). For example, if $n_d = 30$ and $l = 8$, then the victim has to perform over 650 trillion checksum tests in order to reconstruct each of the 30 messages [33]. Such a computation is, of course, not feasible for the victim, and even if it were, it would introduce many false positives. Potentially, the victim would have to perform searches of data structure consisting of billions of entries.

The number of packets to be collected for reconstruction procedure is measured in thousands. In particular, because one IP address has to be fragmented and marked into a separate packet, they need 8 marked packets to represent one address pair, as proposed by Savage et al. In addition, due to the nature of probabilistic marking, the number of packets, X, required for the victim to reconstruct a path of length d has the following bounded expectation:

$$E(X) < \frac{ln(d)}{p(1-p)^{d-1}} \tag{2.1}$$

In particular, if $p = 10\%$, and the attack path has a length of 10, then a victim can typically reconstruct this path after receiving 75 packets from the attacker [45]. Moreover, this scheme is easily spoofed by an adversary that knows this algorithm.

In Advanced and Authentication Marking Scheme (AMS), Song and Perrig [49] improve the performance of probabilistic packet marking and suggest the use of hash chains for authenticating routers. They also use a 5-bit distance field, but they do not fragment router messages. Instead, they assume the victim knows the universal tree U, so the full IP address is encoded into 11 bits hash values by two sets of universal random hash functions in the packet marking. To reconstruct the attacking graph, the victim uses the upstream router map as a road-map and performs a breadth-first search from the root to identify the corresponding router which was encoded in the marking fields.

Though assumption has to be made that the victim is aware of the network topology, the number of packets required for reconstruction can be decreased to 1000 packets. They also proposed an authentication marking scheme using a 5 bits Hash Message Authentication Code (HMAC) appending on the marked message. But the computation for HMAC seems not very practical, though the evasion of the scheme becomes difficult for the attackers. Why AMS is advantageous over FMS lies in that without the combinatorial computations of checksum tests, AMS can successfully deal with DDoS attack up to near thousand attackers. With threshold $m > 7$ (where m is the number of hash functions are used in their scheme), AMS can be robust against DDoS with even 1500 distributed attacker sites and only has 20 false positives when there are 2000 attacker sites. But within a reasonable region, where the victim is satisfied by the furthest approximate sources it can reach, this scheme works with great accuracy and efficiency.

2.3.2 ICMP Traceback Messaging

An alternative approach, based on ICMP messaging, is to have each router R decide, with some probability q (typically, $q = 1/20000$ is mentioned), for each packet P to send an additional ICMP message packet to the destination. The message may contain several fields: Back link, which is information on the previous hop; Forward link, information on the next hop and Timestamp etc. [19]. So with enough Traceback messages from enough routers along the path, the traffic source and path can be determined.

The main drawback of this approach is that it causes additional network traffic even when no DDOS is present. Even so, it is not efficient, for identifying all the n internal nodes in the attack tree T

requires, a large even unacceptable number of attacking packets are required for the path reconstruction. For example, if the longest path length is 20 hops and there are about 1000 nodes on the reconstructed attack tree, the expected number of attack packets needing to arrive at the victim V before V will have sufficient information to reconstruct T is 7.5 million [30].

Processing overhead is incurred at the destination during reconstruction. Potentially, the victim has to perform searches of data structures consisting of thousands of entries. Reconstruction data structures will also require a large amount of memory.

Bandwidth overhead for this scheme is minimal, and will be about 0.005% derived from the fact that about 1 in every 20,000 packets will be marked[18]. However, the way the scheme is described, there is nothing to prevent an attacker from generating fake iTraces. DDoS attacks involve a massive amount of traffic from many different sources; plausible-looking fake chains could easily deceive a victim according to [19]. If a router that marks the packets becomes subverted, it can be reconfigured to generate incorrect iTraces, resulting in an incorrectly reconstructed path[18].

2.3.3 Logging

In a logging solution, we either ask routers to log the packets they process or we augment the data packets themselves to contain a full log of all the routers they have encountered on their way to their destinations [33]. With any of these solutions, a victim queries router or their storage agents to see whether they sent suspect attack packets. The drawback with these approaches is that they require additional storage at the routers. They also require a way of publishing the data stored at routers in a timely manner.

An approach belongs to the latter kind of solution is called Source Path Isolation Engine (SPIE) enabling the ability to identify the source of a particular IP packet given a copy of the packet to be traced, its destination, and an approximate time of receipt. One of SPIE's key innovations is to reduce the memory requirement (down to 0.5% of link bandwidth per unit time) [48]. Bloom Filters and storing only digests instead of full messages is the key for link bandwidth reduction. As packets traverse the network, digests of the packets get stored in the an module that they call it DGA. In this scheme, constant fields from the IP header and the first 8 bytes of the payload of each packet are hashed by several hash functions to produce several digests. Digests are stored in a space-efficient data structure that is called Bloom Filter, which reduces storage requirements by several orders of magnitude. When a given bloom filter is about 70 percent full, it is archived for later querying, and another one is used. The duration of using a single bloom filter is called a time period. Hash functions also change for different time periods [18]. SPIE [50] [51] can support traceback of large packet flows for longer periods of time in a fashion similar to probabilistic marking schemes — rather than discard packet digests as they expire, discard them probabilistically as they age.

The most pressing challenges for SPIE are increasing the window of time in which a packet may be successfully traced and reducing the amount of information that must be stored for transformation handling [48].

2.3.4 Tracing Hop-by-hop

In some cases, such as in reflector-based DDOS attacks we can use patterns in the attack packets to filter out DDOS packets at a firewall. Likewise, the approach of hop-by-hop tracing, which is also known as link testing, uses a pattern-based approach to do traceback of a DOS attack while it is in progress.

The approach of the automated Pushback mechanism [33], for example, is the solution currently supported manually by many router manufacturers. In this approach, a network administrator logs into the routers nearest the victim, and using statistics and pattern analysis, determines the next upstream routers in the attack tree T. The approach is then repeated at the upstream routers for as long as the attack continues. However, while under DDoS attack, controlling messages will proliferate in a fan-shape manner and cause resource usage to grow exponentially [56].

This scheme also requires immediate action during the attack, and requires considerable coordination between network administrators (to either communicate directly or setup access points for the agents of partnering administrators). This technique also requires some pattern-based way to separate legitimate packets from attack packets. Because of its iterative nature, this approach has limited traceback capabilities in a large-scale DDOS. In addition, it has limited forensic appeal, since it can only be used while the attack is occurring.

2.3.5 Controlled Flooding

A similar approach is used by Burch and Cheswick [20] to perform traceback by iteratively flooding from certain portions of the Internet to see its effects on this portion's incoming traffic. By carefully measuring incoming traffic to the attacked system and loading the links of the suspected path even more, a drop in the rates of the attack packets should be observed. The process can be repeated for the next hop and so on until the source of the attack is identified.

However, It relies on the fact that during DoS attacks the links of the attack path should be heavily loaded. This assumption may not hold for modern backbone networks with abundant bandwidth available on the links[18]. The number of false positives and false negatives are not discussed in the literature, but obviously it would become a problem when the suspected attacking network is relatively large. The number of packets to be used as flooding traffic for the scheme to successfully complete a traceback is considerable, which is also the cause of extremely high bandwidth overhead. It has the same drawback as the pushback method that it is incapable in very large scale DDoS attacks and it is only used while the attack is occurring.

2.4 DDoS Attack Countermeasures

There are many kinds of measures that can be used to moderate the effect and reduce the likelihood of DoS/DDoS attacks, including special configuration of network elements, improvement of Internet infrastructures and connection establishments, firewall installation, and active monitoring for packet filtering. However, few measures are effective under DDoS attacks, and most are even vulnerable to DoS attacks themselves, which is why none of them can be used extensively today.

There are some basic defensive mechanisms to augment the strength of a system, or increase the obstruction of offense.

1. The basic principle in decreasing the likelihood of being attacked is to reduce the number of targets that can be attacked. Thus, stopping all unnecessary or non-essential system services or network ports can be very helpful.

2. To increase the difficulty of TCP-SYN attacks, significantly enlarging the length of backlog queue or reducing the timeout period can help to cope with more simultaneous half-open connections.

This is the simplest solution for defending against DDoS attacks. However, the reconfiguration may delay, or even deny, legitimate access as well, or lead to a potential increase in resource usage [56].

Installing firewall can moderate the effect of TCP SYN attacks. To block the arrival of potentially malicious TCP connection request packets at the destination hosts, a firewall can work as a TCP connection request proxy that answers the request on its behalf, or as a TCP connection request monitor that will send a third message to the destination host to release its resources [46]. However, the firewall also introduces new problems. It will cause new delays for every connection, including those for legitimate users. This method assumes the firewall is not vulnerable to TCP SYN flooding, because it becomes the new target of DoS attacks. Nevertheless, even a specialized firewall can be disabled by a flood of 14000 packets per second, according to Moor's research [56].

Another category of techniques is to prevent the exhaustion of the victim's resources is to limit the resource allocation and usage for each user or service. Congestion control based mechanisms are such representatives. As most DDoS attacks incur network congestion while flooding the victims by large traffic, a congestion control mechanism should help alleviate the resources consumption suffered by normal traffic. Random Early Detect (RED) [28] is an approach to identify and preferentially drop the flows that do not obey TCP-friendly end-to-end congestion control. Some other works, such as Fair Queueing [25] and Class-Based Queueing (CBQ) [29] try to allocate certain available bandwidth to each flow so that they can be all served. Class Based Queuing configures different traffic priority queues and rules that determine which packets should be put into which queue. The drawback with these methods is that packets or flows are unidentifiable concerning whether they are attack traffic or normal traffic.

A great deal of research focuses on how to continuously monitor TCP/IP traffic in a network, looking for irregularity in packet behavior. If any of the attack symptoms appear, the monitor agents will react appropriately to moderate the effect of the attack and maintain services for legitimate users. The differences between these methods concern how to determine whether or not a packet is malicious. Once determined, monitor agents will discard suspected packets or reset pending connection requests [56]. The proposals to be examined in the following text should be able to represent a wide variety of different techniques on how to determine whether or not a packet is malicious.

2.4.1 Ingress/Egress Filtering

Ingress/Egress filtering [27] at every ISP is recommended to prevent DoS attacks using IP addresses spoofed packets. This approach is to implement ingress and egress filtering on border routers at each ISP; by checking the consistency of the claimed source in each received message with the valid range of sources from which the router could receive the message, it detects messages whose claimed sources are modified (spoofed) and discards them. Additionally, the implemented ACL (access control list) at downstream interface should not accept addresses with internal range as the source, as this is a common spoofing technique used to circumvent firewalls. On the upstream interface, source addresses outside of your valid range should be restricted, which will prevent attackers within the particular ISP network from sending spoofed traffic to the Internet.

However, Ingress/Egress filtering is effective only if used in large-scale applications. In general, routers in large backbone networks with complex topology cannot make a clear distinction between inbound and outbound traffic. Furthermore, mobile IP situations can be created in which legitimate addresses appear at an apparently wrong interface. It is still a question whether this "global security" would be practical in the near future since they require global cooperation; furthermore, messages whose claimed sources are modified between adjacent routers in the middle of the network will not be detected by ingress filtering.

2.4.2 Route-based Distributed Packet Filtering (DPF)

A route-based distributed packet filtering (DPF) technique shares some similarity with Ingress filtering method by installing packet filters at certain amount of autonomous systems distributed in the Internet. The two techniques are basically distributed filtering systems relying on the Internet throughout deployment. At an extreme, with all autonomous systems and their routers implementing route-based packet filtering DPF method is no much different from the effect of perfect ingress filtering, with no spoofed IP flows being able to escape.

In this approach [41], Park et al. proposes to use routing information to determine if a packet arriving at a router, e.g., border router at an AS, is valid with respect to its claimed source/destination addresses, since a source and destination address pair should follow the reachability constraints imposed by routing and network topology.

The main merit of this method as shown by their performance evaluation is that, with a partial coverage on at 18% of such filters equipped autonomous systems, significant amount of spoofed packets can already be dropped and the origins of those spoofed IP flows succeed reaching their targets can be localized to within a small, constant number of sites (less than 5 for Internet AS topologies). However, the cooperation of thousands of autonomous systems is still an unfavorable requirement, with every ingress/egress router of which has to install the filter. In addition, the effectiveness depends intimately on the connectivity structure of the underlying AS graph. Another problem with this method is that computing appropriate filtering tables alongside existing interdomain routing protocols (e.g., BGP) is a nontrivial problem due to the destination-based structure of Internet routing protocols.

2.4.3 IP Traceback Based Intelligent Packet Filtering

Minho Sung and Jun Xu proposed a novel technique [53] for defending against Internet DDoS Attacks based on knowledge of the reconstructed attack paths from some underlying IP traceback scheme (we hereafter refer to it as IP traceback based packet filtering method). A newly designed mark format makes it possible to identify a suspected packet, coming from an "infected" edge (the network edges on the attack graph) and these packets are preferentially filtered out since they are more possibly from an attacker.

They use some packet marking IP traceback scheme to reconstruct the attack graph, and call the network edges on the attack graph "infected" edges. Consequently, packets marked with the "infected" edge markings will be preferentially filtered out, while packets from a normal client, on the other hand, with high probability will not be filtered out, since typically most of the edges on the normal path to the victim are not infected. They modified the underlying traceback scheme a bit for effective filtering effect. The marking performed by each router is divided as two types: signaling is for traceback purpose as the underlying IP trace-back scheme requires (around 5% of total marks); the other is for marking packets with edge information so as to serve as the filtering signatures (as proposed at 95%). A defense-line with implementation of filters around the victim will impose rate-limit on incoming traffic flows based on the inscribed marks in each packet.

However, to make sure significant amount of traffic can be filtered based on the second type of marks, only 5% of the marks are signaling marks. This implies that the reconstruction of the whole attack graph will be 20 times slower than in the underlying IP traceback scheme, which is a considerable delay when considering the relatively short lifetime of most DDoS attacks. In addition, as the attackers become more distributed, the normal paths between normal clients and attack paths have high probability to overlap, which means an "infected" edge would also be traversed by a non-attack packet, especially when they near the victim. Anyway, this method presents an appraisable idea to integrate the packet filtering mechanism with IP traceback scheme.

2.4.4 Source-end DDoS Attack Recognition and Defense

At the same time, some researchers propose that DDoS attacks should be stopped as close to the sources as possible. In the work of Jelena Mirković et al. [39], a DDoS defense system called D-WARD (DDoS Network Attack Recognition and Defense) is proposed, which is deployed at the source networks routers (stub networks or ISP networks) and prevents the machines from participating in DDoS attacks. As its name indicates, D-WARD autonomously detects and suppresses DDoS flows originating at this network. This system observes the outgoing and incoming traffic and gathers lightweight statistics on the flows, classified by destination. These statistics, along with built-in traffic models, define legitimate traffic patterns. Any discrepancy between observed traffic and a legitimate traffic pattern for a given destination is considered to be the signal of a potential DDoS attack. The source router then decides to throttle all traffic to the suspected target of the attack and at the same time attempts to separate attacking flows from legitimate flows and identify the attacking machines. In short, D-WARD monitors two-way traffic between a set of "policed" address set and the rest of the Internet and compares the online traffic statistics to predefined models of normal traffic, and non-complying flows are rate-limited.

This approach has the benefit of preventing malicious flows from entering the network and consuming resources. They also plan to investigate the possibility of also deploying this system on the core routers in the future.

According to their measurement for false positives, they report a low level of false positives, less than 0.5%. While they use the the number of flow and connection misclassifications (the number of times that any flow was misclassified as attack or suspicious, and the number of times that any connection was misclassified as bad) as false positives. Since they make use of a model of normal traffic or system behavior, there are two issues has to be addressed [40]:

1. Threshold setting. Anomalies are detected when the current system state differs from the model by a certain threshold. The setting of a low threshold leads to many false positives, while a high threshold reduces the sensitivity of the detection mechanism.

2. Model update. Systems and communication patterns evolve with time, and models need to be updated to reflect this change. Anomaly-based systems usually perform automatic model update using statistics gathered at a time when no attack was detected.

Furthermore, the DDoS defense capability is confined by the limited attack traffic detection.

Table 2.1: Classification by deployment location.

	Single node	**Multiple collaborating nodes**
Victim end		
Victim's network	IP traceback based packet filtering	
Intermediate network		Pushback Ingress/Egress filtering Route-based DPF
Source end network		D-WARD

2.4.5 Classification of DDoS Defense Methods

The reviewed methods can be implemented on a single node or on multiple collaborating nodes that can be part of the victim, the victim's network, or the intermediate network. Table 2.2 shows the defense strategies according to the region they take actions.

Only the IP traceback based packet filtering method falls into the category of taking act on a single node. However, this refers to the filtering procedure without talking about the other part of the method — packet marking for IP traceback. This method hereby also requires deployment over the Internet. The other methods involve the participation of the intermediate networks more or less.

Based on the response strategy of DDoS defense mechanisms, we differentiate between *preventive* and *reactive* mechanisms.

- The goal of preventive mechanisms is either to eliminate the possibility of DDoS attacks altogether or to enable potential victims to endure the attack without denying services to legitimate clients.

- Reactive mechanisms strive to alleviate the impact of an attack on the victim. In order to attain this goal they need to detect the attack and respond to it.

Two basic methods to relieve the impact of DDoS attacks are packet filtering and rate-limiting.

- Filtering mechanisms use the characterization provided by a detection mechanism to filter out the attack stream completely. Unless detection strategy is very reliable, filtering mechanisms run the risk of accidentally denying service to legitimate traffic.
- Rate-limiting mechanisms impose a rate limit on a stream that has been characterized as malicious by the detection mechanism. Rate limiting is a lenient response technique that is usually deployed when the detection mechanism has a high level of false positives or cannot precisely characterize the attack stream. The disadvantage is that they allow some attack traffic through, so extremely high scale attacks might still be effective even if all traffic streams are rate-limited.

The packet filtering and rate-limiting mechanisms could address a specific aspect of solving the problem. While as is frequently pointed out, there is no "silver bullet" against DDoS attacks. In our work, we would defend against DDoS attacks by applying both approaches flexibly to solve the problem more effectively.

□ **End of chapter.**

Table 2.2: Classification by response strategy.

	Preventive mechanisms	**Reactive mechanism**
Filtering mechanism	Ingress/Egress filtering Route-based DPF	IP traceback based packet filtering
Rate-limiting mechanism		D-WARD Pushback

Chapter 3

ADAPTIVE PACKET MARKING SCHEME

3.1 Scheme Overview

In this chapter, we would present a practical IP traceback approach. It addresses the issues concerned by both the victims and network operators such as per-packet marking space limitation, network overload and computation overhead.

flag = 1	dist	edge
1 bit	5 bits	16 bits

(a) *did* marking field

flag = 0	cord	rid
1 bit	14 bits	10 bits

(b) *rid* marking field

Figure 3.1: Marking fields.

In our proposed marking algorithms, we employ 25 bits space in the IPv4 packet header as marking fields. Probabilistically, each participating router adaptively inscribes onto a traversing packet with its local partial path information. There are two types of markings: router identification (*rid*) marking and domain identification (*did*) marking. The *rid* marking is executed if the packet enters the network for the first time; in contrast, *did* marking is performed when the packet traverses along the following domains towards the victim. Fig.3.1 shows the marking fields for domain *id* marking and router *id* marking respectively.

Two types of markings, either the router *id* marking or the domain *id* marking, are to be performed by a router adaptively by checking whether the router concerned is the ingress point of the to-be-marked packet or not.

At first, however, the border routers, with implementation of our marking scheme should be capable of determining which type of marking to perform. Physically, these routers are connected to end-hosts or other routers through different interfaces; a router therefore checks through which interface it receives a packet to see whether a packet is forwarded by another router from outside the domain concerned or sent by an end-host at the customer side.

The domain *id* marking would be performed if the router receives a packet routed from outside the current domain, and the router *id* marking would be performed when the packet comes from an end-host. Fig. 3.2 depicts the basic logic flow of the marking procedure consisting of two types of markings.

The victim under a DDoS attack reconstructs the attack graph in two phases. First, it identifies all the intermediate domains taking part in forwarding the attack packets, and recovers the inter-domain attack

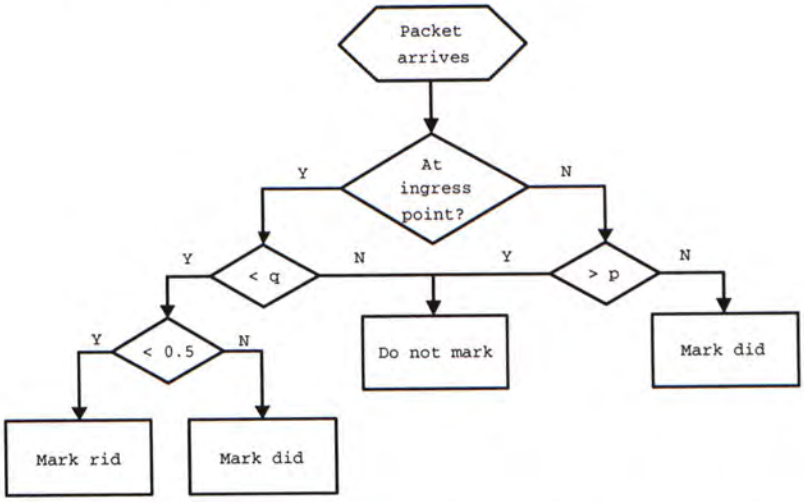


Figure 3.2: Marking algorithm flow chart with two types of markings.

paths by inspecting the domain *ids* marked in the received packets. Second, from the router *id* markings in the received packets, the victim can identify the source routers. In general, the inter-domain attack paths reconstruction leads to the identification of the *source domains* (where the source router may reside), and then the identified source domains are associated with the router *id* markings to uncover the source routers as we wish.

This work presents a novel design of probabilistic packet marking scheme at the granularity of domain, while at the same time, the attack sources can be identified. And this inter-domain IP traceback design has been proved to possess the following advantages. First, we use much less number of packets to identify the attack source(s). In particular, it requires only two uniquely marked attack packets to identify a source router. Second, this approach generates quite low false rates which will be demonstrated by the simulation results. Third, our traceback mechanism ensures incremental deployment and requires fewer routers to participate. Actually, only the border routers need to take part in this traceback mechanism.

Furthermore, keeping track of the domains traversed instead of intermediate routers on the attack paths and using *ids* instead of full 32-bit IP addresses have some underlying advantages. First, a domain is an administrative unit on the Internet, which has the capability to conduct defenses against the attacks when it is identified to be involved in an attack and notified by the victim. Second, it still makes sense to identify the source domains even the individual source routers are not identified correctly. We observe that systems of some domains are more likely to be compromised (to launch a DDoS attack, attackers usually compromise a bunch of vulnerable systems as attack agents) due to the domains' poor security features such as weak intrusion defense mechanisms and flawed security policies.

3.2 Adaptive Packet Marking Scheme

3.2.1 Design Motivation

Our adaptive packet marking scheme is based on the probabilistic packet marking technique, but a novel IP packet marking scheme is proposed, which is motivated by the below issues.

The IP traceback approaches, such as iTrace or the proposed probabilistic packet marking schemes, rely on observing a high volume of spoofed traffic comprised of thousands or millions of packets, so the attacker can undermine the traceback by spreading the attack traffic across many attacking hosts (also referred to as agents, slaves, or reflectors in a reflector DDoS attack [21]), greatly increasing the amount of time required by the traceback scheme to gather sufficient packets to analyze. Therefore, an effective traceback scheme should use as few packets as possible to reveal an attack path. Using a relatively short *id* instead of a full IP address, we do not need to spread a mark across multiple packets, and we thus feature a relatively small number of packets to fulfill the traceback.

In addition, some people are challenging the necessity of the full-path traceback solution [18]; identifying all the intermediate routers that the attack packets traversed, may be unattractive to the victims and ineffective for DoS (DDoS) countermeasures.

- First, the full-path traceback is as good as the address of an ingress point in terms of identifying the attacker.

- Second, each packet in a datagram network is individually routed so packets may take different routes even if their source and destination are identical.

- Third, the addressing within ISPs' networks is not necessarily understandable to the public since ISP may use private addressing plans within their own networks [18].

Therefore, we propose a domain based IP packet marking scheme to identify the intermediate domains instead of the individual routers, except the one serving as the attack source.

The proposed marking scheme overloads 25 bits space in IPv4 header; the 25 bits space consists of the 16-bit Fragment Identification field, 1-bit fragmentation flag and 8-bit Type of Service (ToS) field. Employing the 25 bits in the IP header for marking was first advocated by

Dean et al. [24]. The issue of overloading certain bits in the IP header would be discussed in a later section.

As every host or router on the Internet is identified using a 32-bit IP address [55], it is a challenging issue to overload the 25-bit marking space in the IP header with a 32-bit IP address. In our proposal, we aim at identifying the attack paths formed by the domains and the source routers. To specify a domain, we do not need to use 32-bit IP addresses, as long as we can uniquely identify each domain with a different identification. The 16-bit Autonomous System Number (ASN) is available for this purpose; it enables us to uniquely identify up to 2^{16} (65536) domains. Meanwhile, for each domain, we assign each border router with a 10-bit *rid*; this 10-bit space should be sufficient to encode up to 2^{10} (1024) border routers within a domain.

However, to defend against the attack, the victim may demand to block the malicious traffic at the source routers, so the victim needs to retrieve the IP addresses from the *ids*. To obtain a border router's IP address based on a 10-bit *rid*, it takes just one table look-up. This could be implemented as an ID-to-IP mapping table published on web sites, or it could be maintained at the victims individually. In practice, the *rid* assignment and retrieval would involve a management overhead; however, the amount of overhead is considered acceptable for the deployment of any marking scheme.

3.2.2 Marking Algorithm Basics

Our proposed marking scheme exploits the idea of probabilistic edge sampling [45], to mark the packets with a low probability in order to reduce the marking overhead of the routers. Probabilistic edge sam-

```
Marking procedure at router R:
for each packet $w$
    let x be a random number from [0..1)
    if $x < p$ then
        write R into w.start and 0 into w.distance
    else
        if w.distance = 0 then
            write R into w.end
        increment w.distance


Path reconstruction procedure at victim V:
let G be a tree with root V
let edges in G be tuples (start, end, distance)
for each packet $w$ from attacker
    if w.distance = 0 then
        insert edge (w.start, v, 0) into G
    else
        insert edge (w.start, w.end, w.distance) into G
remove any edge (x, y, d) with d $\neq$ distance from x to V in G
extract path ($R_i..R_j$ ) by enumerating acyclic paths in G
```

Figure 3.3: Edge sampling algorithm.

pling is a technique commonly used by most of packet marking based IP traceback methods. Its main idea is to let routers probabilistically mark packets with partial path information during packet forwarding. The path information includes the IP addresses of two adjacent routers in an Internet map, and the distance between the victim and the router last marked the packet.

Fig. 3.3 outlines the algorithm for edge sample with probabilistic packet marking.

However, the edge sampling algorithm requires 72 bits of space in every IP packet (two 32-bit IP addresses and 8 bits for distance to represent the theoretical maximum number of hops allowed using IP). There would not be sufficient space to append this much additional data to a packet in flight. Therefore, Savage et al. [45] proposed to encode each edge in half the space by representing it as the exclusive-or (XOR) of the two IP addresses making up the edge, as depicted in Fig. 3.4. This is a basic technique to reduce the storage requirement is storing edges instead of endpoints.

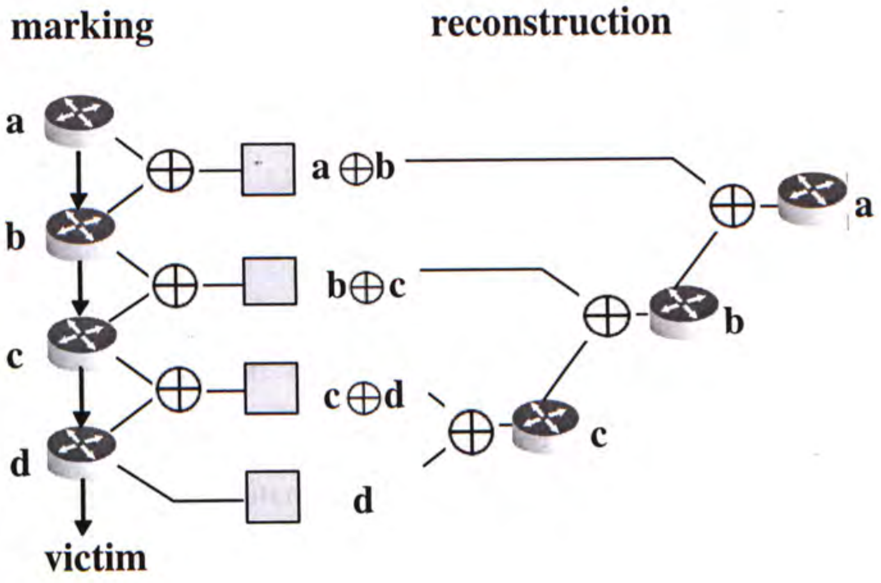


Figure 3.4: XOR encoding technique.

Each packet record the xor of two adjacent endpoints, thus the required space for two endpoints are reduced by half. When the packet with the xor value arrives at the victim, the victim need to decode the two endpoints back. The first node to be decoded is the node 1 hop away from the victim. As we know the IP address of victim, a, and one of the packet has the mark: a ⊕ b (suppose b is the first node upstream the victim, a), the IP address of b could be decoded through b = (a $\oplus$ b) $\oplus$ a. And all the nodes can be decoded in this way hop by hop.

3.2.3 Domain id Marking

The domain *id* marking algorithm allows the victim to infer the inter-domain attack paths by inspecting the domain *ids* in the received packets. A so-called domain *id* (*did*) is to be marked along with some other functional fields into the packets according to the probabilistic edge sample algorithm described above.

As shown by Fig. 3.1(a), there are three fields in the marking format: 16 bits "edge" field, 5 bits "dist" field and 1 bit "flag". The "edge" field stores one edge, made up by two endpoints encoded by XOR. These two endpoints may not necessarily be two routers connecting to each other. They are most likely two border routers that belong to two domains with connection between the domains instead of the individual routers. The "dist" field represents the number of hops traversed since the edge it contains is sampled. A flag is used to indicate whether this is a domain *id* marking or a router *id* marking.

Basically, the domain *ids* of two neighboring domains are encoded by exclusive-or (XOR) to make up the edge, and it can be decoded back during reconstruction in virtue of XOR's property that $a \bigoplus b \bigoplus a = b$. This XOR encoding technique, used to reduce per-packet storage requirement is depicted in Fig. 3.4.

Fig. 3.5 shows the domain *id* marking scheme. Marking probability p determines whether to mark a packet or not. To mark the packet, router R sets the distance to be zero and writes the domain *id* into

Algorithm Domain *id* marking by border router R

```
for each packet from an upstream domain do
    generate a random number $x$ within [0..1]
    if $x < p$ then
        pkt.edge = did
        pkt.dist = 0
    else
        if pkt.dist is 0 then
            pkt.edge = pkt.edge $\bigoplus$ did
        increment pkt.dist
```

Figure 3.5: Domain id marking algorithm.

the edge field. Otherwise, if the distance is zero, router R overwrites the edge field with XOR of edge value present in the to-be-marked packet and its own *id*. Therefore, forming an edge between itself and the previous router. Whenever the packet is decided to be marked, its distance value would be incremented by 1. Repeatedly, this procedure takes place for the following domains as the packets traverse along the path.

Note that the distance field is used as hop counts, and is always incremented. The compulsory incrementing is critical to minimize the spoofing of the markings by an attacker. Any packets written by the attacker will necessarily have a distance greater or equal to the length of the true attack path. Therefore, a single attacker is unable to forge any edge between themselves and the victim (for a distributed attack, of course, this applies only to the closest attacker) and the victim does not have to worry about "chaff" while reconstructing the valid suffix of the attack path [45]. We also remark that incremental deployment is ensured, because we can identify a domain even if only one border router within that domain sees the attack packet and marks it by our marking scheme.

3.2.4 Router id Marking

The router *id* marking algorithm is used to identify the source routers that serve as ingress points of attack packets. A router performs router *id* marking with certain marking probability if it receives a packet from the customer side.

As shown by Fig. 3.1(b), there are three fields in the marking format: 10 bits "rid" field, 14 bits "cord" field and 1 bit "flag". The "rid" field stores the *rid* if the packet is marked. The "cord" field is a 14 bits values computed from the *did* of the domain that the packets belong to. A flag is used to indicate whether this is a domain *id* marking or a router *id* marking.

Fig. 3.6 outlines the algorithm for router *id* marking by router R. A packet is either marked with *rid* marking of R (with higher probability) or *did* marking of the domain containing router R. The marking probability is possibly different from the one used for *did* marking, p; we denote it as q. When a packet is issued into the network for the first time, the router R decides whether to mark it or not based on the probability q.

To complete the inter-domain attack path, the source domain *id* should also be conveyed to the victim (Recall that we refer to the domain where a source router resides as source domain.); so we make it equally likely to mark a packet with the router *id* or the domain *id* at the source router. In practice, we use a larger marking probability

Algorithm Router id marking by router R

```
for each packet $pkt$ passing through R do
    generate a random number $x$ within [0..1]
    generate a random number $r$ within [0..1]
    if $x < q$ then
        if $r < 0.5$ then
            pkt.edge = $did$
            pkt.dist = 0
            pkt.flag = 1
        else
            pkt.rid = $rid$
            pkt.cord = hash($did$)
            pkt.flag = 0
```

Figure 3.6: Router id marking algorithm.

q in router *id* marking procedure, which is double of the probability p that we use in domain *id* marking. It is like flipping a coin to decide to mark a packet with domain *id* or router *id* at the entry point. To mark a packet with a domain *id*, we set flag to be one and write the domain *id* into the edge field; while to mark the packet with a router *id*, we set flag to be zero and write the 10-bit router *id* into the *rid* field. The *cord* is calculated and known beforehand and is marked into the packet along with the *rid*.

As a 10-bit router id is identifiable only within a domain; only by associating the *rid* with its corresponding source domain *id*, can we uniquely identify the source routers globally. We therefore write a 14-bit checksum *cord* side by side with the *rid* field to associate a source router with its domain. The 14 bit checksum hashed from the *did* can successfully associate a *rid* marking with its domain if no more than

```
**Algorithm** Path reconstruction at victim V

let $max\_d$ be maximum attack path length
let $G$ be reconstructed attack graph, initialized with vertex $V$
let $M$ be the upstream inter-domain Internet map
let $D.updid$ be the set of $ids$ of the domains one hop
upstream of domain $D$ in $M$
let $D.rset$ be the set of source routers contained by domain $D$
//Inter-domain attack graph reconstruction
for $d = 1$ to $max\_d$ do
    for each node $D$ at distance $d - 1$ in $G$ do
        for each packet $pkt$ with distance $d$, flag 1 do
            $candidate = pkt.edge \bigoplus D$
            if $candidate$ is in $D.updid$ then
                insert a new edge ($D$, $candidate$) in $G$
//Source router identification
for each attack packet $pkt$ with flag 0 do
    for each node $D$ in $G$ do
        if hash($D$) is equal to $pkt.cord$ then
            insert $pkt.rid$ into $D.rset$
output all the attack paths in $G$
```

Figure 3.7: Path reconstruction algorithm at the victim.

2^{14}(16384) source domains are involved in an attack.

3.2.5 Attack Graph Reconstruction

After collecting sufficient number of marked packets, the victim would perform attack graph reconstruction to identify the attack paths and the potential attack sources. Fig. 3.7 describes the reconstruction procedure.

The attack graph reconstruction procedure consists of two phases. The first phase is inter-domain attack graph reconstruction using packets marked with domain *ids* and then by the end of first stage, the source domains are identified. In the second phase, the algorithm relies on the packets with router *id* markings to identify the source routers.

The inter-domain attack graph reconstruction part uses packets with *did* markings (i.e. with flag equal to 1). The packets with *did* markings are first sorted by their distance from the victim, and the reconstruction algorithm starts from recovering the domains one hop away from the victim using the did-marked packets at distance equal to 1: the property of XOR that $\alpha \bigoplus \beta \bigoplus \alpha = \beta$ allows us to decode the first upstream domain β of the victim domain α given an attack packet marked with the value: $\alpha \bigoplus \beta$ and distance equal to 1. Likewise, an upstream domain of β relative to V could be decoded in the same manner and so forth for the entire path. Thus the domain *ids* on the attack paths are decoded hop-by-hop during inter-domain attack path reconstruction.

In case of DDoS attack, there would be multiple paths in IP traceback, and all domain-id-marked packets at distance d, pkt_i, will generate a number of candidate domain *ids* by XORing the edge it contains with the *ids* of previously reconstructed domains at distance $d - 1$. If we denote a candidate as D_{ij}, which is decoded from pkt_i and the known endpoint of the edge is node D_j. The victim could check the upstream domain topology, M as a road-map to verify candidate D_{ij} by checking if an edge does exist between the candidate and node D_j on M. We denote the set of upstream domains of domain D as D.updid, so the candidate is verified only if it is contained by D.updid. Node D_j and the verified candidate D_{ij} would therefore make up an edge on the reconstructed attack path.

In the second phase, the algorithm relies on *rid* markings to identify the source routers. Let D.rset be the set of source routers within domain D. To locate the source routers with their respective domain, the victim first sorts the *rid*-marked packets by their cord field, and then performs a breadth-first search from the victim, on the reconstructed inter-domain attack graph, level by level until it gets to the node with a matching checksum and adds their router *ids* into D.rset (it is initially empty and it does not include any duplicates).

The victim thus reconstructs the attack graph that incorporates the identified intermediate domains and the source routers. This process would be completed in a quite short interval, less than 10 seconds for 1000 distributed attackers (refer to the simulation results); so that this scheme enables a pretty fast response against the attacks.

During the DDoS attacks, the packets arriving at the victim would include both legitimate packets and attack packets, and if the victim collects all the marked packets for a sufficiently long period of time, we may reconstruct the "traversed paths" of all the marked packets towards the victim. But according to probabilistic packet marking scheme, in a relatively short period of time (usually, the number of duplicates of markings is used to indicate if we have collected sufficient packets for attack paths reconstruction), the majority of marked packets that the victim received are attack packets, so that the reconstructed attack graph would only consist of attack paths (the presence of marked legitimate packets is too scarce to make up even one path). The reasons behind this are: 1) for IP traceback methods, only the flooding DDoS attacks problem is considered, which is a common assumption for IP traceback methods. The attack traffic would have a much higher volume than the legitimate traffic. 2) When the marking probability is low, within a short time, the victim can only collect sufficient attack packets to reconstruct the attack paths only. Therefore, the attack graph reconstructed by the IP traceback scheme would contain only the attack paths that the attack packets traversed.

3.2.6 IP Header Overloading

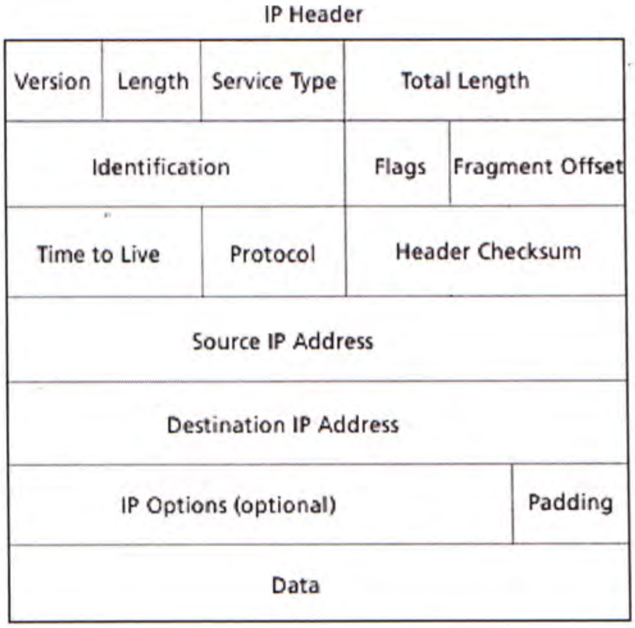


Figure 3.8: IP header format.

We have no choice but overloading the IP header of an IP packet, if we do not want the additional information add extra traffic to the network. Fig. 3.8 depicts the IP header and its fields. We are only interested in the fields that we could employ in the marking scheme.

It is not easy to overload an IP packet header while having the minimum impact on the Internet infrastructure. The fields defined in the IP protocol are always bearing their own purpose and function. The fields where the marking is to be done must be selected carefully. There are not perfect solution in the literature for this problem yet. We are actually forced to make compromises that disadvantage certain kind of traffic.

The proposed marking scheme overloads 25 bits space in IPv4 header; the 25 bits space consists of the 16-bit Fragment Identification field, 1-bit fragmentation flag and 8-bit Type of Service (ToS) field. Employing the 25 bits in the IP header for marking was first advocated by Dean et al. [24].

Type of Service (ToS) field. ToS field is also known as reliability field. ToS is used to indicate whether or not this data unit requires high reliability or normal service. If high reliability is indicated, it may be necessary to apply additional services at the upper layers to provide a high-reliability transmission, or it may mean that the data unit should not be routed over certain routes which may not provide a particular level of reliability. The ToS field is currently not set except for extreme unusual cases.

Fragment ID field. is a 16-bit field used by IP to permit reconstruction of fragments; The identifier field is a 16-bit field used to correlate data unit fragments. When a data unit is fragmented, a number is assigned by the source to the fragments so that the receiver can match the IDs and reassemble the packet. The ID for associated fragments is the same, so the receiver can determine which fragments belong to each other.

Fragmentation flag. There is a 3-bit flag field used to qualify data units for fragmentation and to identify the last fragment in a series of fragmented data units. The first bit in the 3-bit field is always set to zero. The second bit is used to identify whether or not fragmentation is allowed for a data unit. The first bit is used in our marking scheme since current Internet standards simply require it to be zero.

The standard does not define how ToS should be implemented within an internet. It is left to the network provider to determine how these should be implemented within its own network. The protocol does provide the mechanisms for various service types if network providers choose to use them. Nevertheless, the bits are almost always uninitialized.

The Fragmentation ID field is commonly used as a marking field and the backward compatibility was discussed in Savage's paper [45]. Placing a marking in the IP Identification field of every packet in the network is incompatible with the current IPv4 fragmentation mechanism (except under very strict network assumptions such as no packet reordering or loss). Despite the fact that fragmented traffic represents between 0.25% and 0.75% of packets in the Internet [47] [52]. The IP header overloading employed in the IP traceback scheme negatively impacts users that require fragmented IP datagrams and or desire the differentiated traffic.

These problems are hardly unique to traceback and are inherent limitations that come about from attempting to coexist with or co-opt protocol features that did not anticipate a new use [45]. Researchers proposed to selectively enable traceback support in response to operational needs as one possible solution to this issue. The packet marking is enabled in the network only if a "request" for IP traceback is broadcasted. Since a network requesting such support is presumably already suffering under an attack, any minor service degradation for fragmented or authenticated flows would be acceptable.

There are also some proposals on marking in the IPv6 header; however, it is not to be discussed in this work.

3.3 Experiments on the Packet Marking Scheme

We have performed a good number of simulation experiments to examine the feasibility and to evaluate the performance of our proposed packet marking based IP traceback scheme. In this section, we present how the simulation experiments are carried out and depict the corresponding experimental results. Various satisfactory experimental results indicate that our proposed scheme has very good performance and can be effectively put into practice.

3.3.1 Simulation Set-up

We conducted our experiments using an Internet map based on the traceroute dataset of the real Internet from CAIDA's Skitter Internet mapping project [13]; the dataset contains 178,207 distinct traceroute paths widely distributed over the entire Internet. The routes are all from a single origin to multiple hosts on the Internet. In our simulations, we assume that this origin is the victim and the attackers and legitimate users are randomly distributed among the destination hosts in the map. The attack paths are randomly chosen from the paths in the map. The whole traceroute dataset are used as the upstream topology map from the victim, which is required for the attack graph reconstruction.

This traceroute has each hop as a router IP address. As we would conduct our simulation at a granularity of domain, we split the 32-bits IP address in half. For the experiments, we use the first two bytes of an IP address as a domain *id* used in the *did* marking based on the assumption that the routers sharing the same first two bytes belong to one domain. On the other hand, the last two bytes of an IP address would be processed to be used as router *id* which would be used in the router *id* marking. Thus we process the traceroute dataset to make it an Internet topology at a domain level.

The simulation results are obtained through numerous experiments using IP traceback simulation programs written in C in Windows platform(Pentium IV processor with speed in 1100MHz). They will be presented as several plots and evaluated in the later sections. Each data point in the plots corresponds to an average value of around 1,000 experiment runs. Since the metrics: minimum number of packets for the reconstruction of an attack path of a certain length, false positives, number of attack sources, and attack path reconstruction time are independent on the actual values of the parameters concerning bandwidth and data rates, we will use relative ratios for characterizing these parameters instead of the actual rates.

For performance evaluation purposes, we assume that the packet sending rate at all attack sources is the same and the transmission rate along each path is constant throughout the simulations. Moreover,

Table 3.1: Control parameters and performance metrics used in evaluating IP traceback scheme.

p	Marking probability for did marking
q	Marking probability for rid marking
Number of packets	Minimum number of packets needed for the reconstruction of an attack path of a certain length
False positives	It refers to a router added to the reconstructed attack graph by the traceback mechanism, which is however not included in the original attack path.
Reconstruction time	The estimated time required for reconstructing the path back to the attackers

there will not be any abnormal or high bandwidth traffic congesting the network except the flooding packets generated from the attack sources.

3.3.2 Experimental Results and Analysis

The experiments are performed to assess the performance of our marking scheme characterized by a number of parameters, namely the minimum number of packets for the reconstruction of an attack path of a certain length, false positives, number of attack sources, and attack path reconstruction time. The control parameters and performance metrics are listed in Table. 3.1.

Each router on the attack paths simulates marking the packets as depicted in the marking algorithms. And the victim simulates applying the proposed reconstruction algorithm to reconstruct all the attack paths.

The results are as presented in Fig. 3.9 to Fig. 3.11. Each data point in Fig. 3.9 to Fig. 3.11 corresponds to an average value based on around 1000 experiment runs.

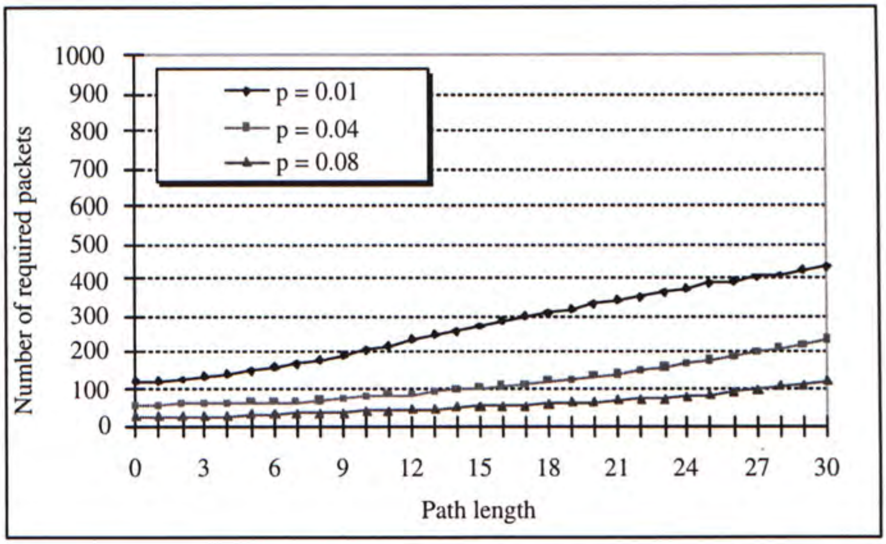


Figure 3.9: Number of packets required for attack paths reconstruction for different path lengths and different marking probabilities .

Fig.3.9 shows the minimum number of packets required for the reconstruction of attack paths of different lengths and different marking probabilities. Since a packet will normally traverse no more than 30 routers in the Internet to reach its destination, the attack path lengths considered in the experiments range from 0 to 30. In general, for each marking probability, the number of required packets for path reconstruction increases linearly with the path length. For the case with marking probability 1%, and path length 30, the required number of packets would be around 400; if the marking probability is 4%, roughly 200 packets would be required.

When compared with other IP traceback methods, our proposed marking scheme requires fewer packets for reconstruction. For instance, for a marking probability of 4% and path length 30, scheme 2 of Advanced Marking Schemes (AMS) with $m > 5$, around 1000 packets are required for reconstruction, where m is the number of hash functions used to encode the router identification. When with $m > 6$, 4000 packets are required [49].

Further more, by analysis, we can show the number of required packets could be cut by half because of our shorter path. The probability of a packet arriving at the victim site would be marked on its way is $P = p(1-p)^{r-1}$. p is usually assigned as $1/d_{max}$, where d_{max} represents the maximum number of hops on a path from attacker to the victim. Because $(-1/x)$ is greater than $\exp(-1/(1-x))$. Therefore, the probability P is at least $(1/d_{max}) * \exp(-(d-1)/(d_{max}-1))$, which, since d is less than d_{max}, is at least $1/e * d_{max}$.

If we conservatively assume a partial path within each domain contains two routers, a maximum domain-based path length would be half of its equivalent router based path length. This implies that our marking scheme needs to handle attack paths with average path length equal to one half the path lengths of those handled by other marking schemes.

Moreover, we need only one marked packet to identify a domain; whereas other marking schemes normally employ full IP address markings for full-path reconstruction, and several packets are usually required to identify each router on an attack path. For instance, eight marked packets are employed in both Savage's method [45] and Song and Perrig's method [49] to encode each router's identity. So our marking scheme needs substantially fewer packets for attack graph reconstruction.

Fig. 3.10 presents the number of false positives for different num-

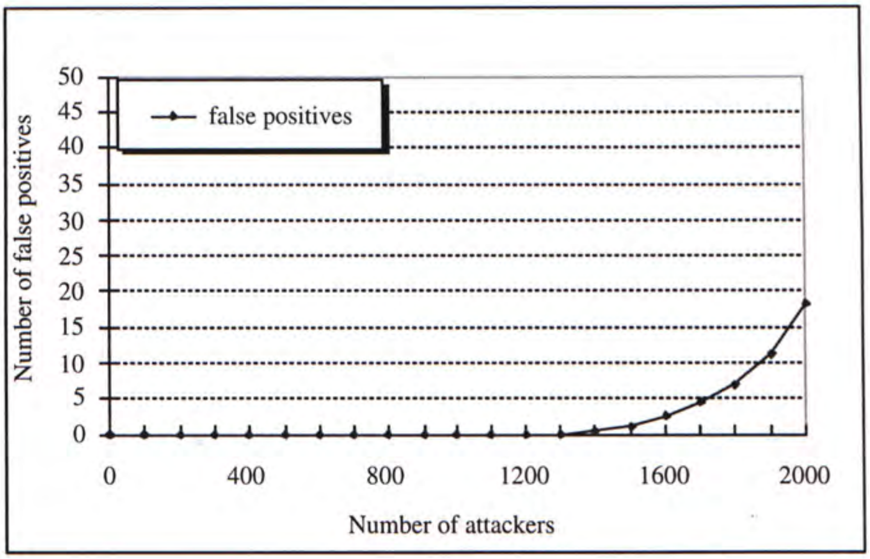


Figure 3.10: False positives generated for different number of attackers.

ber of attackers in the range 100 to 2000. It shows that our marking scheme generates relatively small amount of false positives. Even in the presence of 2000 attackers, it generated only few false positives.

Therefore, regarding the amount of false positives generated, our marking scheme is comparable with scheme 2 with $m > 7$ of Song and Perrig's method [49], which also produced less than 20 false positives in presence of 2000 highly distributed attackers. The hash functions we use to encode an individual router id from the corresponding IP address are relatively simple, so some of the false positives could be attributed to the possible collisions of the hash values of the hash function employed. Had we managed to find a collision-resistant hash function for the experiments, the number of false positives could have been even smaller.

While our marking scheme has a computation complexity of around $\Omega(dn^2)$, the method of Savage, et al. [45] and the method of Song and

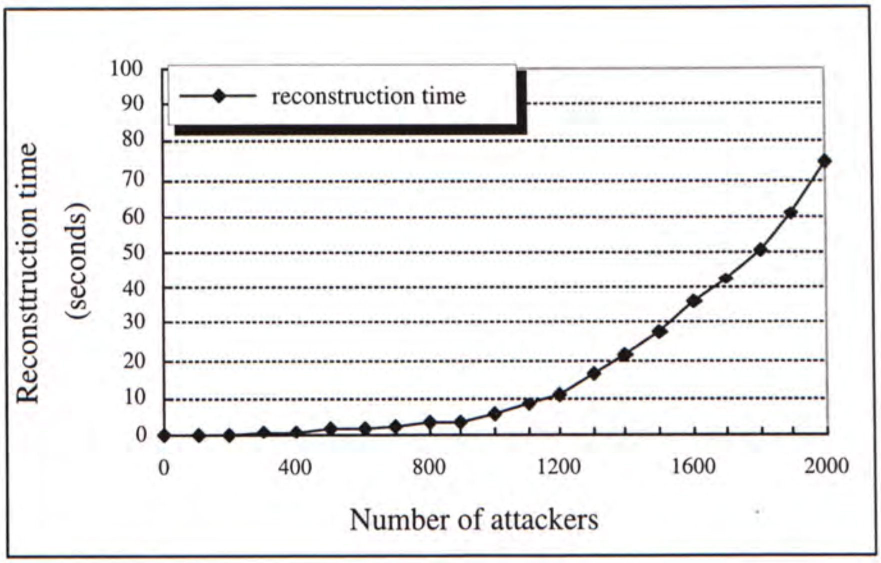


Figure 3.11: Reconstruction time for different number of attackers.

Perrig [49] have a complexity of around $\Omega(dn^8)$ and $\Omega(dn^2)$ respectively, where d is the maximum path length and n is the number of attacking hosts.

Since our domain based marking scheme involves a smaller distance d, its complexity is relatively small. Fig. 3.11 presents the reconstruction times of our reconstruction algorithm for different number of attackers, measured on a 1500MHz Pentium IV PC platform. The results show that in general the attack graph reconstruction could be completed quite rapidly. Even for the case of 2000 attackers, it takes only about 80 seconds to reconstruct the attack graph, which is considered quite a fast response to a highly distributed large-scale DDoS attack.

Through the simulation experiments on the proposed marking scheme, we observe the following:

(a) It requires a much smaller number of packets for attack paths reconstruction than other methods such as AMS [49];

(b) It can handle multiple attack sources effectively in very large scale;

(c) The number of false positives generated even in the presence of 2000 attack sources is relatively small;

(d) It performs attack paths reconstruction quite rapidly and takes only around 50 seconds to reconstruct as many as 2000 attack paths on a Pentium IV PC platform (Processor 1100MHz). Thus it could be used to locate attack sources in real time, which is one of the critical steps in defending against DDoS attacks.

□ **End of chapter.**

Chapter 4

DDoS DEFENSE SCHEMES

The task of identifying the actual sources of the attack packets is addressed in the last chapter. However, IP traceback techniques neither prevent nor stop the attack; IP traceback technique by itself has limited capability of sustaining the victim's service availability during a DDoS attack [21]. Therefore, another key task defeating a DDoS attack is to mitigate the attack traffic towards the victim. In this chapter, we focus on preserving the service availability of the victim by improving the throughput of normal traffic, so that the DDoS attacks can be ultimately thwarted.

We proposed two schemes to efficiently defend against DDoS attacks: packet filtering at the victim end and traffic rate-limiting at the source end. They are based on the packet marking scheme for IP traceback which enables us to detect and thus discard the attack packets with quite high probability in real time.

4.1 Scheme I: Packet Filtering at Victim-end

4.1.1 Packet Marking Scheme Modification

In the marking scheme for IP traceback, two types of marking algorithms are supported. The packets are to be marked by the *rid* marking or *did* marking, depending on whether it is at the ingress point to the network. We accordingly have two marking probabilities, denoted as p and q for *did* marking and *rid* marking respectively.

However, the probabilistic nature of the marking algorithms are not compatible to the marking based packet filtering mechanism. In the packet filtering mechanism, each packet flowing to the victim will be checked against its marking to find the malicious ones. It hence requires that every packet bear a marking. We therefore modified the *rid* marking algorithm. The router would perform deterministic marking for all packets issued from its network. The *did* marking is the same as the previous proposal; a relatively small probability would be enough to ensure that we could infer the inter-domain attack paths. Fig. 4.1 shows the modified *rid* marking algorithm, which would be used in the packet marking based filtering mechanism.

First, by marking all the packets deterministically, we ensure that even a spoofed marking (a packet with erroneous information injected by attackers) would also be overwritten with a correct marking by the participating routers. Second, the sources of the packets can be easily deduced from the *rid* markings. Third, marking packets at the source network is a better choice than leaving the processing overhead to the backbone routers.

Algorithm Router id marking revised

```
for each packet pkt passing through R do
    generate a random number $x$ within [0..1]
    if $x < q$ then
        pkt.edge = did
        pkt.dist = 0
        pkt.flag = 1
    else
        pkt.rid = rid
        pkt.cord = hash(did)
        pkt.flag = 0
```

Figure 4.1: Modified router id marking algorithm for packet filtering.

4.1.2 Packet Filtering Algorithm

After identifying the attack source(s) through IP traceback, packet filtering would be carried out to alleviate the damaging effect caused by the DDoS attacks. And markings corresponding to the source routers and the nodes on the identified attack paths would serve as filtering signatures to instruct the packet filtering.

A filtering agent can be deployed either on the victim machine or, preferably, on a dedicated machine such as firewall or gateway in front of the victim on the path to the victim. A set of filtering agents can also be deployed at the ISP's side of the last hop link, so that the attack packets can be filtered even before it gets to consume the victim's network bandwidth. Fig. 4.2 presents the packet filtering process by a filtering agent. It reduces the intensity of the attack by removing most of the attack traffic.

Basically, a filtering agent checks if a packet is an offending packet

Algorithm Packet filtering algorithm

```
let S represent the bitmap for rid markings
let T represent the bitmap for did markings
let $rid_f$, $did_f$ be filtering probabilities for packets with rid
markings and did markings respectively
for each packet pkt
        if pkt contains a rid marking i then
                if (S[i] == 1) then
                        drop pkt with probability $rid_f$
        if pkt contains a did marking j then
                if (T[j] = = 1) then
                        drop pkt with probability $did_f$
```

Figure 4.2: Packet filtering algorithm.

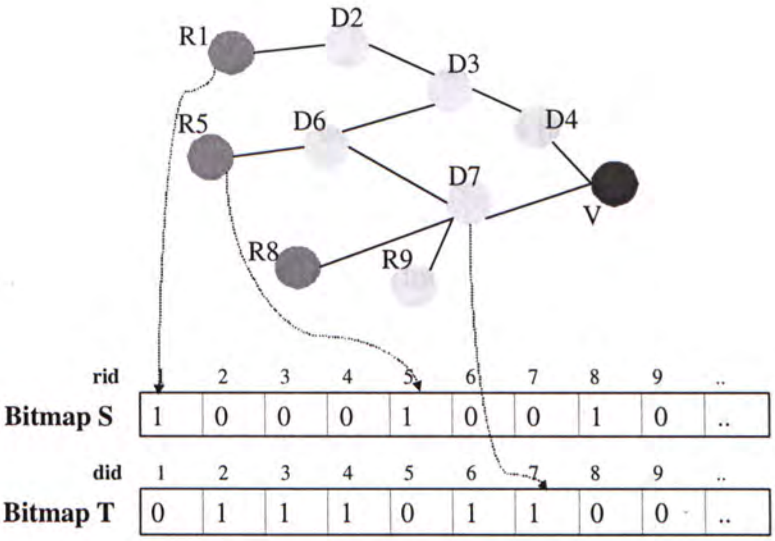
by examining if the inscribed markings match a filtering signature. With our marking scheme, a marked packet can contain either one of two types of markings: *rid* or *did* marking. Accordingly, the filtering agent would check the marking concerned against one of two bitmaps: either the one for *did* markings or the one for *rid* markings, which encode the attack graph, as shown in Fig. 4.3.

Let S be the bitmap with each entry indexed by the *rids* and containing one bit with binary values: it would be set to 1 if its index *rid* marking corresponds to any identified attack source, and set to 0 otherwise. In the same manner, bitmap T encodes the set of *did* markings corresponding to the domains on attack paths.

4.1.3 Determining the Filtering Probabilities

The marked packets can be classified into three types:

I. Legitimate packets with markings matching none of the attack



rid	1	2	3	4	5	6	7	8	9	..
Bitmap S	1	0	0	0	1	0	0	1	0	..

did	1	2	3	4	5	6	7	8	9	..
Bitmap T	0	1	1	1	0	1	1	0	0	..

Figure 4.3: Attack graph encoded as bitmaps.

signatures, that is, indexed to a zero value on the bitmaps.

II. Packets with *did* markings matching signatures from bitmap T; however, it is unsure whether they are legitimate or attacking ones.

III. Attack packets matching signatures from bitmap S.

Obviously, the victim's bandwidth is for the type I packets, though some legitimate packets may fall into the second category. A packet with type II marking may also be a legitimate packet, if it is marked where the attack path and its path have one part overlapped. It is thus possible to filter packets with *did* markings falsely, which is referred to as a *false drop.* In addition, the victim bandwidth should be fully utilized so that at least some type II or type III packets can reach the destination and later trigger a recovery phase. The following process is used to decide the filtering probabilities for the three types of packets.

1. The legitimate traffic (bearing type I markings) at rate R_l should utilize the victim bandwidth without restriction; where we denote the victim bandwidth as w_0. There is no packet filtering imposed on this type of traffic.

2. For the second type of traffic (incoming at rate R_2; and R_3 represents the incoming rate of the type III packets), the filtering probability did_f will be set such that the regulated traffic rate of type II packets is equal to the remaining bandwidth if there is victim bandwidth left after the first step, as shown in Eq. 4.1.

3. If there are still bandwidths left after step 1 and step 2, set filtering probability rid_f according to Eq. 4.2.

$$did_f = \min(1, \max(\frac{R_2 + R_1 - W_0}{R_2}, 0)) \tag{4.1}$$

$$rid_f = \min(1, \max(\frac{R_3 + R_2 + R_1 - W_0}{R_3}, 0)) \tag{4.2}$$

There always exists delay between the time measuring the traffic volume and the time performing the traffic regulation with the afore-calculated parameters, so the algorithm shown above has to be executed in short enough time intervals, and the above incoming traffic rate are measured periodically by the filtering agents.

The legitimate traffic throughput with varying filtering probability did_f would be demonstrated in the simulation section. The server utilization improved by the algorithm will be demonstrated together with that of Scheme II in the simulation section.

In case of multiple upstream filtering agents around the victim, the calculations of the respective filtering probabilities will be based on their local parameters. The probabilities are determined the same way as shown above; however, the available victim bandwidth is divided a priori to be allocated to each upstream link.

4.1.4 Suppressing Packets Filtering with *did* Markings from Nearby Routers

One possible solution to reduce the number of false drops is to suppress filtering packets with a *did* marking from a nearby router. We notice that markings from nearby routers would more likely incur false drops based on the following two observations. First, as the probability of receiving a packet marked d hops away by the victim is (with the marking probability p), it is more likely that the packet has a marking from routers near the victim. Second, the attack graph could be generally viewed as a tree rooted from the victim, and the normal paths and attack paths overlap more near the victim, resulting in more false drops. Therefore, we let pass the packets with markings from routers within certain distance to the victim; and we introduce a parameter r to denote this distance. How r value would affect the performance would be presented in the experiments results section.

4.2 Scheme II: Rate Limiting at the Sources

Although we proposed to filter attack packets at the victim side to provide the fast relief from the overwhelming traffic for the victim, there are several advantages of source-end defenses over the victim-end defenses. First, attacking DDoS attacks at source-end provides better traffic control on the attack traffic. As not all the attack packets would be marked by the source routers and those marked by an intermediate domain may not result in a filtering decision (considering a possible overlapping), the filtering agents can only remove part of the attack traffic. Secondly, the sources should be punished directly and the network re-sources are thus preserved. Thirdly, the victim may collapse under an extremely aggressive DDoS attack, where we can alleviate the traffic only at the source-ends.

Therefore, as we have been able to trace back the sources, we propose the traffic rate-limiting mechanism as the complementary source-end defense method to protect the victim right at the attacking sources. Rate-limiting the traffic is a better choice than simply dropping, since the victim's bandwidth can be thus best utilized.

In this section, we would present how the attack traffic rate-limiting mechanism defends against DDoS attacks by mitigating the overwhelming volume of traffic issued coordinately by the distributed attacking sources to the victim.

4.2.1 Algorithm of the Rate-limiting Scheme

Fig. 4.4 describes the algorithm how we regulate the excess traffic from the identified attacking sources.

With the IP traceback scheme, the victim can reconstruct the attack graph with knowledge of the attacking sources identified. The amount of traffic from each source can be estimated based on the number of packets received at the victim with the specific markings. *Att*[*i*] is a list of estimated rate from the sources and *Sum_att* is the estimated

Algorithm Rate-limiting at source-ends

```
let N be the number of sources identified
let RL[i] represent the source routers performing the traffic
rate-limiting
Let $Att[i]$ be the estimated rate from the list of sources
Let $R_{excess}$ be the excess traffic arriving at the incoming links
Let $Sum\_att$ be the estimated rate of all attack traffic
Let $Total\_arr$ be the total incoming traffic rate

    Estimate the total arrival rate at the victim link
    $R_{excess} = Total\_arr - Vbw$
    $Sum\_att = 0$
    for i = 0 to N do
        $Sum\_att+ = Att[i]$
    //Note $R_{excess}$ may be larger than $Sum\_att$
    for i = 0 to N do
        request RL[i] to drop its traffic with probability
        $MIN(R_{excess}/Sum\_att, 1)$
```

Figure 4.4: Rate-limiting algorithm at source-ends.

rate of all attack traffic. The victim also needs to calculate R_{excess}, the excess traffic arriving at the incoming links. This is the amount of traffic that would have to be dropped even after the victim-end packet filtering has taken effect. $R_{excess} = Total\_arr - Vbw$, where $Total\_arr$ is the total arrival rate of incoming traffic, including both legitimate and attack traffic and Vbw is the victim bandwidth.

$$A_i = \frac{Att[i]}{Sum\_att} * R_{excess} \tag{4.3}$$

is the amount of traffic that source i contribute to the excess traffic, so that source i should drop the packets with the probability of

$$\frac{A_i}{Att[i]} \tag{4.4}$$

i.e.,

$$\frac{R_{excess}}{Sum\_att} \tag{4.5}$$

Note R_{excess} may be larger than Sum_att because not only the attack traffic but also the legitimate traffic may contribute to the amount of traffic exceeding the victim's capacity. If this is the case, the best we can help is to stop all the attacking traffic and the rest legitimate traffic has to compete for the limited bandwidth.

In most cases, the computations above should be carried out by a gateway or firewall delegated the victim in case that the congested victim has run out of resources for handling these things. Obviously, our rate-limiting mechanism is performed with least legitimate traffic would be harmed.

However, the authentication of the rate-limiting requests at the source domains is much of concern, since it could be abused by attackers to cause more severe service disruption. Many sophisticated authentication methods have been proposed in the literature [44] [34], and we do not plan to discuss them here. Actually, a candidate solution could be Intrusion Detection Exchange Protocol and Intrusion Detection Message Exchange Format specifying the communication protocols and intrusion detection language [10].

4.3 Performance Measurements for Scheme I & Scheme II

A number of experiments are carried out to assess the performance of the proposed packet filtering mechanism and traffic rate-limiting mechanism. They are conducted in a simulation environment, using real-world Internet traceroute data from CAIDA's Skitter Internet mapping project [13]. The dataset is composed of more than 178,000 distinct traceroute paths widely distributed over the entire Internet.

The overall performance of our proposed DDoS attack defense mechanism obviously depends on the performance of the two constituent parts — adaptive packet marking scheme for IP traceback and the defending mechanisms.

We presented the performance of our marking scheme in the last chapter. The related parameters and results will be sequentially used as parameters in the experiments in this work, for measuring the following metrics.

The metrics for measuring the effectiveness of both the source-end and victim-end traffic mitigation mechanisms include *Legitimate Traffic Survival Ratio* (LTSR), *Attack Traffic Drop Ratio* (ATDR), and

Table 4.1: Control parameters and performance metrics

p, q	Marking probability
g	The ratio of the sending rate of attacking traffic over the rate of legitimate traffic
rid_f , did_f	Probability to filter a packet with *rid* (*did*) marking created by a router on an attack path
LTSR	The percentage of normal packets that can make their way to the victim during a DDoS attack
False drop	It refers to a legitimate packet dropped falsely by the filtering mechanism.
ATDR	The percentage of attack traffic/packets dropped

the number of *false drops*. Table.4.1 lists the control parameters and performance metrics.

- First of all, we use Legitimate Traffic Survival Ratio (LTSR) to measure the effectiveness of packet filtering; quantitatively, it measures the the percentage of normal packets that can make their way to the victim during a DDoS attack.

- Second, a main challenge to IP traceback based packet filtering scheme is the reliability of a filtering decision based on the markings inscribed in a packet. Since both attack packets and legitimate packets could carry the same markings as long as they traverse along a common attack path, false drops are likely to be incurred.

- Lastly, we define Attack Traffic Drop Ratio (ATDR) as the percentage of attack traffic dropped, and this ratio is desired to be high, ideally to be 1.

The simulation results are as presented in Fig. 4.5 to Fig. 4.10. Each data point in Fig. 4.5 to Fig. 4.10 corresponds to an average value based on around 1000 experiment runs. For simplifying the demonstration, we temporarily assume that all the Internet border routers participate in both the IP traceback marking scheme and they are all compliant to perform the rate-limiting effectively. The analysis with consideration of non-participating routers is to be discussed in the last set of experiments.

The first set of experiments, as shown in Fig. 4.5 to Fig. 4.7, presents the effectiveness of the victim-end packet filtering mechanism with different settings.

Since the performance metrics are independent of the actual bandwidth, for simplicity, we assume the available bandwidth capacity is 1000 units, utilized fully by 1000 normal clients if no attacks are present. During the attack, DDoS attack traffic would occupy 80% bandwidth of the total victim bandwidth and the throughput of normal traffic is only 20% without the filtering mechanism.

Fig. 4.5 and Fig. 4.6 show that with a greater filtering probability for packets with *did* markings, more attack packets would be dropped and more normal packets could be preserved. When the filtering probability did_f approaches 1, the attack packets drop ratio ATDR would also be close to 1; that is, most of the attack packets could be filtered (as shown in Fig. 4.6). However, even if all the attack traffic could be filtered, the bandwidth available for the normal traffic LTSR would still be below 100%. For instance, for the case of *did* marking probability p equal to 4%, LTSR is below 90% even when ATDR is close to 1. This could be explained by the fact that some normal packets could be filtered as false drops as the normal traffic and attack traffic may traverse along an overlapping subpath.

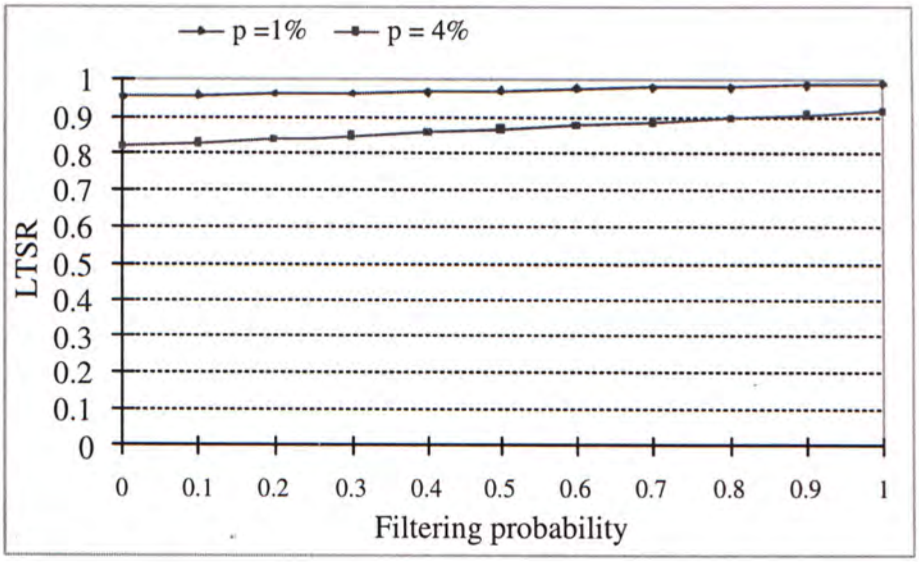


Figure 4.5: Plot of LTSR versus filtering probabilities for packets with *did* markings.

The performance of the filtering mechanism, with the inclusion of the feature of suppressing filtering packets with *did* markings from routers close to the victim is also examined by experiments. The number of attackers considered range from 100 to 1000. The attack paths and the normal paths have some overlapping.

Packets filtering suppression within distance r (r = 2,.. 4) hops away from victim are examined as presented in Fig. 4.7. The packets are marked with *did* markings with marking probability of 4%. The number of false drops is reduced by half when the distance is increased from 2 to 4, and even more with a higher r. This feature would be adopted under such circumstances that we have preserved enough bandwidth for the normal traffic with enough attack packets removed

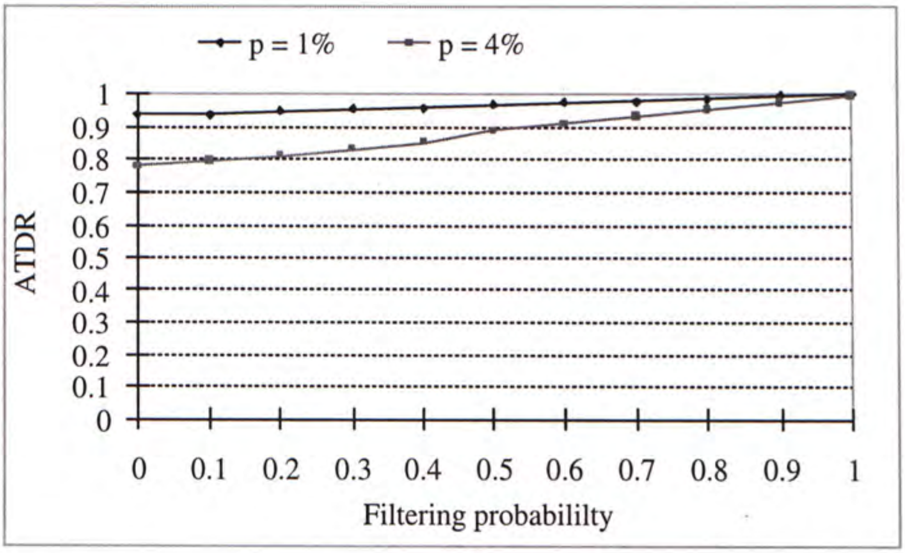


Figure 4.6: Plot of ATDR versus filtering probabilities for packets with *did* markings.

and thus the avoidance of the false drops would be more concerned. However, with including this filtering suppression feature, the legitimate traffic survival ratio (LTSR) may become lower, if it has not enough bandwidth to get through, since we have to allow some attack traffic through.

Another set of experiments discuss about the source-end rate-limiting mechanism.

Fig. 4.8 shows how LTRS and ADTR vary as the number of the attackers increases, from 500 to 2000, with 500 attackers increment in each single experiment; at the same time, the LTRS without any defense responses under DDoS attacks are also shown for comparison purpose.

Fig. 4.8 demonstrate that the proposed rate-limiting mechanism is an efficient DDoS defense mechanism with nearly 100% legitimate traf-

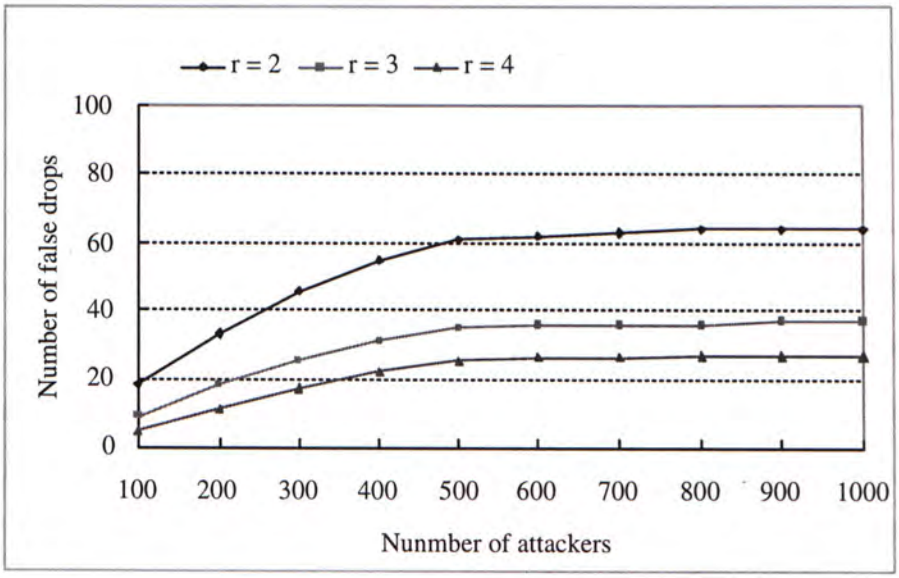


Figure 4.7: Plot of number of false drops versus number of attackers if suppressing packets filtering r hops away from the victim.

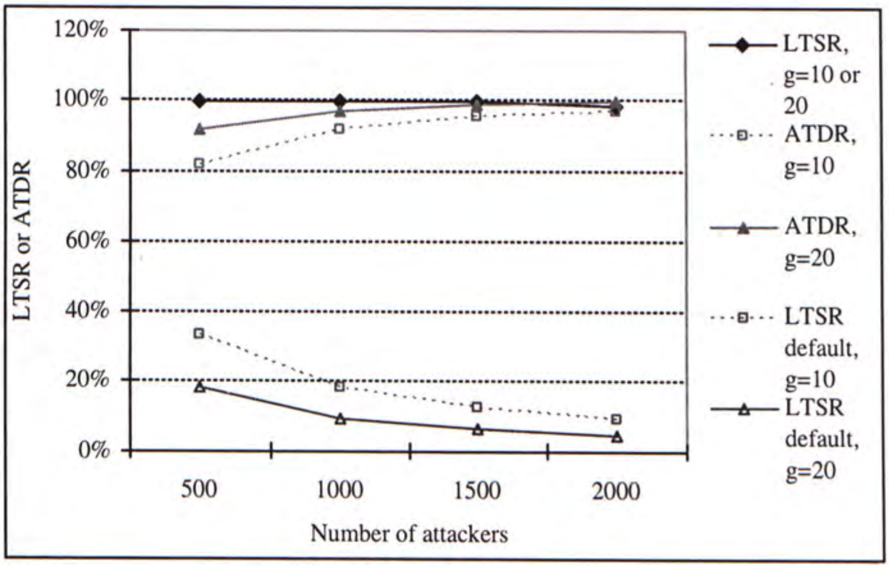


Figure 4.8: Plot of LTRS and ADTR versus different number of attackers and plot of LTSR without DDoS defense response.

fic throughput and the majority of the attack traffic is removed without affecting legitimate traffic at all. However, when the number of attackers becomes more than 1000, the IP traceback will inevitably produce a small number of false positives, resulting in some legitimate packets rate-limited, so LTSR is slightly below 100% as number of attackers exceeds 1000. The amount of removed attack packets, i.e., ATDR is determined by both the available bandwidth and the aggressiveness of the on-going DDoS attack; it is always no larger than enough for us to bring the excess traffic down to control.

Note that for rate-limiting mechanism, the existence of path overlapping will not affect the parameters and results, as part of the legitimate packets would get marked on the overlapped sub-path but not marked as *rid* marking, so will not be counted in the sending rate, $Att[i]$.

Fig. 4.9 compares how the two defense schemes perform as well as the results without any defenses (which corresponds to the column called "Default") in terms of LTSR and victim's bandwidth utilization. In our testing, the simulated DDoS attack is composed of 2000 distributed attackers and 1000 legitimate users randomly selected across the Internet. The capacity of the victim's bandwidth is 2000 units.

When there is no any DDoS attack defense mechanism, the victim suffered a lot under the simulated attack, with only 9.5% legitimate traffic surviving. With victim-end packets filtering alone (where we set the marking probability p at 4%, and the did_f is set to be 1; filtering suppression is also applied, with r equal to 3), 91.2% legitimate traffic can be protected. The traffic limiting-rate scheme guarantees the optimal server utilization and meanwhile adequately removes the attack

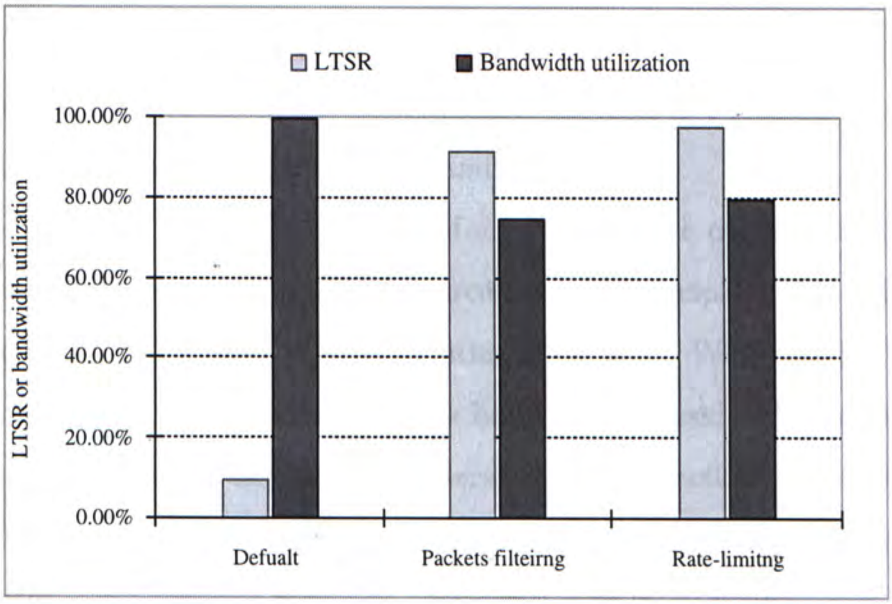


Figure 4.9: Histograms of LTSR and bandwidth utilization without any DDoS defenses, with packets filtering at the victim and with the source-ends rate-limiting mechanisms respectively.

traffic at the source routers. As demonstrated in the simulation results of the proposed IP traceback scheme, a small number of false positives will be produced by the underlying IP traceback scheme so there would be a small portion of legitimate traffic loss.

The throughput of legitimate traffic is significantly improved either with packet filtering mechanism or with traffic rate-limiting. In another also IP traceback based packet filtering method [53], a maximum legitimate traffic throughput is around 60%. Furthermore, the proposed defense schemes only slightly affect any legitimate traffic, which is addressed as poor traffic in [49] and [45].

Note that the above two sets of experiments are conducted in an ideal environment, with all the source routers participating in and all compliant to perform the rate-limiting effectively. While in the real world, there are two kinds routers have to be considered: the non-compliant routers and legacy routers. A non-compliant router is a router not willing to participate in our defense scheme, and a legacy router is not software router so that can not be modified to adopt our defense scheme. We hereafter call them non-participating routers for both cases.

The effectiveness of the packets filtering mechanism with varying percentages of non-compliant routers in the simulation networks is shown in Fig. 4.10. Note that if we assume the legacy routers are distributed uniformly no the network, on average, there should be equally percentage of legacy routers among both sources ones and intermediate ones. The defense effectiveness decreases as the number of non-compliant routers increases. When the deployment of our scheme exceeds 50%, the results are two times better than without defenses.

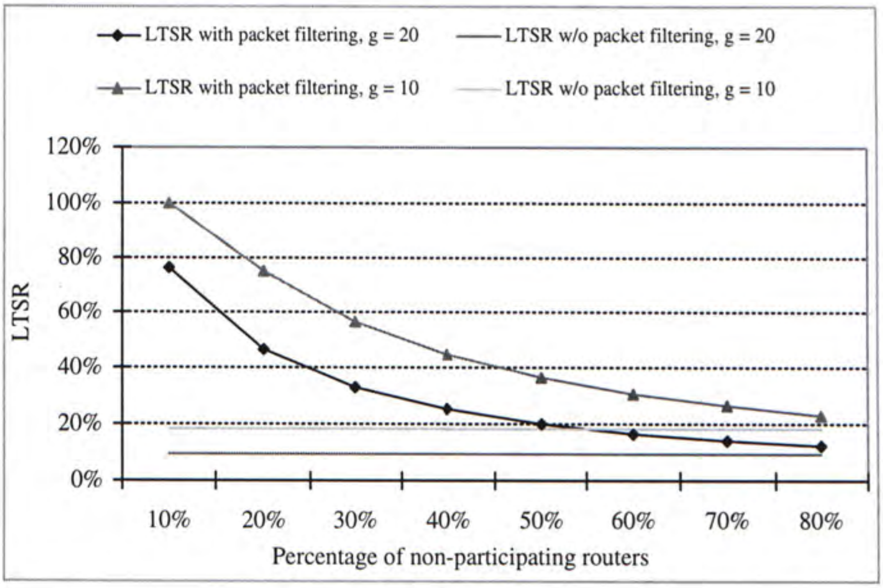


Figure 4.10: Plot of LTSR with varying percentage of non-participating routers deployment.

Therefore, there is a tradeoff between the performance improvement and the amount of coordination between the victim and the intermediate network routers.

In summary, our simulation studies based on real-world Internet topologies demonstrate that the schemes are very effective in improving the throughput of legitimate traffic during DDoS attacks. The operations required of routers (adaptive packets marking) are fully in line with the operations of IP traceback, making the schemes as practically deployable as possible.

□ **End of chapter.**

Chapter 5

CONCLUSION

This thesis first presents a novel inter-domain based packet marking scheme for IP traceback. This scheme enables us to detect and thus to discard the attack packets with quite high probability in real time. We therefore make use of the packet marking scheme to develop a practical approach composed of packet filtering and traffic rate-limiting mechanism to efficiently defend against DDoS attacks.

5.1 Contributions

We view the contributions of this work as follows:

- **Effective DDoS defense**

- The adaptive packet marking scheme is an efficient IP traceback scheme. It can handle multiple attack sources effectively and can reconstruct essentially all attack paths with the presence of thousands of highly distributed attacking hosts.

- From the experiments results, it can be observed that the packet filtering scheme and rate-limiting scheme are highly effective in preserving the amount of the legitimate traffic and guaranteeing the optimal utilization of the victim's bandwidth.

- **Innovations behind the proposals**

- The innovation of the adaptive packet marking scheme lies in that it reconstructs inter-domain attack paths and identifies attack sources at the same time. The identification of a source router and an intermediate domain are of different importance for IP traceback in view of the victim and this is noticed and investigated by us for the first time, to the best of our knowledge.

- We integrate the domain based packet marking scheme with the packet filtering scheme, filtering malicious packets according to the bitmaps encoded from the reconstructed attack graph. In the literature, there are very few proposals suggesting to use traceback information as filtering signatures.

- When compared to other source-end defense methods which involve detecting anomaly traffic rate from the network monitored, the traffic rate-limiting in our scheme is performed based on the actual attack source information. Thus the chance of damaging legitimate traffic by the scheme can be minimized.

- **IP traceback performance improvements**

The performance of IP traceback schemes are commonly measured in terms of the minimum number of packets for the reconstruction, number of false positives, attack path reconstruction time and deployment overhead. The adaptive packet marking scheme is extensively tested and proved to possess the following advantages compared to other methods:

- It requires a much smaller number of packets for attack paths reconstruction than other methods such as AMS [49] as presented in section 3.3.2, as it uses the relatively short *id* instead of a full IP address and keeping track of the inter-domain attack path instead of full path traceback;
- It performs attack paths reconstruction quite rapidly and takes only around 80 seconds to reconstruct as many as 2000 attack paths on a Pentium IV PC platform. It thus could be used to locate attack sources in real time, which is one of the critical steps in defending against DDoS attacks;
- As demonstrated in section 3.3.2, the number of false positives generated even in the presence of 2000 attack sources is relatively small. It thus provides great reliability to be utilized by the packet filtering and rate-limiting mechanism to determine whether a packet is malicious or not.
- As the marking algorithms involve only the border routers and the overhead incurred to the routers is kept to a minimum, this scheme ensures a practical implementation.

- **Throughput of legitimate traffic significantly improved**

Experiment results show that the overall performance of the proposed defense schemes are promising.

- As demonstrated in section 4.3, with the introduction of the packet filtering scheme, around 80% attack packets can be successfully removed. With an extremely strict setting of filtering probability for packets with *did* markings, almost all attack packets can be filtered out, though some normal packets could be filtered as false drops as well.

- Still as demonstrated in section 4.3, the throughput of normal traffic with the packet filtering scheme could be improved from 20% (without filtering) to around 80%. So the service availability could be much improved.

- The packet filtering process is of great accuracy and reliability. We derived different filtering probabilities to filter packets, since some of them are more likely to be attack packets and some may be "poor traffic" if they traverse the overlapped subpath of the attack paths.

- At the source end, the rate-limiting scheme has been shown in section 4.3 to be capable of improving the throughput of legitimate traffic from 10% (without applying rate-limiting under DDoS attack) to over 90% in an ideal environment.

- **Effective even with partial deployment**

In practice, it may not be possible to require all domain border routers to participate in the packet marking scheme.

- The packet filtering scheme could achieve satisfactory performance even if there is only a partial deployment of the packet marking scheme. Few of the existing proposals have addressed how the existence of non-compliant routers or legacy routers in the real world would affect their effectiveness.

- As demonstrated in section 4.3, when the packet filtering scheme involves around 50% of the routers for packet marking, the normal traffic throughput is at least 2 times better than the case without applying the defense scheme.

To summarize, the proposed defense is innovative, effective, lightweight, relatively easy to be deployed, and thus practical for real implementation to defend against DDoS attacks.

5.2 Discussion and Future Work

In practice, there is little incentive for a source domain to conduct traffic rate-limiting, due to lack of substantial benefit for this action. If it can be only viewed as an alms deed, we can expect little concern from any ISP, especially there is risk that a traffic control may harm its credit to the customers [38]. So we have to discuss the deployment challenge of our proposed defense schemes.

The involvement of non-victim networks is a problem commonly exists in the IP traceback approaches and DDoS defenses. These solutions, such as Pushback approach, ingress filtering technique and etc., require not only router support, but also wide-spread deployment. For the packet filtering scheme, we do not need the coordination of the non-victim networks; for the rate-limiting scheme, the involvement of network administrations is limited to the source networks, which however has the responsibility to stop the attacking sources within its network. In addition, the sites deploying the rate-limiting scheme would benefit from protecting the bandwidth from illegitimate use. They do not have to deal with the legal implications of hosting DDoS attacking hosts.

Another problem is that everyone on the Internet could claim himself as a DDoS attack victim. The attacker could conduct a DDoS attack by sending a forged request, asking the source routers to stop traffic to a destination. Therefore, our method is robust only if we have methods to guarantee that the rate-limiting request is sent exactly from the IP address it claims.

In the adaptive packet marking scheme, we have included a checksum for associating the *rids* with its source domain. We view the checksum with potential to be used to check the integrity of markings in a packet. We may also make use of the checksum to authenticate a rate-limiting request. For instance, the authenticated marking can be included in the request for authentication purpose. In the near future, we would explore this authentication potential of our proposal to improve the current work. This process should be simple enough to fulfill speed requirements and sophisticated enough to protect against counterfeits. Furthermore, the authentication process itself should be DDoS resistant.

Bibliography

[1] UDP Port Denial-of-Service Attack, 1996. CERT Advisory CA-1996-01, http://www.cert.org/advisories/CA-1996-01.html.

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

[3] Denial of Service Attacks - An Emerging Vulnerability for the "Connected" Network, 1999. A White Paper prepared by SonicWALL, Inc.

[4] The DoS Project's "Trinoo" Distributed Denial of Service Attack Tool, Oct 1999. http://staff.washington.edu/dittrich/misc/trinoo.analysis.

[5] The "Stacheldraht" Distributed Denial of Service Attack Tool, Dec 1999. http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.

[6] The "Tribe Flood Network" Distributed Denial of Service Attack Tool, Oct 1999. http://staff.washington.edu/dittrich/misc/tfn.analysis.

[7] Analysis of the Shaft Distributed Denial of Service Tool, Mar 2000. http://security.royans.net/info/posts/bugtraq_ddos3.shtml.

[8] TFN2K - An Analysis, Sept 2000. http://www.xfocus.net/articles/200009/6.html.

[9] Yahoo on Trail of Site Hackers, Feb 2000. http://www.wired.com/news/business/0,1367,34221,00.html.

[10] IETF's Intrusion Detection Exchange Format Working Group, Apr 2002. http://www.ietf.org/html.charters/idwg-charer.html.

[11] Overview of Attack Trends, Apr 2002. CERT Coordination Centre, http://www.cert.org/archive/pdf/attack_trends.pdf.

[12] Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy Issues, Oct 2002. http://www.foxnews.com/story/0,2933,66438,00.html.

[13] CAIDA's Skitter Project, 2004. http://www.caida.org/tools/measurement/skitter/results.xml.

[14] C. Adams and J. Gilchrist. RFC 2612: The CAST-256 Encryption Algorithm, Jun 1999. http://www.cis.ohiostate.edu/htbin/rfc/rfc2612.html.

[15] H. Aljifri. IP Traceback: A New Denial-of-Service Deterrent? *IEEE Security and Privacy*, (24-31), 2003.

[16] S. H. Bass. Spoofed IP Address Distributed Denial of Service Attacks: Defense-in-Depth, Aug 2001. SANS Institute 2001, As part

of the Information Security Reading Room. `http://www.sans.org/rr/papers/60/469.pdf`.

[17] A. Belenky and N. Ansari. IP Traceback with Deterministic Packet Marking, Apr 2003. IEEE COMMUNICATIONS LETTERS.

[18] A. Belenky and N. Ansari. On IP Traceback. *IEEE Communications Magazine*, (7):142–153, Jul 2003.

[19] S. M. Bellovin. ICMP Traceback Messages, Mar 2000. Internet Draft: draft-bellovin-itrace-00.txt.

[20] H. Burch and B. Cheswick. Tracing Anonymous Packets to Their Approximate Source. Unpublished report, Dec. 1999.

[21] R. K. C. Chang. Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial. *IEEE Communications Magazine*, pages 42–51, Oct. 2002.

[22] Z. L. Chen and M. C. Lee. An IP Traceback Technique against Denial-of-Service Attacks. *19th Annual Computer Security Applications Conference (ACSAC 2003)*, (96-105), Dec 2003.

[23] K. Choi and H. Dai. A Marking Scheme Using Huffman Codes for IP Traceback. *Proceedings of 7th IEEE International Symposium on Parallel Architecture, Algotithms and Networks, 2004*, pages 421–428, May 2004.

[24] D. Dean, M. Franklin, and A. Stubblefield. An Algebraic Approach to IP Traceback. *ACM Transactions on Information and System Security (TISSEC)*, 5, May 2002.

[25] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queueing Algorithm. In *ACM SIGCOMM*, Aug 1989.

[26] R. Farrow. Spoofing Source Addresses. Information Network Security. `http://www.spirit.com/Network/net0300.html`.

[27] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing, May 2000. RFC 2827.

[28] Floyd and V. Jacobson. Random Early Detection gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4)(pp. 97-413), Aug 1993.

[29] S. Floyd and V. Jacobson. Link-sharing and Resource Management Models for Packet Networks. *IEEE/ACM Transactions on Networking*, 3(4):356–389, Aug 1995.

[30] M. T. Goodrich. Efficient Packet Marking for Large-Scale IP Traceback. In *CCS'02*, Washington, DC, USA, Nov 2002.

[31] K. J. Houle and G. M. Weaver. Trends in Denial of Service Attack Technology. In *Technical report from CERT Coordination Center*, Oct 2001.

[32] A. Hussain, J. Heidemann, and C. Papadopoulos. A Framework for Classifying Denial of Service Attacks. *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 99–110, Aug 2003.

[33] J. Ioannidis and S. M. Bellovin. Implementing Pushback: Router-based Defense against DDoS Attacks. In *Proceedings of Network*

and Distributed System Security Symposium, the Internet Society, 2002.

[34] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-hashing for Message Authentication. *Internet RFC 2104*, Feb 1997.

[35] V. Kuznetsov, A. Simkin, and H. Sandström. An Evaluation of Different IP Traceback Approaches. `http://www.sm.luth.se/csee/csn/publications/ip_traceback.pdf`, 2002.

[36] M. LaMonica. Microsoft releases anti-slammer tools, Feb 2003. `http://zdnet.com.com/2100-1105-983603.html`.

[37] F. Lau, S. Rubin, M. Smith, and L. Trajkovic. Distributed Denial of Service Attacks. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, Nashville, TN USA, Oct 2000.

[38] H. F. Lipson. Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy Issues, Nov 2002. CERT Coordination Center.

[39] J. Mirković, G. Prier, and P. Reiher. Source-End DDoS Defense. *Proceedings of Second IEEE International Symposium on Network Computing and Applications (NCA'03)*, pages 171–178, Apr 2003.

[40] J. Mirković and P. Reiher. A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. *ACM SIGCOMM Computer Communication Review*, 34, Apr 2004.

[41] K. Park and H. Lee. On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internet. In *SIGCOMM'01*, San Diego, California, USA, Aug 2001.

[42] V. Paxson. End-to-end Internet Path Dynamics. *ACM Transactions on Networking*, 7(3):277–292, 1999.

[43] V. Paxson. An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks. *ACM SIGCOMM Computer Communication Review*, 31, July 2001.

[44] A. Perrig, R. Canetti, D. Song, and D. Tygar. Efficient and Secure Source Authentication for Multicast. *Proceedings of Network and Distributed System Security Symposium (NDSS01)*, Feb 2001.

[45] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Pactical Network Support for IP Traceback. *IEEE/ACM Transactions on Networking (TON)*, 9(3):226–237, Jun 2001.

[46] C. Schuba, I. Krsul, M. Kuhn, E. Spafford, A. Sundaram, and D.Zamboni. Analysis of Denial of Service Attacks on TCP. *Proceedings of IEEE Symposium on Security and Privacy*, page 208, 1997.

[47] C. Shannon, D. Moore, and K. C. Claffy. Beyond Folklore: Observations on Fragmented Traffic. *IEEE/ACM Transactions on Networking (TON)*, 10:709–720, Dec 2002.

[48] A. C. Snoeren, C. Partridge, L. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, and W. T. Strayer. Single-packet IP traceback. *IEEE/ACM Transactions on Networking (TON)*, 10:721–734, Dec 2002.

[49] D. X. Song and A. Perrig. Advanced and Authenticated Marking Schemes for IP Traceback. *Proceedings of the IEEE Infocom*

conference, Twentieth Annual Joint Conference of the IEEE Computer and Communication Societies, 2:878–886, Apr 2001.

[50] W. Stayer, C. Jones, F. Tchakountio, and A. Snoeren. SPIE Demonstration: Single Packet Traceback. *DARPA Information Survivability Conference and Exposition 2003*, 2:106–107, Apr 2003.

[51] W. Stayer, C. Jones, F. Tchakountio, A. Snoeren, B. Schwartz, R. Clements, M. Condell, and C. Partridge. Traceback of Single IP Packets Using SPIE. *DARPA Information Survivability Conference and Exposition 2003*, 2:266–270, Apr 2003.

[52] I. Stoica and H. Zhang. Providing Guaranteed Services Without Per flow Management. *ACM SIGCOMM Computer Communication Review, Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 29:81–94, Aug 1999.

[53] M. Sung and J. Xu. IP Traceback-based Intelligent Packet Filtering: A Novel Technique for Defending Against Internet DDoS Attacks. *IEEE Transactions on Parallel and Distributed Systems*, 14(9):861–872, Sept 2003.

[54] M. Tanase. IP Spoofing: An Introduction, Mar 2003. `http://www.securityfocus.com/infocus/1674`.

[55] A. S. Tanenbaum. *Computer Networks*, chapter 4. Prentice-Hall, 2002.

[56] B. Wang and H. Schulzrinne. Analysis of Denial-of-Service Attacks on Denial-of-Service Defensive Measures. In *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, Dec 2003.

[57] B. T. Wang and H. Schulzrinne. A Robust Pakcet-Filtering Method for High-bandwidth Aggregates. *Proceedings of Canadian Conference on Electrical and Computer Engineering, 2004*, 2:905–908, May 2004.