Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems

TAO PENG, CHRISTOPHER LECKIE, and KOTAGIRI RAMAMOHANARAO

Department of Computer Science and Software Engineering, The University of Melbourne, Australia

This article presents a survey of denial of service attacks and the methods that have been proposed for defense against these attacks. In this survey, we analyze the design decisions in the Internet that have created the potential for denial of service attacks. We review the state-of-art mechanisms for defending against denial of service attacks, compare the strengths and weaknesses of each proposal, and discuss potential countermeasures against each defense mechanism. We conclude by highlighting opportunities for an integrated solution to solve the problem of distributed denial of service attacks.

Categories and Subject Descriptors: C.2.0 [**Computer-Communication Networks**]: General—*Security and protection (e.g., firewalls)*; C.2.3 [**Computer-Communication Network**]—*Network operation*

General Terms: Reliability, Security

Additional Key Words and Phrases: Botnet, bandwidth attack, DNS reflector attack, DoS, DDoS, Internet security, IP spoofing, IP traceback, IRC, resource management, SYN flood, VoIP security

ACM Reference Format:

Peng, T., Leckie, C., and Ramamohanarao, K. 2007. Survey of network-based defense mechanisms countering the DoS and DDoS problems. ACM Comput. Surv. 39, 1, Article 3 (April 2007), 42 pages DOI = 10.1145/1216370.1216373 http://doi.acm.org/10.1145/1216370.1216373

1. INTRODUCTION

The Internet was originally designed for openness and scalability. The infrastructure is certainly working as envisioned by that yardstick. However, the price of this success has been poor security. For example, the Internet Protocol (IP) was designed to support ease of attachment of hosts to networks, and provides little support for verifying the contents of IP packet header fields [Clark 1988]. This makes it possible to fake the source address of packets, and hence difficult to identify the source of traffic. Moreover, there is no inherent support in the IP layer to check whether a source is authorized to access a service. Packets are delivered to their destination, and the server at the destination must decide whether to accept and service these packets. While defenses such as firewalls can be added to protect servers, a key challenge for

This work was supported by the Australian Research Council.

Authors' addresses: Department of Computer Science and Software Engineering, ICT Building, 111 Barry Street, The University of Melbourne, Parkville VIC 3052, Australia; email: {tpeng,caleckie}@csse.unimelb.edu.au,kotagiri@unimelb.edu.au.

defense is how to discriminate legitimate requests for service from malicious access attempts.

If it is easier for sources to generate service requests than it is for a server to check the validity of those requests, then it is difficult to protect the server from malicious requests that waste the resources of the server. This creates the opportunity for a class of attack known as a *denial of service attack*.

A denial of service (DoS) attack aims to deny access by legitimate users to shared services or resources [Gligor 1984]. This can occur in a wide variety of contexts, from operating systems [Gligor 1984] to network-based services [Needham 1994]. On the Internet, a DoS attack aims to disrupt the service provided by a network or server. It can be launched in two forms [Hussain et al. 2003]. The first form aims to crash a system by sending one or more carefully crafted packets that exploit a software vulnerability in the target system. For example, the "ping-of-death" attack sends a large International Control Message Protocol (ICMP) ping packet that is fragmented into multiple datagrams to a target system, which can cause certain operating systems to crash, freeze, or reboot due to buffer overflow [CERT 1996]. The second form is to use massive volumes of useless traffic to occupy all the resources that could service legitimate traffic. While it is possible to prevent the first form of attack by patching known vulnerabilities, the second form of attack cannot be so easily prevented. The targets can be attacked simply because they are connected to the public Internet. In the rest of this article, unless otherwise stated, when we use the term *DoS attack*, we are referring to the second form of attack that uses massive volumes of useless traffic.

When the traffic of a DoS attack comes from multiple sources, it is called a *distributed denial of service (DDoS) attack*. By using multiple attack sources, the power of a DDoS attack is amplified and the problem of defense is made more complicated. The impact of DDoS attacks can vary from minor inconvenience to users of a Web site to serious financial losses for companies that rely on their online availability to do business. On February 9, 2000, Yahoo, eBay, Amazon.com, E*Trade, ZDnet, Buy.com, the FBI, and several other Web sites fell victim to DDoS attacks resulting in substantial damage and inconvenience [Garber 2000]. From December 2005 to January 2006, 1,500 separate IP addresses were victims of DDoS attacks, with some attacks using traffic rates as high as 10 Gb/s [Scalzo 2006; Vaughn and Evron 2006].

More importantly, traditional operations in essential services, such as banking, transportation, power, health, and defense, are being progressively replaced by cheaper, more efficient Internet-based applications. Internet-based attacks can be launched anywhere in the world, and unfortunately any Internet-based service is a potential target for these attacks. As emergency and essential services become reliant on the Internet as part of their communication infrastructure, the consequences of DDoS attacks could even become life-threatening. Hence, it is crucial to deter, or otherwise minimize, the damage caused by DDoS attacks.

This survey presents techniques for defending against DoS and DDoS attacks, and evaluates their effectiveness against a variety of DoS and DDoS attacks. Earlier surveys provide an introduction to DDoS attacks. For example, Chang [2002] provided a survey on DDoS attack defense in terms of attack detection and packet filtering, and addressed some of the technical challenges posed by those tasks. Mirkovic and Reiher [2004] also presented taxonomies for classifying attacks and defenses. What is lacking in the literature is a detailed comparison of the relative strengths and weaknesses of each defense proposal, and how they can be integrated to provide a comprehensive solution to DDoS attacks. In this extensive survey, we address these shortcomings by (1) describing the inherent design features of the Internet which created the potential for different types of DDoS attacks, (2) characterizing the impact of different types of DDoS attacks, and (3) providing an in-depth study of

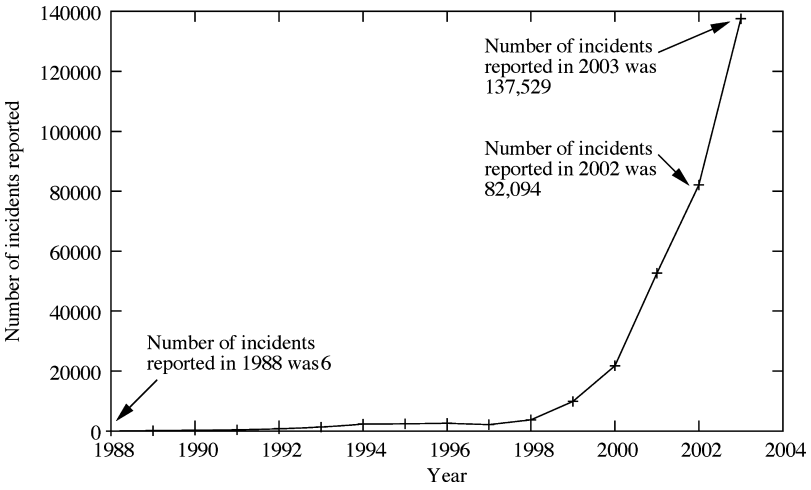

Fig. 1. The number of Internet security incidents reported from 1988 to 2003.

proposed DDoS defense mechanisms, as well as potential countermeasures against these defenses.

In this survey, our main contributions are twofold. First, we provide a detailed study of the challenges posed by the DoS and DDoS attack problem and their root causes. Second, we provide a thorough analysis of the strengths and weaknesses of state-of-art DoS and DDoS defense mechanisms, and highlight opportunities for an integrated solution to solve the DDoS attack problem.

The rest of the article is organized as follows. Section 2 describes the basic operation of DoS and DDoS attacks, as well as their prevalence in the Internet. Section 3 examines the fundamental design features of the Internet that have enabled bandwidth attacks to occur. Section 4 includes a description of different types of bandwidth attacks. Section 5 includes a detailed review of the proposed solutions to DoS and DDoS attacks. Section 6 highlights the opportunities for an integrated solution to DDoS attacks. Section 7 provides a description of the remaining threats and open issues that remain to be addressed in solving DoS and DDoS problems, and concludes the article.

2. BACKGROUND AND PROBLEM STATEMENT

2.1. Growth in Internet Attacks

The original aim of the Internet was to provide an open and scalable network among research and educational communities [Lipson 2002]. In this environment, security issues were less of a concern. The occurrence of the Morris Worm [Rochlis and Eichin 1989] in 1988 marked the first major computer security incident on the Internet. However, at that time, the world was not so dependent on the Internet as it is now.

Unfortunately, with the rapid growth of the Internet over the past decade, the number of attacks on the Internet has also increased rapidly. According to CERT, the number of reported Internet security incidents has jumped from six in 1988 to 82,094 in 2002, and to 137,529 in 2003 [CERT 2006]. Due to the excessive number of security incidents, CERT has decided not to publish the number of incidents reported since 2004. The growth in the number of incidents reported between 1998 to 2003 is shown in Figure 1.

In 2005, the Computer Security Institute (CSI) and the FBI released a survey on the prevalence and character of computer crime based on the responses from 700 security

Table I. Percentage of CSI/FBI Cybersecurity Survey Responders Who Observed a DoS Attack During 1999–2005

Year	Percentage of Respondents Observing DoS Attack
1999	30
2000	27
2001	36
2002	40
2003	42
2004	39
2005	32

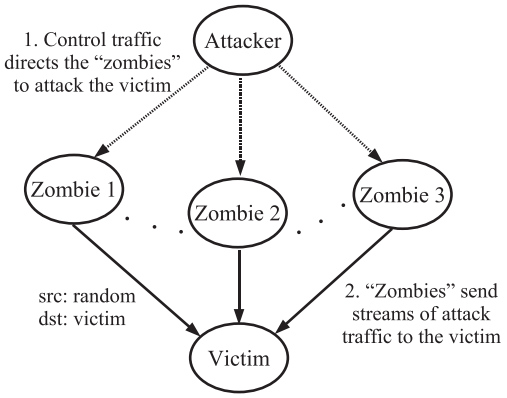

Fig. 2. Structure of a typical DDoS attack (based on Paxson [2001]).

analysts and Chief Security Officers (CSO) from mid-to-large firms in the U.S. [Gordon et al. 2005]. Table I lists the percentage of the participants who were targeted by a DoS attack between 1999 and 2005. We can see that a considerable proportion of respondents have suffered from DoS attacks.

2.2. DoS and DDoS Attacks

DoS attacks generally achieve their goal by sending large volumes of packets that occupy a significant proportion of the available bandwidth. Hence, DoS attacks are also called *bandwidth attacks*. The aim of a bandwidth attack is to consume critical resources in a network service. Possible target resources may include CPU capacity in a server, stack space in network protocol software, or Internet link capacity. By exhausting these critical resources, the attacker can prevent legitimate users from accessing the service.

A crucial feature of bandwidth attacks is that their strength lies in the volume rather than the content of the attack traffic. This has two major implications:

(1) Attackers can send a variety of packets. The attack traffic can be made arbitrarily similar to legitimate traffic, which greatly complicates defense.
(2) The volume of traffic must be large enough to consume the target's resources. The attacker usually has to control more than one computer to generate the attack traffic. Bandwidth attacks are therefore commonly DDoS attacks.

A typical DDoS attack contains two stages as shown in Figure 2. The first stage is to compromise vulnerable systems that are available in the Internet and install attack tools in these compromised systems. This is known as turning the computers into

Table II. Comparison Between Bandwidth Attacks and Flash Crowds

	Bandwidth Attack	Flash Crowd
Network impact	Congested	Congested
Server impact	Overloaded	Overloaded
Traffic	Malicious	Genuine
Response to traffic control	Unresponsive	Responsive
Traffic type	Any	Mostly Web
Number of flows	Any	Large number of flows
Predictability	Unpredictable	Mostly predictable

"zombies." In the second stage, the attacker sends an *attack command* to the "zombies" through a secure channel to launch a bandwidth attack against the targeted victim(s) [Dietrich et al. 2000]. Note that the packets in the attack traffic may use a fake source IP address in order to make it harder for the target of the attack to identify the source of the attack traffic.

The number of coordinated sources in a DDoS attack can vary from dozens to hundreds or more than 100,000 compromised machines. A prominent example is that during the spread of the "Code Red" worm [CERT 2001], over 300,000 "zombie" machines were compromised to launch a denial of service attack on the White House Web site [Evans and Larochelle 2002].

There are several unique features of DDoS attacks that make effective defenses extremely difficult to design. First, the traffic volume generated by a DDoS attack can exceed 10 Gb/s [ARBOR 2005], which can occupy the capacities of most corporate Internet links, and exceed the throughput of many network security devices. Second, the attack packets come from many sources and can be geographically distributed, which makes IP source traceback extremely difficult. Third, the traffic from each attack source of a DDoS attack does not need to be conspicuous to constitute a powerful attack. Therefore, DDoS attack traffic will tend to appear "legitimate," which makes it extremely difficult to filter attack traffic without disrupting legitimate traffic. In particular, this type of DDoS attack can appear similar to a *flash crowd*, which occurs when a large number of legitimate users access a server at the same time. A comparison between bandwidth attacks and flash crowds is shown in Table II.

Generally, a large number of traffic sources are required to launch an effective DDoS attack. Unfortunately, recruiting and engaging a large army of compromised machines has become technically trivial as many automated DDoS attack tools are available via hacker Web pages or chat rooms. In the following section we discuss how attackers can obtain these traffic sources.

2.3. Botnets

These days, online computers, especially those with a high-bandwidth connection, have become a desirable target for attackers. Attackers can gain control of these computers via direct or indirect attacks. Direct attacks refer to sending packets containing a malicious payload that exploits a vulnerable computer, for example, an unpatched Windows home PC. Generally, these attacks are conducted via automated software so that the number of compromised computers can be maximized in a short period. The requirement for launching direct attacks is that publicly available services on the targeted computers contain software vulnerabilities. For example, the Blaster Worm spread by exploiting a vulnerability in the Remote Procedure Call (RPC) service [CERT 2003], which allowed malicious code to be executed in the remote host. Unfortunately, this kind of vulnerability occurs frequently and has been increasing. According to CERT [2006] statistics as shown in Figure 3, the number of vulnerabilities reported in 2005 was 5,990, which is 35 times the number in 1995.

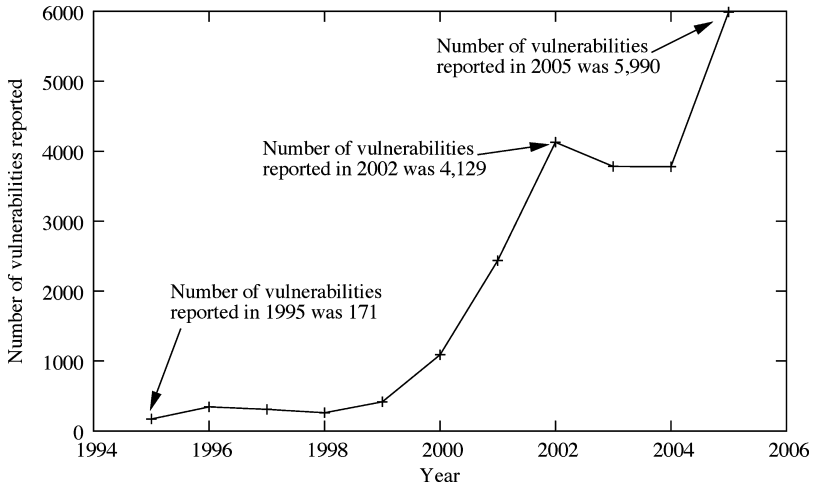

Fig. 3. The number of vulnerabilities reported each year according to CERT.

In contrast, indirect attacks can exploit insecure actions that may be performed by users. These attacks generally require human interaction. For example, users are decoyed to open a malicious HTML file that exploits a vulnerability in Microsoft Internet Explorer, or to install free software with malicious software embedded.

Once these attackers have compromised a computer, they install a "bot," which is another name for a "zombie." The term *bot* (derived from the word *robot*) is used in industry jargon to describe an automaton or automated process in both the real world and the computer world. A bot generally supports a communication channel with the attacker, as well as the ability to execute particular tasks, for example, launching DoS attacks, according to the attacker's instructions.

2.3.1. Botnet Communication. A common way for attackers to control the bots is to use Internet Relay Chat (IRC) channels. IRC is a form of real-time communication over the Internet. It is mainly designed for group (many-to-many) communication in discussion forums called *channels*, but also allows one-to-one communication. Once installed in the compromised computers, the bot will automatically join a specific IRC channel on an IRC server, and wait there for further instructions. These compromised computers that can be managed by the attacker through the IRC channel are called a *botnet*.

In fact, IRC channels are not the best solution for an attacker to communicate with the bots in terms of efficiency and robustness. With increases in botnet size, IRC channels are likely to be congested. Moreover, relying on IRC servers for communication creates a single-point-of-failure. In fact, removing the IRC server used by the botnet has proved to be an effective DoS attack defense approach. There are two main reasons that explain why IRC-based bots are so popular. First, IRC servers are freely accessible to the public, and they are easy to set up. Second, many attackers are familiar with IRC communication [Honeynet 2005]. However, future botnets can use non-IRC-based communication, for example, by making use of decentralized and encrypted peer-to-peer communication.

2.3.2. Botnet Function. Botnets can be used for a wide variety of purposes. Nevertheless, DDoS attack capability is a common feature of botnet software [Honeynet 2005].

Generally, each type of botnet software contains a set of flooding mechanisms, such as SYN flood, ICMP flood, and HTTP flood (described in Section 4). A set of sophisticated configuration commands are provided to control the attack parameters, such as sending rate and packet size. Another important feature of botnets is the ability to *update* software from a remote server. In this way, an attacker can fix existing software bugs and add new functions into the botnet software. For example, an attacker can instruct all bots to download a new type of flooding mechanism to defeat a DDoS protection system. Hence, the botnet owner has the capability to design a specific attack for a particular target, and maximize the similarity between attack traffic and legitimate traffic. As noted by Davis [2006], attackers are now using open source software development methodologies to improve the effectiveness of botnet software by making it easier for multiple contributors to develop and test new software features.

3. EFFECTS OF THE DESIGN PRINCIPLES OF THE INTERNET ARCHITECTURE ON SECURITY

In this section, we will revisit the original design principles of the Internet and discuss their implications in terms of DoS attacks.

3.1. Resource Sharing

The Internet is based on packet-switching, unlike its counterpart, the public telephone network, which is based on circuit-switching. For circuit-switched networks, each service (e.g., a phone call) will be allocated a separate channel until the end of the service. A user's service will not be disrupted by other users' behavior. In contrast, IP networks were originally designed to provide a best-effort, packet-switched service, where users share all the resources, and one user's service can be disturbed by other users' behavior. By occupying most of the shared resources, bandwidth attacks can disrupt service for legitimate users. This interuser dependency is a fundamental factor that enables DoS attacks to occur [Gligor 1984].

3.2. Simple Core and Complex Edge

One of the design principles is that the Internet should keep the core networks simple and push any complexity into the end hosts [Mirkovic et al. 2005]. This means that intermediate routers, especially core routers, only need to deliver IP packets without needing to understand services above the network layer. Most changes to the Internet are implemented at the end hosts. This encourages the development of new protocols and new applications.

However, this also means that core routers do not have resources to implement sophisticated applications, for example, mandatory authentication schemes. The lack of authentication at the network layer leads to a serious problem, known as *IP spoofing*. IP spoofing refers to creating an IP packet containing fake information. *IP source address spoofing* occurs when an IP packet is generated using a different source IP address than the actual address that is assigned to the source computer. Without an integrity check for each IP packet, attackers can spoof any field of an IP packet and inject it into the Internet. For the same reason, routers generally do not have packet-tracing functions, for example, keeping all previous connection records. In practice, this cannot be done due to the huge amount of traffic that needs to be stored. Therefore, once an IP packet is received by the victim, there is no way to authenticate whether the packet actually comes from where it claims to be coming from.

3.3. Multipath Routing

Another design principle is that packets can travel on any path between the source and the destination. This makes the Internet extremely robust in comparison to traditional telephone networks. However, it makes traceability of packets extremely difficult. IP packets are forwarded based on their destination address, rather than a predefined path. Many factors, such as delay on a link, can contribute to the changeability of the path a packet is traveling. Hence, the set of IP addresses that appear at a given interface of a router can be highly variable. If a router receives a packet from a source that has not been seen before, then the router has no way of knowing whether this is a spoofed packet, or a legitimate packet that is following a new route as a result of congestion or failure elsewhere in the network. While this flexibility helps make the Internet robust, it also makes IP address authentication difficult.

3.4. Fast Core Networks and Slow Edge Networks

Another design principle for the Internet is to provision links according to their usage. Core networks need to accommodate heavy traffic from many sources to many destinations. Hence, these links have high capacity, for example, OC-192 (10 Gb/s) links are common for tier-1 ISPs. In contrast, an edge network only needs to support its end users, which requires less capacity. This certainly maximizes the utilization of links and minimizes their cost. However, a drawback is that traffic from high-capacity core links can overwhelm the low-capacity edge links if many sources want to talk to a single destination. This is exactly the case for a DDoS attack [Mirkovic et al. 2005].

3.5. Decentralized Internet Management

The Internet is an aggregation of numerous networks, interconnected to provide global access to the end users [Mirkovic et al. 2005]. There is no central authority or management hierarchy in the Internet and each interconnected network is managed locally. Thanks to this management approach, the Internet has grown rapidly. However, this has also provided attackers with easy-to-access resources, and made cooperative DDoS attack defense across multiple subnetworks extremely difficult.

Many DDoS defense approaches need to be deployed at numerous locations to be effective. However, due to the lack of central control, it is extremely difficult to enforce global deployment in the Internet, which makes highly distributed solutions unattractive. On the other hand, the distributed nature of DDoS attacks renders a single-point solution ineffective. Moreover, due to privacy and other commercial concerns, network service providers generally are reluctant to provide detailed information about the traffic patterns within their networks and cooperate in tracing attack sources. More importantly, there is no automated support for tracing attack sources. Once a source tracing request is authenticated and authorized, it has to be enforced via human intervention. The whole process is expensive and slow, which is particularly ineffective for tracing attacks that only last a short period of time [Lipson 2002].

4. METHODS OF ATTACK

There are two major impacts of bandwidth attacks. This first is the consumption of the host's resources. Generally, the victim could be a Web server or proxy connected to the Internet. The victim has limited resources to process the incoming packets. When the traffic load becomes high, the victim will drop packets to inform senders, which consist of both legitimate users and attack sources, to reduce their sending rates. Legitimate users will slow down their sending rates while the attack sources will maintain or increase

their sending rates. Eventually, the victim's resources, such as CPU and memory, will be exhausted and the victim will be unable to service legitimate traffic.

The second impact is the consumption of network bandwidth, which is more disruptive than the first. If the malicious flows are able to dominate the communication links that lead to the victim, then the legitimate flows will be blocked. Therefore, not only the intended victim of the attack is disabled, but also any system that relies on the communication links of the attack path. Although a congested router can control the traffic flow by dropping packets, legitimate traffic will also be discarded if there is no clear mechanism to differentiate legitimate traffic from attack traffic.

We define the *attack power* as the level of resources consumed at the victim by the attack. Generally, the attack power consists of two parameters. The first parameter is the traffic volume, which can be represented by the number of packets in a given period. The second parameter is the level of resources consumed per packet, which can be represented by CPU time or memory needed to process the packet.

To help readers gain a deeper understanding of bandwidth attacks, we depict some classic attacks as well as some new and emerging ones in the following sections. We broadly classify bandwidth attacks according to the way the attack power is magnified. The first category is attacks that take advantage of the Internet protocols. The second category is attacks that aim at a particular application. The third category is attacks that use innocent third parties to distribute or amplify attack traffic to the target. The fourth category is attacks that disrupt the Internet infrastructure. In practice, a real attack can belong to multiple categories at the same time.

4.1. Protocol-Based Bandwidth Attacks

A protocol-based bandwidth attack can normally be launched effectively from a single attack source. Its attack power is based on specific weaknesses of the Internet protocols. Two classic examples of such attacks, namely, SYN floods and ICMP floods, are described in this section.

4.1.1. SYN Flood. In order to describe the SYN flood attack, we first need to define several aspects of TCP connections. We define the client as the one who initiates the TCP connection, and the server as the one who receives the connection request. At the beginning of each TCP connection, the client will negotiate with the server to set up a connection, which is called a three-way handshake. First, the client will send a SYN packet to the server, requesting a connection. Then the server will respond to the connection request using a SYN-ACK packet, and store the request information in the memory stack. Under BSD-style network software [Wright and Stevens 1995], three memory structures are allocated once a SYN packet is received, that is, *socket*, *inpcb*, and *tcpcb*. These data structures are used to store the details of the requested TCP connection, and their combined size for a single TCP connection may typically exceed 280 B [Schuba et al. 1997]. At this point, a connection is in a half-open state, called the *SYN_RECVD state* [Wright and Stevens 1995].

To prevent the system from depleting its memory, each operating system will limit the number of concurrent TCP connections in the SYN_RECVD state. After receiving the SYN-ACK packet, the client will confirm the request using an ACK packet. When the server receives the ACK packet, it checks the memory stack to see whether this packet is used to confirm an existing request. If it is, that TCP connection is moved from the SYN_RECVD state to the ESTABLISHED state. After this, the client and server have finished the three-way handshake and can start data transfer. Another way to remove a connection in the SYN_RECVD state is to either send a RST packet or wait until its timer expires.

The SYN flood attack exploits a vulnerability of the TCP three-way handshake, namely, that a server needs to allocate a large data structure for any incoming SYN packet regardless of its authenticity. During SYN flood attacks, the attacker sends SYN packets with source IP addresses that do not exist or are not in use. During the three-way handshake, when the server puts the request information into the memory stack, it will wait for the confirmation from the client that sends the request. While the request is waiting to be confirmed, it will remain in the memory stack. Since the source IP addresses used in SYN flood attacks can be nonexistent, the server will not receive confirmation packets for requests created by the SYN flood attack. Each half-open connection will remain on the memory stack until it times out. More and more requests will accumulate and fill up the memory stack. Therefore, no new request, including legitimate requests, can be processed and the services of the system are disabled. Generally, the space for the memory stack allocated by the operating system is small, and even a small scale SYN flood attack can be disruptive. On the other hand, SYN floods can be also launched from compromised machines using genuine source IP addresses given these compromised machines are configured to ignore the SYN/ACK packets from the target.

SYN floods remain one of the most powerful flooding methods. Mechanisms to defend against SYN flood attack are discussed in Section 5.4.1.

4.1.2. ICMP Flood. The Internet Control Message Protocol (ICMP) is based on the IP protocol and is used to diagnose network status. An ICMP flood is a type of bandwidth attack that uses ICMP packets.

On IP networks, a packet can be directed to an individual machine or broadcast to an entire network. When a packet is sent to an IP broadcast address from a machine on the local network, that packet is delivered to all machines on that network. When a packet is sent to that IP broadcast address from a machine outside the local network, it is broadcast to all machines on the target network (as long as routers are configured to pass along that traffic).

IP broadcast addresses are usually network addresses with the host portion of the address having all one bits. For example, the IP broadcast address for the network 10.*.*.* is 10.255.255.255, and for the network 10.50.*.* is 10.50.255.255. Network addresses with all zeros in the host portion, such as 10.50.0.0, can also produce a broadcast response.

The "smurf" attack is a type of ICMP flood, where attackers use ICMP echo request packets directed to IP broadcast addresses from remote locations to generate denial of service attacks. There are three parties in these attacks: the attacker, the intermediary, and the victim (note that the intermediary can also be a victim) [CERT 1998]. Figure 4 gives an example of the *smurf attack*. First, the attacker sends one ICMP echo request packet to the network broadcast address and the request is forwarded to all the hosts within the intermediary network. Second, all of the hosts within the intermediary network send the ICMP echo replies to flood the victim. Solutions to the smurf attack are discussed in CERT [1998], which include disabling the IP-directed broadcast service at the intermediary network. Nowadays, smurf attacks are quite rare in the Internet.

4.2. Application-Based Bandwidth Attacks

Another way to amplify attack power is to force the target to execute expensive operations. For example, many Web sites provide search engines to allow users to find a particular Web page. An attacker can exploit this application by sending a large number of queries to a Web site's search engine. In this way, the Web site is forced to

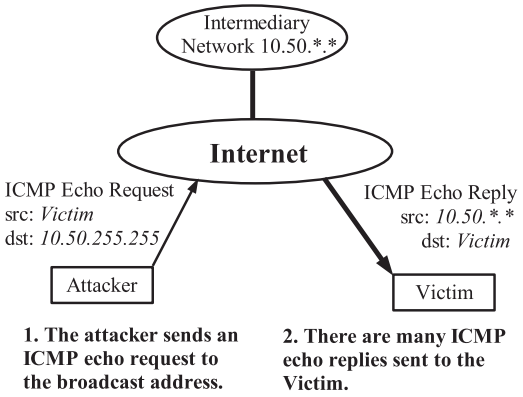

Fig. 4. A smurf attack, using an intermediary network to amplify ICMP echo requests.

perform CPU and memory-intensive database operations and leave few resources to serve legitimate users. We call this type of attack an *application-based bandwidth attack*, which aims to take advantage of the disproportionally large resource consumption at the server. In this section we will depict attacks that target two important Internet applications, namely, the World Wide Web and Voice over IP.

4.2.1. HTTP Flood. The World Wide Web (WWW) is one of the most popular applications currently running on the Internet and has driven the rapid growth of the Internet [Wang 1999]. WWW applications generally use the Hypertext Transfer Protocol (HTTP) over TCP port 80. Thanks to this popularity, most firewalls on the Internet will leave TCP port 80 open to allow HTTP traffic to pass. Unfortunately, the ubiquity of WWW applications has also made HTTP a prime target for attackers.

Generally, an HTTP flood refers to an attack that bombards Web servers with HTTP requests. According to a recent study [Honeynet 2005], HTTP floods have become a common feature in most botnet software. To send an HTTP request, a valid TCP connection has to be established, which requires a genuine IP address. Attackers can achieve this by using a bot's IP address. Moreover, attackers can craft the HTTP requests in different ways in order to either maximize the attack power or avoid detection. For example, an attacker can instruct the botnet to send HTTP requests to download a large file from the target. The target then has to read the file from hard disk, store it in memory, load it into packets, and then send the packets back to the botnet. Hence, a simple HTTP request can incur significant resource consumption in the CPU, memory, input/output devices, and outbound Internet link.

However, the behavior of the HTTP requests of the previous example can be conspicuous. Repetitive requests to a large file can be detected and hence blocked. To better mimic legitimate traffic, attackers can instruct the botnet to send an HTTP request to the target Web site, then parse the replies and follow the links recursively. In this way, the HTTP requests from the attacker are very close to normal Web traffic, which makes it extremely difficult to filter this type of HTTP flood.

4.2.2. SIP Flood. In the past few years, the deployment of Voice over IP (VoIP) telephony has become popular thanks to its low cost. A widely supported open standard for call setup in VoIP is the Session Initiation Protocol (SIP) [Rosenberg et al. 2002]. Generally, SIP proxy servers require public Internet access in order to accept call setup

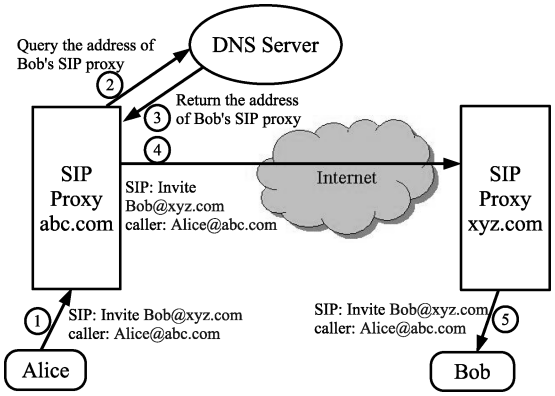

Fig. 5. SIP Invite packets.

requests from any VoIP client. Moreover, to achieve scalability, SIP is typically implemented on top of UDP in order to be stateless.

Figure 5 illustrates the process of call setup using SIP. For simplicity, some details of the SIP signaling process have been intentionally omitted. As shown in Figure 5, if Alice wants to talk to Bob, she will first send an Invite packet to Bob. Generally, this packet is sent to Alice's SIP proxy server, which will look up the address of Bob's SIP proxy server and send an Invite packet to that proxy. When Bob's SIP proxy receives the Invite packet, it will pass it to Bob's registered address and Bob's phone will ring. After this, either Bob picks up the phone to start the conversation or there is no answer. Interested readers can refer to RFC 3261 [Rosenberg et al. 2002] for details.

In one attack scenario, the attackers can flood the SIP proxy with many SIP Invite packets that have spoofed source IP addresses [Sisalem et al. 2005; Kuhn et al. 2005; Chen 2006]. To avoid any antispoofing mechanisms, the attackers can also launch the flood from a botnet using nonspoofed source IP addresses. There are two categories of victims in this attack scenario. The first category of victims are the SIP proxy servers. Not only will their server resources be depleted by processing the SIP Invite packets, but their network capacity will also be consumed by the SIP Invite flood. In either case, the SIP proxy server will be unable to provide VoIP service. The second category of victims are the call receivers. They will be overwhelmed by the forged VoIP calls, and will become nearly impossible to reach by legitimate callers.

4.3. Distributed Reflector Attacks

An important goal for attackers is to hide the true sources of their attack traffic. Figure 6 [Paxson 2001] illustrates another type of bandwidth attack called a *distributed reflector denial of service (DRDoS) attack*, which aims to obscure the sources of attack traffic by using third parties (routers or Web servers) to relay attack traffic to the victim. These innocent third parties are also called *reflectors*. Any machine that replies to an incoming packet can become a potential reflector. The *DRDoS attack* contains three stages. The first stage is very similar to the *typical DDoS attack* described in Section 2.2. However, in the second stage, after the attacker has gained control of a certain number of "zombies," instead of instructing the "zombies" to send attack traffic to the victims directly, the "zombies" are ordered to send to the third parties spoofed traffic with the victim's IP address as the source IP address. In the third stage, the third parties will then send the reply traffic to the victim, which constitutes a DDoS attack. This type of attack shut

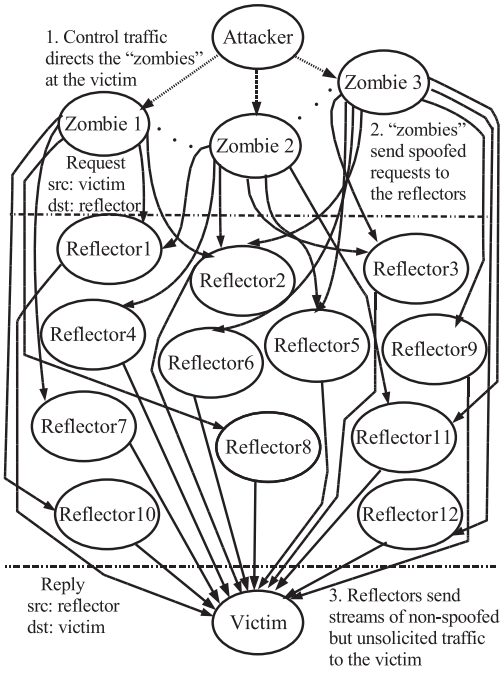

Fig. 6. Structure of a distributed reflector denial of service (DRDoS) attack (based on Paxson [2001]).

down `www.grc.com`, a security research Web site, in January 2002, and is considered to a be a potent and increasingly prevalent Internet attack [Gibson 2002].

In comparison to a traditional DDoS attack, the traffic from a DRDoS attack is further dispersed by using the third parties, which makes the attack traffic even more distributed and difficult to identify. Moreover, the source IP addresses of the attack traffic are from innocent third parties, which makes attack source traceback exteremely difficult. Finally, as noticed by Paxson [2001] and Gibson [2002], DRDoS attacks have the ability to amplify the attack traffic, which makes the attack even more potent. In the following section, we use a real-world example to show the serious threat posed by DRDoS attacks.

4.3.1. DNS Amplification Attacks. A particularly effective form of reflector attack makes use of the existing Domain Name System (DNS) [Mockapetris 1987a] servers. The role of the Domain Name System is to provide a distributed infrastructure to store and associate different types of *resource records (RR)* with Internet domain names, such as `unimelb.edu.au`. Relevant examples of resource records include type TXT RR, which allow an administrator to insert arbitrary text into a DNS record; type A RR, which map a host name into a 32-bit IP address; and type SOA[1] RR, which provide the name of the primary source of an Internet domain and other related information. One important function of DNS is to translate domain names into IP addresses. A recursive DNS server accepts a query and resolves a given domain name on behalf of the requester. Generally, a recursive name server will contact other authoritative names servers if necessary and eventually return the query response back to the requester [Mockapetris

[1]SOA stands for *start of authority.*

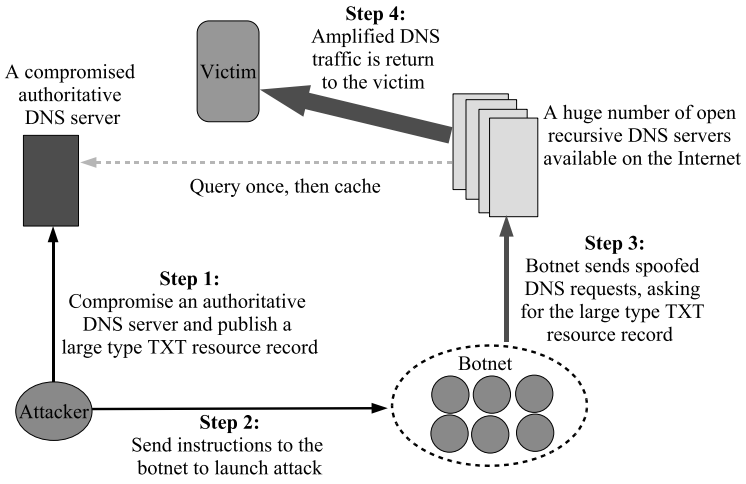

Fig. 7. An example of a DNS amplification attack.

1987b]. The sizes of the DNS query and query response are disproportional. Normally, a query response includes the original query and the answer, which means the query response packet is always larger than the query packet. Moreover, one query response can contain multiple types of RR, and some types of RR can be very large. For example, if a DNS name server receives a 60-B EDNS[2] query [Vixie 1999] containing a large buffer advertisement, its reply can include a 122-B type A resource record, a 4000-B type TXT resource record, and a 222-B type SOA resource record [Vaughn and Evron 2006]. This renders an amplification factor of 73.

Figure 7 illustrates an example of a DNS amplification attack that was observed in early 2006 [Scalzo 2006]. In this attack scenario, an attacker first compromises an authoritative DNS server and publishes a large (e.g., 4000 B) type TXT RR. Then the attacker instructs the botnet to send spoofed DNS requests with the victim's IP address to open DNS recursive servers, asking for the large TXT RR. Finally, the open DNS recursive servers resolve the query and return the amplified DNS responses back to the victim. In theory, 140 Mb/s initiating traffic from a botnet can result in a 10 Gb/s DNS flood to the victim. This gives the attacker an opportunity to generate a powerful DDoS attack from even a small botnet. Unfortunately, the opportunity to launch such an attack is widely available in the Internet. According to a survey conducted in 2005 [Measurement 2005], 75% of the 7.5 million external DNS servers allow recursive name service to arbitary queries. Moreover, attackers do not need to place their own large resource records to implement a successful DNS amplifcation attack. There are many well-known public resource records that are large [Scalzo 2006].

4.4. Infrastructure Attacks

An *infrastructure attack* aims to disable the services of critical components of the Internet. The result of an infrastructure attack is potentially catastrophic as the whole Internet may be affected. For example, DNS root servers provide information about the servers that are responsible for top-level domains, such as `.com`. They are indispensable elements to enable DNS to function. An infrastructure attack can tie up both the network and host resources of a DNS root server, disrupting all Internet services

[2]EDNS stands for *Extended DNS*.

that depend on these servers. On 21 October 2002, all 13 Internet DNS root servers were attacked simultaneously by coordinated distributed denial of service attacks. The attack lasted about 1 h and 15 min, and the attack volume was approximately 50 to 100 Mb/s (100 to 200 kpkts/s) per root name server, yielding a total attack volume of approximately 900 Mb/s (1.8 Mpkts/s) [Vixie et al. 2002; CAIDA 2006]. Thanks to the overprovisioning of host resources, all the root servers were reported to be able to answer all queries they received. However, some root servers were unreachable to many parts of the Internet or incurred longer response times to DNS queries due to attack-related congestion. Had the attacker increased the attack traffic rate or extended the attack time, more catastrophic damage would have been done to the overall Internet. A detailed analysis of attacks against DNS can be found in Cheung [2006].

Normally, critical network infrastructure is highly provisioned. Therefore, significant attack power is required to launch a successful infrastructure attack.[3] Given the scale of the potential impact of an infrastructure attack, global cooperation is essential for an effective defense.

4.5. Summary

In this section, we have presented a number of classic and recent bandwidth attacks. The purpose of this categorization has been to highlight the main features of each category of attack. It is important to note that these categories of attack are not mutually exclusive. In practice, an attack can use features of multiple categories. For example, the aforementioned DNS root server attack used SYN flood and ICMP floods as part of its arsenal [Vixie et al. 2002]. Network operators need to take that into consideration when designing their defenses.

5. EXISTING DoS ATTACK DEFENSE PROPOSALS

Generally, there are four broad categories of defense against DoS attacks: (1) *attack prevention*, (2) *attack detection*, (3) *attack source identification*, and (4) *attack reaction*. *Attack prevention* aims to stop attacks before they can reach their target. In the context of this survey, it refers to filtering spoofed packets close to or at the attack sources, which is one of the most effective defense approaches for DoS attacks that use spoofed traffic. *Attack detection* aims to detect DoS attacks when they occur. Attack detection is an important procedure to direct any further action. *Attack source identification* aims to locate the attack sources regardless of whether the source address field in each packet contains erroneous information. It is a crucial step to minimize the attack damage and provide deterrence to potential attackers. *Attack reaction* aims to eliminate or curtail the effects of an attack. It is the final step in defending against DoS attacks, and therefore determines the overall performance of the defense mechanism. The challenge for attack reaction is how to filter the attack traffic without disturbing legitimate traffic. Strictly speaking, DoS attacks include DDoS attacks. For the convenience of discussion, we consider DoS attacks to be attacks that are launched from a single or a few hosts (e.g., fewer than 10), and refer to DDoS attacks as attacks that are launched from many hosts (e.g., at least an order of magnitude more).

[3]In addition, as routers, especially core routers, act as bridges between Internet end users, any attack that disrupts the service of routers can constitute a successful infrastructure attack. Recently, it has been demonstrated [US-CERT 2005] that CISCO IOS, a widely used router operating system, has exploitable software vulnerabilities. Such attacks targeting routers need to be addressed by having a secure and hardened router operating system. As this type of attack does not need a large volume of packets to be effective, it is outside the scope of this survey article.

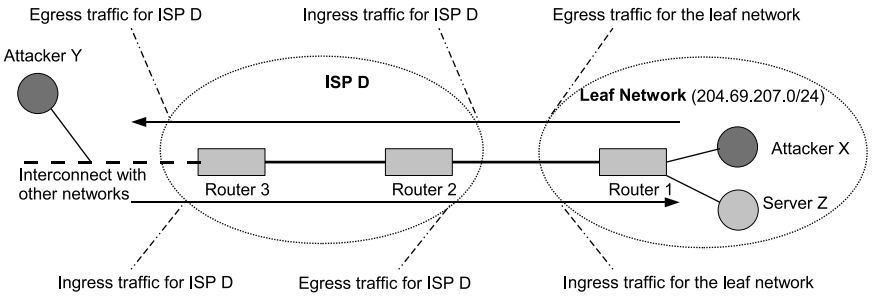

Fig. 8. An example of ingress/egress filtering based on RFC 2827.

5.1. Attack Prevention

Attack Prevention aims to stop attacks before they actually cause damage. This approach assumes the source address of attack traffic is spoofed, which is true in many situations since attackers need spoofed traffic to hide the real source of the attack traffic and exploit protocol vulnerabilities as discussed in Section 4.1. This approach normally comprises a variety of packet filtering schemes, which are deployed in routers. The packet filters are used to make sure only valid (nonspoofed) traffic can pass through. This greatly reduces the chance of DDoS attacks occurring.

However, it is not easy to specify a filtering rule that can differentiate spoofed traffic from legitimate traffic accurately. Moreover, some types of filtering schemes require wide deployment to be effective. Unfortunately, the Internet is an open community without central administration, which makes prevention a taxing and daunting task.

5.1.1. Ingress/Egress Filtering. *Ingress filtering* means filtering the traffic coming into your local network, and *egress filtering* means filtering the traffic leaving your local network. When describing ingress/egress filtering, a reference point is needed to avoid confusion. We use an example from the original ingress filtering proposal [Ferguson and Senie 2000] to illustrate these two concepts.

5.1.1.1. Analysis of Ingress/Egress Filtering. As shown in Figure 8, ISP D provides Internet access to a leaf network, which can be a university or enterprise network. Router 1 is the edge router for the leaf network, which is connected to router 2, the edge router for ISP D. Router 3 is another edge router for ISP D, which is used to interconnect with other networks.

The purpose of ingress/egress filtering is to only allow traffic to enter or leave the network if its source addresses are within the expected IP address range. Suppose an attacker X resides within the leaf network. An input filter is placed in the input port of router 2 that is connected to the leaf network. This input filter only admits packets having a source IP address with the 204.69.207.0/24 prefix. If attacker X sends traffic with spoofed IP addresses that do not have the 204.69.207.0/24 prefix, that traffic will be dropped by the input filter in router 2. This filtering function provided by router 2 is called *ingress filtering* as it deals with traffic coming into the network of ISP D. However, if router 1 provides the same function, that function is called *egress filtering* as it deals with traffic leaving the leaf network.

In another scenario, suppose an attacker Y resides outside of ISP D and the leaf network. The attacker sends packets having a source IP address with the 10.0.0.0/8 prefix to server Z, which is located in the leaf network. An input filter is placed in the input port of router 3, which connects to the rest of the Internet. This input filter drops nonroutable IP addresses (e.g., 10.0.0.0/8). This function provided by router 3 is called

ingress filtering. Similarly, it is called *egress filtering* if provided by router 2, and *ingress filtering* if provided by router 1.

A key requirement for ingress or egress filtering is knowledge of the expected IP addresses at a particular port. For some networks with complicated topologies, it is not easy to obtain this knowledge. One technique known as *reverse path filtering* [Baker 1995] can help to build this knowledge. The technique works as follows. Generally, a router always knows which networks are reachable via any of its interfaces. By looking up source addresses of the incoming traffic, it is possible to check whether the return path to that address would flow out the same interface as the packet arrived upon. If they do, these packets are allowed. Otherwise, they are dropped. Unfortunately, this technique cannot operate effectively in real networks where asymmetric Internet routes are not uncommon.

5.1.1.2. Discussion. Since leaf networks normally have reasonably simple topologies, it is relatively easy to have knowledge of the expected IP addresses at a particular port. Moreover, routers in leaf networks generally have more spare computing resources than those in ISPs. Consequently, it is sensible to deploy ingress/egress filtering at leaf networks. More importantly, both ingress and egress filtering can be applied not only to IP addresses, but also protocol type, port number, or any other criteria of importance.

5.1.1.3. Effectiveness Against DoS and DDoS Attacks. DoS attacks tend to take advantage of IP spoofing to hide the attack sources and amplify the attack power. Therefore, both ingress and egress filtering provide some opportunities to throttle the attack power of DoS attacks. However, it is difficult to deploy ingress/egress filtering universally. If the attacker carefully chooses a network without ingress/egress filtering to launch a spoofed DoS attack, the attack can go undetected. Moreover, if an attack spoofs IP addresses from within the subnet, the attack can go undetected as well.

DDoS attacks can also choose networks without ingress/egress filtering or use subnet spoofing to avoid filtering. More importantly, nowadays DDoS attacks do not need to use source address spoofing to be effective [Handley 2005]. By exploiting a large number of compromised hosts, attackers do not need to use spoofing to take advantage of protocol vulnerabilities or to hide their locations. For example, each legitimate HTTP Web page request from 10,000 compromised hosts can bypass any ingress/egress filtering, but in combination they can constitute a powerful attack. Hence, ingress and egress filtering are ineffective to stop DDoS attacks.

5.1.2. Router-Based Packet Filtering. Router-based Packet Filtering (RPF) by Park and Lee [2001b] extends ingress filtering to the core of the Internet. It is based on the principle that for each link in the core of the Internet, there is only a limited set of source addresses from which traffic on the link could have originated. If an unexpected source address appears in an IP packet on a link, then it is assumed that the source address has been spoofed, and hence the packet can be filtered. In order to explain the operation of RPF, we first need to introduce several key concepts from interdomain routing in the Internet.

5.1.2.1. Analysis of Router-Based Packet Filtering. The Internet is divided into a set of routing domains, known as Autonomous Systems (ASs), where each AS corresponds to one or more networks that are controlled by a single administration entity, for example, a university or a corporation. Traffic is routed between ASs by *border routers* that use the Border Gateway Protocol (BGP) [Rekhter and Li 1995]. Each AS has one or more border routers depending on its topology, and is identified by a unique 16-bit AS ID. When viewed at the level of ASs, the whole Internet is connected by border routers. For

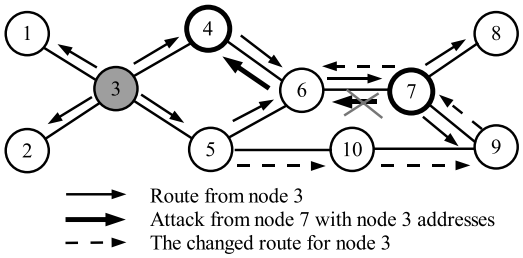

Fig. 9. Router-based packet filtering.

example, in Figure 9 each node represents a border router for one AS. From this point, we use the terms *AS* and *border router* interchangeably.

RPF uses information about the BGP routing topology to filter traffic with spoofed source addresses. Consider the example network in Figure 9, where an attack source in AS7 is flooding a target in AS4 with DoS attack traffic. The attack traffic (shown using bold arrows) has been spoofed so that its source address appears to come from AS3. Suppose RPF is deployed at AS6. The attack traffic from AS7 can be filtered if AS6 knows the BGP routing topology in the network. In particular, consider the routing topology for all paths from AS3 (shown as normal arrows), which is the spoofed source address of the attack traffic. Given this routing topology, there is no way that traffic from AS3 could arrive at the RPF at AS6 on the link from AS7 to AS6. Thus, all attack traffic that uses the spoofed source address of AS3 can be filtered at AS6, since it arrives on the link from AS7.

5.1.2.2. Discussion. Simulation results show that a significant fraction of spoofed IP addresses can be filtered if RPF is implemented in at least 18% of ASs in the Internet [Park and Lee 2001b]. However, there are several limitations of this scheme. The first limitation relates to the implementation of RPF in practice. Given that the Internet contains more than 10,000 ASs, RPF would need to be implemented in at least 1800 ASs in order to be effective, which is an onerous task to accomplish. Moreover, RPF requires modifications to the BGP message scheme [Rekhter and Li 1995], so that source addresses are included in BGP messages. This would significantly increase the size and processing time for BGP messages.

The second limitation is that RPF may drop legitimate packets if there has recently been a route change. For example, consider the case where the route from AS3 to AS6 has changed due to a link failure or a policy change. The new route traverses the AS path 3-5-10-9-7-6, as shown by the dashed arrows in Figure 9. If the RPF in the border route of AS6 has not been updated with this information, then legitimate packets from AS3 to AS4 will be dropped at AS6.

The third potential limitation is that RPF relies on valid BGP messages to configure the filter. If an attacker can hijack a BGP session and disseminate bogus BGP messages, then it is possible to mislead border routers to update filtering rules in favor of the attacker.

Finally, the filtering rules in RPF have a very coarse granularity, that is, at the AS level. The attacker can still spoof IP addresses based on the network topology. Alternatively, the attacker can launch the attack from compromised systems, without resorting to IP address spoofing.

5.1.2.3. Effectiveness Against DoS and DDoS Attacks. RPF is proposed for deployment in core networks. In general, a packet needs to pass multiple RPF filters before reaching the destination. Since it is difficult for an attacker to choose a path without a single

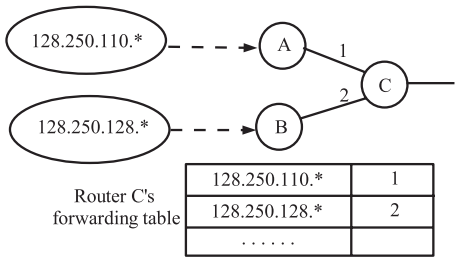

Fig. 10. An example of SAVE message updates.

RPF filter, RPF is effective against randomly spoofed DoS attacks. However, the filtering granularity of RPF is low. This means that the attack traffic can still bypass the RPF filters by carefully choosing the range of IP addresses to spoof.

In contrast, DDoS attacks can either use genuine IP addresses or spoof with carefully chosen source IP addresses. Hence, RPF is ineffective against DDoS attacks.

5.1.3. Source Address Validity Enforcement (SAVE) Protocol. As we discussed before, the router-based packet filter is vulnerable to asymmetrical and dynamic Internet routing as it does not provide a scheme to update the routing information. To overcome this disadvantage, Li et al. [2001] have proposed a new protocol called the *Source Address Validity Enforcement (SAVE) protocol*, which enables routers to update the information of expected source IP addresses on each link and block any IP packet with an unexpected source IP address.

5.1.3.1. Analysis of SAVE. The aim of the SAVE protocol is to provide routers with information about the range of source IP addresses that should be expected at each interface. Similarly to the existing routing protocols, SAVE constantly propagates messages containing valid source address information from the source location to all destinations. Hence, each router along the way is able to build an incoming table that associates each link of the router with a set of valid source address blocks. As shown in Figure 10, after receiving the SAVE messages, router C builds a forwarding table such that the IP address range 128.250.110.* is only expected on link 1 and IP address range 128.250.128.* is only expected on link 2.

5.1.3.2. Discussion. SAVE is a protocol that enables the router to filter packets with spoofed source addresses using incoming tables. It shares the same idea with ingress filtering and RPF that the source address space on each link of the router is stable and foreseen. Any packet that violates the expected source address space will be regarded as forged and will be filtered. SAVE outperforms ingress filtering and RPF in that it overcomes the asymmetries of Internet routing by updating the incoming tables on each router periodically. However, SAVE needs to change the routing protocol, which will take a long time to accomplish. Moreover, as SAVE filters the spoofed packets to protect other entities, it does not provide direct implementation incentives.

If SAVE is not universally deployed, attackers can always spoof the IP addresses within networks that do not implement SAVE. Moreover, even if SAVE were universally deployed, attackers could still launch DDoS attacks using nonspoofed source addresses.

5.1.3.3. Effectiveness Against DoS and DDoS Attacks. SAVE forces a DoS attack to spoof within a subnet, which greatly discourages a DoS attack. However, DDoS attacks do not rely on spoofing to be effective, which makes SAVE ineffective.

Table III. Comparison Between Attack Prevention Techniques

Prevention Techniques	Implementation Difficulty	Common Advantages	Common Limitations
Ingress filtering	Difficult for universal deployment.	Prevent IP source address spoofing.	Need wide adoption to be effective.
Router-based packet filtering	Possible if tier-1 ISPs are involved.	Filter attack traffic before it reaches the target, reduce collateral damage.	Not effective against IP spoofing within the same network or nonspoofed attacks.
SAVE protocol	Difficult due to the need for routing protocol change.		

5.1.4. Summary. To conclude, *attack prevention* aims to solve IP spoofing, a fundamental weakness of the Internet. The comparison between attack prevention techniques is shown in Table III. However, all the attack prevention schemes lack strong incentives for deployment. Unless new policies or legislation are introduced to enforce their deployment, it is doubtful that wide deployment of *attack prevention* schemes will happen in the near future.

More importantly, the attack prevention schemes assume attacks will be greatly reduced if every source address is accountable. However, as attackers gain control of larger numbers of compromised computers, attackers can direct these "zombies" to attack using valid source addresses. Since the communication between attackers and "zombies" is encrypted, only "zombies" can be exposed instead of attackers. According to the Internet Architecture Working Group [Handley 2005], the percentage of spoofed attacks is declining. Only four out of 1127 customer-impacting DDoS attacks on a large network used spoofed sources in 2004. Therefore, relying on attack prevention schemes is not enough to stop DDoS attacks.

5.2. Attack Detection

After attack prevention, the next step in defending against DoS attacks is *attack detection*. A critical measure of performance for any detection scheme is its coverage, that is, what proportion of actual attacks can be detected. Attack detection for DoS attacks is different from general intrusion detection. First, for general intrusions such as *user-to-root* and *remote-to-local* attacks, the attacker can hide the attack by changing the system log or deleting any file created by the attack. Thus these attacks are difficult to detect. However, DoS attacks can be easily detected since the target's services will be degraded, for example, with a high packet drop rate. Second, false positives are a serious concern for DoS attack detection. Since the potency of DoS attacks does not depend on the exploitation of software bugs or protocol vulnerabilities, it only depends on the volume of attack traffic. Consequently, DoS attack packets do not need to be malformed, such as an invalid fragmentation field or a malicious packet payload, to be effective. As a result, the DoS attack traffic will look very *similar* to legitimate traffic. This means that any detection scheme has a high risk of mistaking legitimate traffic as attack traffic, which is called a *false positive*.

If the DoS attack can be detected eventually, a common question is why do we need *attack detection*? There are several reasons for attack detection. First, if a target can detect an attack before the actual damage occurs, the target can win more time to implement attack reaction and protect legitimate users. Second, attack detection can help to identify the attackers so that legal actions can be taken. Third, if attacks can be detected close to attack sources, attack traffic can be filtered before it wastes any network bandwidth. However, there is generally insufficient attack traffic in the early stage of an attack and in the links close to the attack sources. Consequently, it is easy to mistake legitimate traffic as attack traffic. Therefore, it is challenging to accurately detect attacks quickly and close to the attack sources. Finally, "flash crowds" are very similar

to DoS attacks, which can cause network congestion and service degradation. However, "flash crowds" are caused by legitimate traffic, whereas DoS attacks are caused by malicious traffic. Hence, it is important to differentiate DoS attacks from "flash crowds" so that targets can react to them separately. Generally, there are two measures for DoS attack detection. The first is detection time and the second is false positive rate. A good detection technique should have a short detection time and low false positive rate.

Generally there are two groups of DoS attack detection techniques. The first group is called *DoS-attack-specific* detection, which is based on the special features of DoS attacks. The second group is called *anomaly-based* detection, which models the behavior of normal traffic, and then reports any anomalies.

5.2.1. DoS-Attack-Specific Detection. Generally, DoS attack traffic is created at an attacker's will. First, attackers want to send as much traffic as possible to make an attack powerful. Hence, attack traffic does not observe any traffic control protocols, such as TCP flow control. In addition, there will be a flow rate imbalance between the source and the victim if the victim is unable to reply to all packets. Second, attack traffic is created in a random pattern to make an attack anonymous. Third, for each known attack, attack traffic at the target is highly correlated with abnormal traffic behavior at the attack sources.

5.2.1.1. Analysis of DoS-Attack-Specific Detection. Gil and Poletto [2001] proposed a scheme called *MULTOPS* to detect denial of service attacks by monitoring the packet rate in both the up and down links. MULTOPS assumes that packet rates between two hosts are proportional during normal operation. A significant, disproportional difference between the packet rate going to and from a host or subnet is a strong indication of a DoS attack.

A drawback of MULTOPS is that it uses a dynamic tree structure for monitoring packet rates for each IP address. This tree structure can itself become the target of a memory exhausting attack. An alternative approach called *TOPS* [Abdelsayed et al. 2003] provides an efficient method for detecting packet flow unbalances based on a hashing scheme that uses a small set of field length lookup tables. This approach avoids the risk of memory exhausting attacks.

Wang et al. [2002] proposed *SYN detection* to detect SYN floods, and Blažek et al. [2001] proposed *batch detection* to detect DoS attacks. Both methods detect DoS attacks by monitoring statistical changes. The first step for these methods is to choose a parameter for incoming traffic and model it as a random sequence during normal operation. In Wang et al. [2002], the ratio of SYN packets to FIN and RST packets was used, while in Blažek et al. [2001] a variety of parameters, such as TCP and UDP traffic volume, were used. The attack detection is based on the following assumptions. First, the random sequence is statistically homogeneous. Second, there will be a statistical change when an attack happens.

Generally, DoS attack flows are not regulated by TCP flow control protocols as normal flows are. Hence, DoS attack flows have different statistical features compared with normal flows. Based on this assumption, Cheng et al. [2002] proposed to use spectral analysis to identify DoS attack flows. In this approach, the number of packet arrivals in a fixed interval is used as the signal. In the power spectral density of the signal, a normal TCP flow will exhibit strong periodicity around its round-trip time in both flow directions, whereas an attack flow usually does not.

Normally, an attacker performs a DoS attack using large numbers of similar packets (in terms of their destination address, protocol type, execution pattern etc.) generated from various locations but intended for the same destination. Thus, there is a lot of similarity in the traffic pattern. On the other hand, legitimate traffic flows tend to have

many different traffic types. Hence, traffic flows are not highly correlated and appear to be random. Based on this assumption, Kulkarni et al. [2002] proposed a Kolmogorov complexity based detection algorithm to identify attack traffic.

Based on the strong correlation between traffic behavior at the target and traffic behavior at the attack source, Cabrera et al. [2001] have proposed a scheme to proactively detect DDoS attacks using time series analysis. There are three steps to this scheme. The first step is to extract the key variables from the target. For example, the number of ICMP echo packets is the key variable for *Ping Flood* attacks. The second step is to use statistical tools (e.g., AutoRegressive Model) to find the variables from the *potential attackers* that are highly related to the key variable. For example, the number of ICMP echo reply packets at the potential attackers is highly correlated with the key variable for *Ping Flood* attacks. The third step is to build a normal profile using the found variables from the potential attackers. Any anomalies from potential attackers compared with the normal profile are regarded as strong indications of an attack. Steps 1 and 2 are completed during the offline training period and step 3 is done online.

5.2.1.2. Discussion. All DoS-attack-specific detection techniques are based on one or more assumptions. In the following text, we will challenge each assumption as well as provide countermeasures to evade detection.

MULTOPS assumes that the incoming packet rate is proportional to the outgoing packet rate, which is not always the case. For example, real audio/video streams are highly disproportional, and with the widespread use of online movies and online news, where the packet rate from the server is much higher than from the client, false positive rates, will become a serious concern for this scheme. Moreover, MULTOPS is vulnerable to attacks with randomly spoofed IP source addresses.

The simplest way to cripple MULTOPS is to use randomly spoofed IP addresses, which makes the calculation based on genuine IP addresses inaccurate and consumes resources by storing spoofed IP address information. Another countermeasure is to connect to the target from a large number of attack sources in a legitimate manner (e.g., downloading a file from a ftp server). Therefore, the packet rate ratio between *in flows* and *out flows*[4] during the attack will appear to be normal and will be *undetected* by MULTOPS.

The detection scheme in Wang et al. [2002] is based on the fact that a normal TCP connection starts with a SYN packet and ends with a FIN or RST packet. When the SYN flood starts, there will be more SYN packets than FIN and RST packets. The attacker can avoid detection by sending the FIN or RST packet in conjunction with the SYN packets. To beat the detection scheme in Blažek et al. [2001], the attacker can carefully mix different types of traffic to ensure the proportion of each traffic is the same as it is in normal traffic. Therefore, separating different types of traffic cannot make the attack behavior more conspicuous.

Spectral analysis techniques are only valid for TCP flows. As UDP and ICMP are connectionless protocols, the periodic traffic behavior is unexpected. Attackers can use UDP or ICMP traffic to confuse the detection scheme. Moreover, the attacker can mimic the periodicity of normal TCP flows by sending packets periodically. More importantly, attackers can make the reverse traffic from the target have the designed periodicity by using closed-looped protocols. For example, a large number of "zombies" can be directed to make legitimate TCP connections to the target.

The vulnerability of this scheme is that the efficacy of training is based on the features of known attacks. The attacker can disturb or disable the detection scheme by inventing

[4]We define *in flow* as the packet stream going to a host or subnet and *out flow* as the packet stream going from a host or subnet.

Table IV. Basic Assumptions for Different Attack Detection Techniques

Detection Technique	Basic Assumption	Assumption Strength	Technical Complexity
MULTOPS	Incoming traffic rate is proportional to outgoing traffic rate.	Medium	Low
SYN detection	$Number_{SYN\ packets} \cong Number_{FIN+RST\ packets}$.	Weak	Low
Batch detection	Attack traffic is statistically unstable.	Medium	Low
Spectral analysis	Attack flow does not have periodic behavior.	Strong	High
Kolmogorov test	Attack traffic is highly correlated.	Medium	High
Time series analysis	Attacks are limited to known attacks.	Medium	Medium

new attacks. As DDoS attacks do not necessarily need to use any particular type of traffic, it is easy for the attacker to create a new type of attack just by combining different types of attack traffic. This causes multiple key variables from the target, and the correlations between the variables from the potential attackers and the target will become extremely complex, which complicates the process of building a normal profile and makes the detection less effective. The assumption of the Kolmogorov test is based on the fact that multiple attack sources use the same DoS attack tool. Therefore, the resulting traffic is highly correlated. Unfortunately, there is no theoretical analysis to support this assumption. Attack sources can be orchestrated to break the correlation by sending attack traffic at different times, with different traffic types, packet sizes, and sending rates. This is easy to achieve. For example, attackers can use the IP address of a compromised computer as the random seed to generate a set of parameters for configuring attack traffic. By doing this, attack traffic will appear random, which can bypass detection.

To conclude, the efficacies of DoS-attack-specific detection can be evaluated in terms of their assumption strength and technical complexity. As shown in Table IV, most assumptions are not strong, since attackers can change their attack patterns to overthrow the assumption and evade detection. Although the assumption for *spectral analysis* is strong, it only works for TCP flows and it is complicated to implement.

A new DoS attack detection scheme using source IP address monitoring was presented in Peng et al. [2004]. Generally, the set of source IP addresses that is seen during normal operation tends to remain stable. In contrast, during DoS attacks, most of the source IP addresses have not been seen before. By using a carefully prebuilt IP Address Database, it is possible to sequentially monitor the proportion of new source IP addresses seen by the target, and detect any abrupt change using a statistical test called *Cumulative Sum (CUSUM)* [Brodsky and Darkhovsky 1993]. An abrupt change of the proportion of new source IP addresses is a strong indication of a DoS attack. More importantly, this method can improve the detection accuracy by also monitoring the traffic rate per IP address.

5.2.1.3. Effectiveness Against DoS and DDoS Attacks. Having controlled only a few computer systems, DoS attacks rely on several traffic patterns to maximize the attack power. For example, TCP-based DoS attack traffic is generally sent as fast as possible without observing TCP flow control principles. Therefore, DoS attacks differentiate themselves from legitimate traffic via these features, which can be used by DoS attack detection techniques to identify DoS attacks. For this reason, DoS attacks are generally easy to detect.

In contrast, DDoS attacks do not need to change the pattern of traffic from each compromised host to be effective, because there are usually many compromised hosts available. Each compromised host can mimic a legitimate user as closely as possible without degrading the total DDoS attack power. For example, each compromised host can randomly fetch a Web page from the target website, which can easily evade most

of the detection techniques mentioned above. The only promising detection techniques against DDoS attacks are those that capture the *inherent* features of an attack.

5.2.2. Anomaly-Based Detection. *Signature-based detection* and *anomaly-based detection* are two different approaches for network-based intrusion detection systems (IDS). Signature-based detection can identify an attack if the monitored traffic matches known characteristics of malicious activity. In practice, bandwidth attacks do not need to exploit software vulnerabilities in order to be effective. It is relatively easy for attackers to vary the type and content of attack traffic, which makes it difficult to design accurate signatures for DoS attacks [Kompella et al. 2004]. While signature-based detection can be used to detect communication between attackers and their "zombie" computers for known attack tools [Cheng 2006], in many cases this communication is encrypted, rendering signature-based detection ineffective. This limits the effectiveness of signature-based detection for DoS attacks. In contrast, anomaly-based detection can identify an attack if the monitored traffic behavior *does not* match the normal traffic profile that is built using training data. In 1987, Denning [1987] first proposed a real-time intrusion detection model to detect attacks by monitoring a system's audit records for abnormal patterns of system usage. Anomaly-based detection has since become a major focus of research, due to its ability to detect new attacks, including DoS attacks. For this reason, we focus our discussion on the use of anomaly-based detection for DoS attacks, rather than signature-based detection.

5.2.2.1. Analysis of Anomaly-Based DoS Detection. Building a normal profile is the first step for all anomaly-based detection techniques. Since there is no clear definition of what is normal, statistical modeling plays a crucial role in constructing the normal profile. Statistical anomaly detection includes two major parts. This first part is to find effective parameters to generate similarity measures. The parameters can be IP packet length, IP packet rate, and so on. Manikopoulos and Papavassiliou [2002] proposed solving this key issue by using *statistical preprocessing* and *neural network classification*. The second part is to calculate the similarity between the normal profile and new traffic. Statistical methods, such as χ^2 and Kolmogrov-Smirnov tests [Zhang et al. 2001; Manikopoulos and Papavassiliou 2002], have been used to provide similarity metrics to evaluate the difference between the monitoring traffic and the expected normal traffic. If the distance between the monitored traffic and the normal traffic profile is larger than a given threshold, a DoS attack is detected.

Inspired by human immunology, Forrest and Hofmeyr [1999] developed a network-based IDS, called *Lightweight Intrusion detection SYStem (LISYS)*, using the idea of an Artificial Immune System (AIS). LISYS was further extended by Bebo et al. [Williams et al. 2001]. The general idea for AIS-based network intrusion detection includes the following four steps. First, each IP packet is reduced to a string as its identity. For example, this string can contain the source IP address destination IP address, and destination port number. Second, during the training period, all packets that occur frequently are considered *self*, that is, normal. Third, based on *self*, detector strings are created such that they do not match any self string. Fourth, when the number of incoming packets that match the detector string reaches a certain threshold, an attack is reported.

5.2.2.2. Discussion. The common challenge for all anomaly-based intrusion detection systems is that it is difficult or impossible for the training data to provide all types of normal traffic behavior. As a result, legitimate traffic can be classified as attack traffic, causing a false positive. To minimize the false positive rate, a larger number of parameters are used to provide more accurate normal profiles. For example, in an

AIS-based IDS, longer strings can be used to improve the detection resolution. However, with the increase of the number of parameters, the computational overhead to detect an intrusion increases. This becomes a bottleneck, especially for volume-oriented DoS attacks that will be aggravated by the computational overhead of the detection scheme.

More importantly, unlike sophisticated network intrusions that depend on malformed packets or special packet sequences, DoS attacks only need the massive traffic volume to be effective. Thus different packet contents or traffic patterns will not affect the attack power. Unlike other attacks which are constrained to sending traffic that exploits a specific vulnerability, DoS attackers can mimic legitimate traffic to avoid anomaly-based detection. For example, an attacker can first use real data traces (either by using publicly available packet traces or monitoring real network traffic) to create a normal traffic profile, and then create the attack traffic according to this profile. Moreover, a system that uses sophisticated detection algorithms will become a victim itself during a large-scale DoS attack.

5.2.2.3. Effectiveness Against DoS and DDoS Attacks. DoS attack traffic generally deviates substantially from the profile of normal traffic, for example, DoS attacks usually involve a large number of packets from a small number of sources. More importantly, these abnormal features are inherent in DoS attacks if they are to be effective. Therefore, anomaly-based detection can be effective in detecting DoS attacks.

In contrast, DDoS attacks are launched from a large army of compromised hosts. Each host can behave like a "legitimate" source, but the overall effect is a powerful DDoS attack. This invalidates many anomaly-based attack detection techniques. The only hope for detecting DDoS attacks effectively and early is to use features that are difficult or impossible for an attacker to change, for example, the percentage of new IP addresses seen by the target [Peng et al. 2004].

5.2.3. Summary. DoS-attack-specific detection techniques generally use one or more features of DoS attacks, and can identify attack traffic effectively. However, all these techniques are based on one or more assumptions, which are not always reliable. Attackers can evade detection by overthrowing these assumptions. Anomaly-based detection techniques are facing a dilemma of how to choose a tradeoff between processing speed and detection accuracy. Moreover, attackers can use "legitimate traffic" generators to avoid detection.

5.3. Attack Source Identification

Once an attack has been detected, an ideal response would be to block the attack traffic at its source. Unfortunately, there is no easy way to track IP traffic to its source. This is due to two aspects of the IP protocol. The first is the ease with which IP source addresses can be forged. The second is the stateless nature of IP routing, where routers normally know only the next hop for forwarding a packet, rather than the complete end-to-end route taken by each packet. This design decision has given the Internet enormous efficiency and scalability, albeit at the cost of traceability and network security in terms of DoS attacks. In order to address this limitation, many schemes based on enhanced router functions or modification of the current protocols have been proposed to support IP traceability.

5.3.1. IP Traceback by Active Interaction. The main feature for IP traceback schemes in this category is that routers actively interfere with the attack traffic and trace the attack sources based on the reaction of attack traffic.

5.3.1.1. Analysis of Active IP Traceback Schemes. Backscatter traceback [Gemberling et al. 2001; Morrow and Gemberling 2001] is a traceback scheme based on the observation that DoS attacks generally use invalid spoofed source IP addresses. Typically, DoS attack traffic can use randomly spoofed source IP addresses. However, some IP addresses (e.g., IP address 10.*.*.*) have been reserved for private use instead of global routing. They can be used in private networks but are invalid in the Internet. The key procedures for *backscatter traceback* can be summarized as follows.

(1) A sinkhole router propagates a BGP route update to all the other routers for a target machine or network, where the next hop of the route update is a special TEST-NET host address, say 192.0.2.1. This causes all incoming traffic to the target to be captured at the network edge.
(2) A sinkhole router advertises itself as the next hop for a block of unallocated IP address space. Generally, 96.0.0.0/3 is recommended as it is the largest unallocated IP address space.
(3) When all packets headed for the target, including both the legitimate packets and the spoofed attack packets, are dropped at the ISP's network edge, ICMP Unreachable messages are generated by these edge routers to the source addresses. This is referred to as *backscatter*. It is worth noting that the "source IP addresses" can be spoofed IP addresses during a DoS attack, which could be *invalid or unallocated* IP addresses.
(4) For a randomly spoofed DoS attack, it is very likely that a spoofed source IP address falls into the range of 96.0.0.0/3. In that case, the ICMP Unreachable messages will be sent to 96.0.0.0/3, which will then be redirected to a sinkhole router.
(5) The sinkhole router is configured to log incoming ICMP Unreachable messages. These messages include the source address of the edge router that generated the ICMP Unreachable messages, which reveals the ingress point of the flood traffic.

Burch and Cheswick [2000] proposed a *link-testing traceback* technique. It infers the attack path by flooding all links with large bursts of traffic and observing how this perturbs the attack traffic. This scheme requires considerable knowledge of network topology and the ability to generate huge traffic in any network link. Generally, high-speed routers lack tracking ability, such as the ability to tell from which link a packet comes. Stone [1999] proposed an overlay network[5] architecture to overcome this limitation. During DoS attacks, attack traffic (traffic to the target) is rerouted to the overlay network which is called *CenterTrack*. The CenterTrack is normally equipped with routers configured for tracking. Thus, the attack packets can be easily tracked, hop-by-hop, through the overlay network, from the routers close to the target to the attack entry point of the ISP.

5.3.1.2. Discussion. Generally, active IP traceback schemes can locate attack paths reliably and quickly. However, the common shortcoming for all active IP traceback schemes is that substantial control is needed to coordinate all participating routers, which is unlikely for the Internet. Consequently, active IP traceback schemes are only suitable for identifying attack paths within one ISP's network, where the ownership of routers is unanimous.

To evade *backscatter traceback*, an attacker only needs to use a valid (spoofed or nonspoofed) IP address, as the scheme is based on the assumption that DoS attack traffic will always contain invalid source IP addresses, for example, 192.168.*.*. As

[5]An *overlay network* is a new physical or logical connection of a set of nodes on top of the existing network. In Stone's [1999] proposal, it refers to a logical connection.

link-testing traceback needs to flood the link to affect the attack traffic, it is questionable whether a target has the right or power to flood links for tracking purposes. Besides, when the attack traffic has multiple attack paths, there is only a small fraction of attack traffic on one attack path. Consequently, the change of the total attack traffic will be negligible by flooding a single link, which renders the link-testing scheme less effective. The CenterTrack scheme creates a logical overlay network by IP tunneling. The overhead to create the IP tunnel could amplify the negative effect of the DoS attack. In addition, DoS attacks that originate from within the overlay network cannot be tracked. Finally, it is not clear whether this scheme is scalable during a DDoS attack which has multiple entry points to the ISP.

5.3.1.3. Effectiveness Against DoS and DDoS Attacks. DoS attacks generally use only a few attack paths, and the source addresses in the attack packets are generally randomly spoofed. The first feature can be exploited by the *link-testing traceback* technique, and the second feature can be exploited by the *backscatter traceback* technique to identify the attack path.

In contrast, DDoS attack traffic comes from many geographically distributed links, which makes it difficult to infer the attack path. More importantly, most DDoS attacks do not need to use address spoofing, which makes these traceback techniques meaningless.

5.3.2. Probabilistic IP Traceback Schemes. The general idea of all probabilistic IP traceback schemes is that routers probabilistically insert partial path information into the incoming traffic, and the target reconstructs the packet path using the partial path information.

5.3.2.1. Analysis of Probabilistic Traceback Schemes. Savage et al. [2000] proposed to traceback the IP source by *probabilistic packet marking (PPM)*. The main idea of PPM is that each router embeds its IP address (partial path information) into the incoming packets probabilistically while they travel between the source and the destination. Based on the embedded path information, a target can reconstruct the packet transmission path. However, no specific field has been reserved for tracking purposes in the current Internet protocol IP v.4 (although IP v.6 [Deering and Hinden 1998] is expected to have such a field). Consequently, encoding schemes are needed to squeeze the path information into rarely used fields, such as the 16-bit identification field in the IP header. Song and Perrig [2001] have improved the efficiency and security of the PPM scheme by introducing a new hashing scheme to encode the path information, and an authentication scheme to ensure the integrity of the marking information. More details about PPM can be found in Savage et al. [2000]. In Dean et al. [2002], another coding scheme using an algebraic approach to embed path information was proposed to reduce the number of packets needed to reconstruct the attack path.

Bellovin [2000] proposed a similar approach called the *ICMP "traceback" scheme*. In this scheme, when a router receives a packet to a destination d, the router generates an ICMP traceback message, called an *iTrace packet*, with low probability. The iTrace packet contains the address of the router, and is sent to the destination d. For a significant traffic flow, the destination can gradually reconstruct the route that was taken by the packets in the flow. The iTrace packets are generated with a very low probability by routers to reduce the additional traffic, which undermines the effectiveness of the scheme. To prevent attackers from spoofing the ICMP packets, an authentication field is used in the iTrace packet. This scheme was later improved by Wu et al. [2001].

5.3.2.2. Discussion. Unlike active IP traceback, probabilistic approaches trace the source of IP packets passively without interfering with incoming traffic. Therefore,

less control of routers and less computational resources are needed for probabilistic approaches. However, under probabilistic packet marking schemes, the marking field can be overwritten, and all the routers use the same marking probability, with the result that the further the router the less possible it is to receive a marked packet from that router. To overcome this problem, a scheme called *Adjusted Probabilistic Packet Marking* was proposed in Peng et al. [2002a]. Under this scheme, each router adjusts the marking probability according to its distance to the target so that the target can receive the marked packets from all marking routers with the same probability.

One crucial assumption for all probabilistic approaches is that a significant amount of attack traffic transmits across the attack path. However, during a highly distributed denial of service attack (e.g., reflector attacks [Paxson 2001]), the attack traffic comes from a large number of links. Hence the number of attack packets is low on each *independent link*, where attack packets come from only one attack source. Therefore, these probabilistic approaches will fail to traceback the attack sources due to insufficient attack traffic on *independent links*.

Although authentication schemes were proposed to protect the marking field or the iTrace packet, many implementation issues need to be further studied. For example, many authentication schemes use public key infrastructure to sign the marked packet or iTrace packet. However, it is not clear who has the right to sign a packet and how one can validate that signature. Moreover, how to find a tradeoff between the level of security versus the computational overhead is still an open research problem. Without secured marking information or iTrace packets, it was noted in Park and Lee [2001a] that the attacker can generate IP packets with spoofed marking fields to mislead the path reconstruction, which makes probabilistic approaches less effective. More recently, Waldvogel [2002] has proved that attackers can insert fake paths efficiently using Groups of Strongly SImilar Birthdays (GOSSIB) attacks against PPM schemes.

5.3.2.3. Effectiveness Against DoS and DDoS Attacks. DoS attacks satisfy the assumption of *probabilistic IP traceback* techniques, that is, the attack traffic in one link is always substantially larger than normal traffic. Therefore, these traceback techniques are effective against *DoS* attacks.

In contrast, DDoS attack traffic comes from many geographically distributed links. More importantly, most DDoS attacks do not spoof source addresses, which obviates the need for these traceback techniques.

5.3.3. Hash-Based IP Traceback. As discussed before, all the probabilistic approaches fail to identify attack paths when attack traffic is very scarce on each *independent link* during a highly distributed denial of service attack. Similarly, probabilistic approaches also fail to traceback the attack source, where the attack only contains a small number of packets. For example, the "ping-of-death" attack only needs one sufficiently long ICMP packet that is fragmented into multiple datagrams in order to attack a vulnerable target [CERT 1996]. Consequently, a better traceback approach is needed, such that it is not affected by traffic volume and is able to traceback even one single packet.

5.3.3.1. Analysis of Hash-Based IP Traceback. Snoeren et al. [2001] proposed a scheme, called *hash-based IP traceback*, to trace individual packets. In this proposal, routers keep a record of every packet passing through the router. A Bloom filter [Bloom 1970] is used to reduce the memory requirement to store packet records. Moreover, in order to protect privacy, only packet digests, instead of actual packets, are stored. When a traceback is needed, a target will send a traceback query for one packet to its upstream traceback routers. Then a router can identify this packet by checking its records, and pass the query to its neighboring routers. Eventually the packet origin can be located.

Table V. Comparison Between Attack Source Identification Techniques

Identification Techniques	Implementation Difficulty	Defense Strength and Limitations	Common Limitations
Active interaction	Technically trivial	Needs human intervention. Effective when attacks are active.	Cannot guarantee the traceback granularity of a single host. Not effective at deterring attacks launched from compromised hosts.
Probabilistic packet marking schemes	Need wide deployment to be effective	Needs attack traffic to be considerably higher than normal traffic. Vulnerable to marking spoofing.	
Hash-based traceback scheme	Potentially large deployment cost	Can traceback a single packet but that packet needs to be recent.	

5.3.3.2. Discussion. This scheme is arguably the most effective scheme to traceback DDoS attacks. However, the success of traceback depends on the number of tracking routers installed, and the area covered by these routers. Although the Bloom filter is used to compress the storage, it is still a huge overhead for a router to implement this scheme, especially for high speed traffic over a long period. Therefore, wide deployment is not expected in the near future, and the traceback strength is limited. More importantly, if a router with tracking facilities is compromised by an attacker, spoofed information can be generated to mislead the traceback.

5.3.3.3. Effectiveness Against DoS and DDoS Attacks. This traceback technique is effective for both DoS attacks and DDoS attacks. However, in the case of a DDoS attack, even if the attack sources are revealed, it is difficult to take action due to the large number of attack sources.

5.3.4. Summary. A comparison between attack source identification techniques is shown in Table V.

5.4. Attack Reaction

Unlike more subtle attacks, such as *remote-to-local* attacks, DoS attacks try to damage the target as much as possible and attackers do not attempt to disguise the attack since the target will be aware of the attack damage eventually. All the detection and traceback techniques discussed above aim to shorten the time needed to detect the attack, and locate the attack sources. In order to minimize the loss caused by DoS attacks, a reaction scheme must be employed when an attack is underway.

Consider a DoS attack whose aim is to congest the target's communication channel, which includes the target and the network links to which the target is connected. Figure 11 shows a simple model of a DoS attack, where thick lines represent high-bandwidth links and thin lines represent low-bandwidth links. The bottleneck of a target's communication channel can be caused by low-bandwidth network links as well as poorly provisioned hosts. DoS attacks take effect once the resource limit of a bottleneck is reached. Hence, to minimize attack damage, the initial attack reaction is to protect the bottleneck's resources, which is called *bottleneck resource management*. Once the bottleneck resource is protected, the target is able to restore partial service instead of being completely paralyzed by the attack. An early proposal for a resource allocation model against DoS attacks was proposed by Millen [1992].

However, since the Internet is a resource-sharing architecture, resources will be wasted unless attack traffic is filtered at the source. The result of wasted resources will degrade the service quality of any host, including the target, which shares the path with attack traffic. Moreover, if the attack volume is large enough, new bottlenecks will appear, even though the original bottleneck has been protected. As shown in

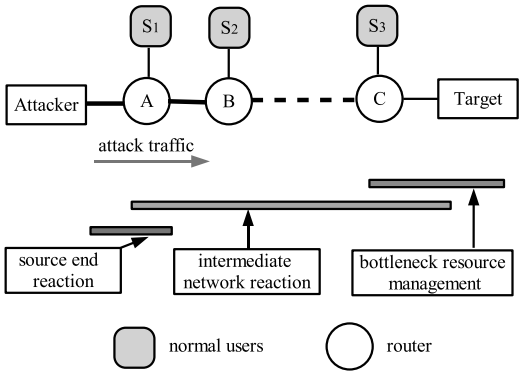

Fig. 11. A model of DoS attack reaction schemes.

Figure 11, the link between router C and the target is the bottleneck. Attack damage can be alleviated if *bottleneck resource management* schemes are used to protect this link. However, when the attack traffic volume is excessively high, the bandwidth limit of link A-B will be reached, and normal users S_1 and S_2 will fail to access the target. To protect S_1 and S_2, attack reaction should be applied at router A. We define *intermediate network reaction* as the attack reaction taken at the routers between the attacker and the victim. In an ideal situation, attack traffic should be filtered at the source (router A), which is called *source end reaction*. These three types of attack reaction are illustrated in Figure 11.

5.4.1. Bottleneck Resource Management. There are two main approaches to using resource management in order to react against bandwidth attacks. One is the *host resource management scheme*, which takes effect in the end host; the other is the *network resource management scheme*, which takes effect in the network link.

5.4.1.1. Analysis of Bottleneck Resource Management. One approach to managing host resources is to modify operating systems to fix software-based vulnerabilities. For example, systems using SYN cookies [Bernstein 1996] do not need to keep half-open states, and are less vulnerable to SYN flood attacks. Moreover, Schuba et al. [1997] proposed a method called *SYNkill*, which actively injects RST packets to the target to reset any suspicious TCP connection. Although this approach can protect the target from SYN floods by releasing the memory allocated to potentially illegitimate SYN requests, it also has several problems. First, it needs an accurate algorithm to differentiate legitimate SYN requests from SYN floods. Otherwise, the legitimate users may be punished as well. Second, as *SYNkill* needs to inject packets as well as monitor the network traffic, it is likely to become a new bottleneck. More importantly, when the SYN flood involves a high traffic rate, the injected RST packets will congest the network links and exacerbate the situation. Another host resource management scheme is to punish attack traffic and reserve resources for well-behaved users or processes using end-to-end resource accounting [Spatscheck and Petersen 1999] and traffic shaping [Kargl et al. 2001]. In Kargl et al. [2001], also proposed to use a server farm together with a *load balancer* to enhance a Web server's capacity. With this increased capacity, the Web server is able to handle more Web requests and is less likely to be disabled by a bandwidth attack. Another approach called *History-based IP Filtering* [Peng et al. 2003] proposed to filter bandwidth attack traffic according to the history maintained by the target. In particular, the target can use an IP Address Database to keep all the IP addresses that

frequently appeared at the target. During a bandwidth attack, the target only admits the packets whose source IP addresses belong to the IP Address Database.

While the host resources are effectively managed, network resources are likely to become the bottleneck during DoS attacks. How to manage and protect network resources becomes a key step for DoS attack defense. In Lau et al. [2000] have shown that *class-based queuing (CBQ)* [Floyd and Jacobson 1995] algorithms can guarantee bandwidth for certain classes of input flows, while *Random Early Detection (RED)* [Floyd and Jacobson 1993] performs poorly with regard to DDoS attacks. This lies in the fact that CBQ classifies traffic and reserves resources for each class of traffic. Yau et al. [2002] have proposed a feedback control scheme on the router to throttle the *aggressive (attack) traffic flow* with max-min fairness. This scheme can proactively rate-limit the attack traffic before it reaches the server, and therefore forestalls the DDoS attack.

5.4.1.2. Discussion. As bottleneck resource management mechanisms aim to deploy at the target or routers close to the target, they are easy to implement. Most commercial DoS attack solutions belong to this type. Both host and network resource management schemes need to classify traffic into several types, and then treat them differently. Unfortunately, it is rather difficult to give an accurate classification as DoS attack traffic can mimic any type of legitimate traffic. Without a proper rule to characterize attack traffic, the target will fail to provide services to legitimate users. Even though a sophisticated algorithm can do a better job on classifying traffic, a large-scale DoS attack can succeed by exploiting the resource-intensive nature of such an algorithm. Consequently, any type of large-scale DoS attacks that simulate normal traffic behavior will defeat bottleneck resource management schemes.

Alternatively, some service providers try to eliminate the bottleneck by simply increasing both host and network resources. For example, high-profile Web sites, such as Yahoo and Microsoft, generally weather DoS attacks by investing an enormous amount of money on expanding the server capacity and the Internet connection bandwidth. This solution is arguably very effective. However, it entails a huge financial expense which only a few Web sites can afford. More importantly, this solution only increases the difficulty for a successful attack, and does not eliminate the DoS attack threat fundamentally. An excessively large DoS attack, such as the "Code-Red worm" [CERT 2001], is still able to succeed.

5.4.1.3. Effectiveness Against DoS and DDoS Attacks. DoS attacks have several features that are different from normal traffic, such as their lack of response to TCP congestion control. These features can then be used to prioritize legitimate traffic and filter DoS attack traffic. Moreover, if a DoS attack mimics legitimate traffic to avoid filtering, then only a small proportion of the target's resources will be occupied by the attack. This is because a few DoS attack sources are forced to share resources *fairly* with a large number of legitimate users.

In contrast, the number of DDoS attack sources may outnumber the legitimate users of the target. Therefore, bottleneck resource management schemes are not effective, as most resources will be "fairly" shared by the DDoS attack traffic and few resources are left for the legitimate users.

5.4.2. Intermediate Network Reaction. As we analyzed above, protecting bottleneck resources only relieves attack damage instead of eliminating attacks completely. It is essential to filter attack traffic *close to attack sources*. The first benefit is to save bandwidth that will otherwise be wasted by attack traffic. The second benefit is to separate attack traffic from legitimate traffic *geographically*. Given that no accurate attack signature is available at a single location, the closer the defense location is to the attack

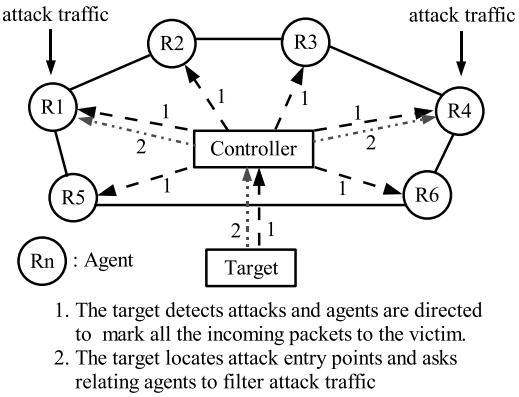

Fig. 12. Intermediate network reaction: controller-agent scheme.

sources, the more that legitimate traffic will be protected. We define *intermediate network reaction* as the defense mechanism that filters attack traffic using routers in between attack sources and a target. Unfortunately, it gets more and more difficult to detect DoS attacks as the distance increases between the detection point and the target, due to reduced attack evidence. Therefore, a communication mechanism is needed to keep the routers between the target and attack sources informed of an attack. Then these routers start to filter attack traffic according to the information provided by the victim or developed by their local defense agents. In the following section, we will introduce three types of intermediate network reaction schemes, where pushback and controller-agent schemes are based on active cooperation between routers and a victim, and secure overlay services are based on anonymous routing and multiple-level filtering.

5.4.2.1. Analysis of Intermediate Network Reaction. Mahajan et al. [2002] provided a scheme in which routers learn a congestion signature that can differentiate legitimate traffic from malicious traffic based on the volume of traffic to the target from different links. In Mahajan's proposal, the congestion signature is the target's IP address [Mahajan et al. 2002]. The router then filters the bad traffic according to this signature. Furthermore, a pushback scheme is given to let the router ask its adjacent routers to filter the bad traffic at an earlier stage. By pushing the defense frontier toward the attack sources, more legitimate traffic will be protected. An improved version of this pushback scheme called *Selective Pushback* [Peng et al. 2002b] sends pushback messages to the routers closest to the attack sources *directly* by analyzing the traffic distribution change of all upstream routers at the target. The benefit of this scheme is twofold. First, traffic distribution analysis can locate attack sources more accurately than purely volume-based approaches, especially during a highly distributed denial of service attack. Second, the pushback message can be sent to the routers closest to the attack sources directly, which can mitigate the attack damage more quickly than the original pushback scheme.

Tupakula and Varadharajan [2003] proposed an agent-controller model to counteract DoS attacks within one ISP domain, which is illustrated in Figure 12. In this model, *agents* represent the edge routers and *controllers* represent trusted entities owned by the ISP. Once a target detects an attack, it sends a request to the controller, asking all agents to mark all packets to the target. After checking the marking field, the target can find out which agent (edge router) is the entry point for the attack traffic. The target

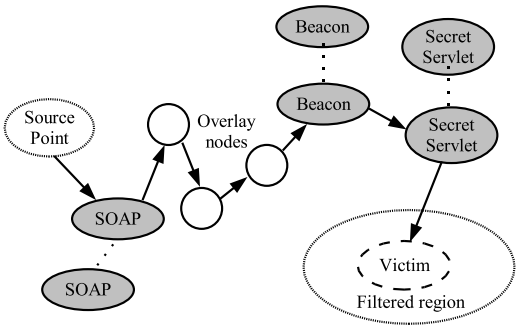

Fig. 13. Basic SOS architecture.

then sends a refined request to the controller, asking some particular agents to filter attack traffic according to the attack signature provided by the target.

Keromytis et al. [2002] proposed an architecture called *secure overlay service (SOS)* to secure the communication between the *confirmed users* and the *victim*. As shown in Figure 13 [Keromytis et al. 2002], all the traffic from a *source point* is verified by a *secure overlay access point (SOAP)*. Authenticated traffic will be routed to a special overlay node called a *beacon* in an anonymous manner by consistent hash mapping. The beacon then forwards traffic to another special overlay node called a *secret servlet* for further authentication, and the secret servlet forwards verified traffic to the victim. The identity of the secret servlet is revealed to the beacon via a secure protocol, and remains a secret to the attacker. Finally, only traffic forwarded by the secret servlet chosen by the victim can pass its perimeter routers. There are two design rationales of SOS. First, SOAPs are essentially acting as a distributed firewall. With a large number of SOAPs working in distributed manner, each SOAP only needs to deal with a small proportion of the attack traffic. Therefore, sophisticated protocols, such as IPsec [Kent and Atkinson 1998], can be used to verify the legitimacy of the traffic. Second, the final node that connects to the victim is unknown to attackers. Therefore, attackers cannot find any vulnerable link to the victim.

5.4.2.2. Discussion. The basic assumption for all schemes is that there is a limited number of attack paths, and not all legitimate traffic shares a path with the attack traffic. Without confidence in accurately differentiating attack traffic from legitimate traffic at a single location, all schemes try to identify attack paths based on network topology. By filtering traffic along the attack paths, at least legitimate traffic that does not share the path with attack traffic will be protected. Unfortunately, the assumption fails when the attack traffic is uniformly distributed. For example, *reflector attack* traffic can easily be geographically distributed by choosing reflectors from different locations. Consequently, all intermediate network reaction schemes are vulnerable to a large scale reflector attack.

This scheme is effective against most DDoS attacks except uniformly distributed attack sources. However, it needs a narrow and accurate congestion signature to make sure only attack traffic is filtered while legitimate traffic is not affected. Since the pushback scheme aggregates attack traffic according to destination IP addresses, it is vulnerable to attack traffic with spoofed source addresses. Moreover, this scheme infers attack sources by checking the traffic volume to the victim on each upstream link. If the attack sources are highly distributed, the traffic volume to the victim on each upstream link will appear to be similar, which invalidates the pushback scheme.

The aim of the controller-agent model is to filter attack traffic at the edge routers of one ISP domain. Since there are two communication processes[6] among the target, controllers, and agents, it is doubtful whether the control messages can get through during network congestion, and whether the attack reaction is quick enough to curtail the attack. Moreover, since this model is limited to a single ISP domain, an attacker can paralyze a target by flooding the whole ISP's network given enough attack power. More importantly, if attack sources are geographically distributed, attack traffic can appear from most, if not all, entry points of an ISP. Therefore, the attack traffic will share most entry points with legitimate traffic. Then the effectiveness of the model depends on the capability to separate attack traffic from legitimate traffic at the entry points, which is a challenging task.

SOS addresses the problem of how to guarantee the communication between legitimate users and a victim during DoS attacks. Keromytis et al. [2002] demonstrated that SOS can greatly reduce the likelihood of a successful attack. The power of SOS is based on the number and distribution level of SOAPs. However, wide deployment of SOAPs is a difficult DoS defense challenge. Moreover, the power of SOS is also based on the anonymous routing protocol within the overlay nodes. Unfortunately, the introduction of a new routing protocol is in itself another security issue.

If an attacker is able to breach the security protection of some overlay node, then it can launch the attack from inside the overlay network. Moreover, if attackers can gain massive attack power, for example, via worm spread, all the SOAPs can be paralyzed, and the target's services will be disrupted.

5.4.2.3. Effectiveness Against DoS and DDoS Attacks. DoS attack traffic is geographically centralized, although it may appear distributed to the target due to IP spoofing. If there is cooperation among the routers, the DoS attack traffic can be separated at a point *close to* the source with minimal collateral damage.

In contrast, DDoS attack traffic can be geographically distributed and there are many simultaneous attack paths. It is difficult to take advantage of the topology as DDoS attack traffic can share links with most of the legitimate traffic. Therefore, intermediate network reaction is only effective if the DDoS attack originated from one or a few *networks*, for example, from the same ISP's network.

5.4.3. Source End Reaction. As the ultimate goal for DoS attack defense is to filter attack traffic at the source, Mirković et al. [2002] proposed a scheme called *D-WARD* to defend against DoS attacks at the source network, where the attack sources are located.

5.4.3.1. Analysis of D-WARD. First, D-WARD collects flow statistics by constantly monitoring two-way traffic between the source network and the rest of the Internet. The flow statistics include the ratio of in-traffic and out-traffic, the number of connections per destination, and so on. Second, it periodically compares the measured statistics with normal flow models, where a separate normal flow model is built for each type of traffic. Third, once a flow mismatches the normal flow model, it will be classified as an attack flow, and will be filtered or rate-limited.

5.4.3.2. Discussion. D-WARD addresses the fundamental DoS attack defense rationale: removing attack traffic at its source. However, it faces the following two challenges. First, for a large-scale DDoS attack, attack traffic generated by one source network can be very small and unnoticed compared with legitimate traffic flows. Hence, detecting attack traffic accurately can be difficult or impossible. A well-organized, geographically distributed DoS attack is likely to defeat this scheme as attackers can control

[6]The first one is the marking process and the second is the filtering process.

Table VI. Comparison Between Attack Reaction Techniques

Reaction Techniques	Implementation Incentives	Defense Strength and Limitations	Technical Challenges
Bottleneck resource management	Users are highly motivated to deploy such schemes.	Can effectively relieve attack damage at the cost of high collateral damage.	How to differentiate attack traffic from legitimate traffic.
Intermediate network reaction	ISPs need to be financially motived, (e.g., value-added security services).	Filters attack traffic before it reaches the target. Limited collateral damage.	How to deal with distributed non-spoofed attacks.
Source end reaction	Very unlikely to be widely deployed unless enforced by legislation.	Stops attack traffic from polluting Internet, an ideal defense scenario.	How to detect an attack at the source before attack traffic aggregation.

the attack traffic from each source network to be within normal range. Second, while D-WARD plays a similar role as ingress filtering, it is more expensive to implement. Consequently, the motivation for deployment is a big concern.

5.4.3.3. Effectiveness Against DoS and DDoS Attacks. Due to the limited number of attack sources, the attack traffic pattern at the source is similar to the pattern at the target. Hence, the DoS attack traffic can be detected and filtered at the source.

In contrast, DDoS attack traffic at the source can look as "normal" as other legitimate users. It is the aggregation of all these "normal" traffic flows at the target that makes a DDoS attack. Hence, detection and filtering at the DDoS attack source can be difficult or impossible without using information sharing among the multiple sources.

5.4.4. Summary. The comparison between attack reaction techniques is shown in Table VI.

6. INTEGRATED SOLUTIONS TO DDOS ATTACKS

While there has been considerable research effort into defenses against DDoS attacks, there has been only limited progress in solving the DDoS problem. Most approaches focus on detecting and filtering attack traffic near the target of the attack. The main limitation of this general approach is that the computational and network resources available to the attacker can readily exceed that of the target. This is because attackers have been able to increase their attack power by gaining control of large numbers of zombie computers. Given the large number of traffic sources at their disposal, attackers no longer need to hide the identity of the "zombies" using spoofing. This means that the "zombies" can engage in more complex transactions such as authentication requests or Web queries, which are difficult to differentiate from legitimate traffic. In order to respond to this growth in attack power, defenders need a more scalable approach to defense. In this section, we highlight opportunities for a more integrated solution to defense against DDoS attacks, which could enable the target to marshal additional resources to assist in defending against large-scale attacks.

Before we examine the needs of an integrated approach to large-scale attacks, let us first examine how smaller scale attacks can be handled at the target. Consider how the difficulty of defending against an attack varies with the number of attack sources and whether those sources use IP address spoofing to hide their true source address. In the simplest case of a single attack source using its true identity, the attack source can easily be identified at the target based on the volume of traffic that it sends. High-volume

sources can be rate-limited, or discarded if they do not respond to flow control requests. In the case of a single source using multiple source addresses, the attack sources cannot be reliably determined based on the volume of traffic that they send, since the traffic volume is split between multiple spoofed source addresses from the target's point of view. Defense at the target relies on trying to filter attack traffic from normal traffic based on some anomalous feature of the attack traffic. The case of multiple attack sources, each using multiple source addresses, also relies on filtering at the target. However, as the attack power grows by using multiple sources, the computational requirements of filtering can become a burden at the target. In practice, many attacks now involve multiple sources using their true source identities. In this case, each attacker can establish valid TCP connections and generate legal requests of the target. This makes filtering at the target a more challenging problem, due to the difficulty in identifying legal, but malicious, requests.

A complementary approach to blocking attack traffic is to limit the rate at which sources can generate requests. If a target service is designed for use by a person, then it may be reasonable to filter all traffic that is generated by an automated source, for example, an attack "zombie." When an unfamiliar source uses a service for the first time, then it must first complete an admission challenge that requires human judgment, such as reading a character string that has been presented as an image [Morein et al. 2003]. This denies access to automated sources, which would be unable to complete the challenge. Such challenges can be reissued to a source if that source starts to generate a large number of requests, that is, the person has been replaced by an automated source. A variant on this approach has been proposed for target services that are intended for use by automated sources, for example, DNS servers. In this case, the admission challenge takes the form of a computational puzzle, which is designed to be easy to set and verify, but hard to solve, for example, a constraint satisfaction problem [Kandula et al. 2005]. In this case, any additional requests from a source are blocked until the initial challenge has been solved. However, this form of puzzle-based challenge requires compatible client software at the source, which may limit the deployment of this approach. Similarly, admission challenges that require human judgement can create more work for legitimate users, and may not achieve user acceptance. Furthermore, both types of challenge still require some computational resources at the target, which can become a bottleneck during an attack.

All of the above defense techniques place the burden of defense on the target of the attack. In contrast, the attacker has the potential to increase their attack power by infecting more "zombie" computers. A possible approach to redress this imbalance is to provide an integrated defense solution that enables filtering and admission challenges to be implemented in a distributed manner throughout the network on behalf of the target, for example, DefCOM [Mirkovic et al. 2003] and COSSACK [Papadopoulos et al. 2003]. The defense measures can then propagate back into the network from the target towards the sources when attacks occur. Under normal conditions, no filtering or admission challenges are required. When an attack begins, these defense measures are first implemented centrally at the target. If the attack persists or worsens, then the target could propagate a distress signal upstream to its Internet Service Provider (ISP), who could deploy proxy defenses at the ingress points to the ISP's network on behalf of the target. In general, the target's ISP could request other upstream ISPs to also deploy the defenses for the target, so that the attack traffic is blocked as close as possible to the source of the traffic.

This form of integrated solution combines filtering and admission challenges with a pushback scheme between the target and the upstream ISPs. While pushback schemes have been proposed for DDoS defense [Mahajan et al. 2002; Peng et al. 2002b], we can identify several open issues that need to be addressed in order to provide an integrated

and effective solution. The first issue is how to implement a pushback signaling scheme that provides sufficient information for effective filtering or admission challenges. The pushback signal may need to encode information about the targets, possible sources, and distinguishing features of normal traffic or attack traffic. A key challenge in providing this pushback signal is how to ensure accuracy without overwhelming the upstream proxy defenses. The second issue is how to ensure that the pushback signal can be trusted, so that it is not open to manipulation by attackers. The problem of managing trust in a distributed environment is a challenging issue for research. The third issue is how to manage any risks of liability if a proxy defense makes an incorrect decision. For example, an ISP is likely to be unwilling to implement such a scheme if they are at risk of being sued for blocking legitimate traffic or passing attack traffic. The final issue is how to ensure the scalability of the pushback approach when it involves multiple ISPs and targets with many simultaneous attacks.

In this section, we have motivated and outlined an integrated approach for defending against DDoS attacks. This potential solution combines filtering and admission challenges in a pushback scheme. A key advantage of this proposed approach is that it could enable the defenders to harness greater computational resources in order to counteract the growth in attack power that is becoming available to attackers. However, many open issues still need to be addressed, both in terms of research and management. So long as vulnerable computers are available to attackers for use as zombies, it seems likely that the balance of power will favor attackers until a scalable defense solution is put in place.

7. CONCLUSION

With the release of new operating systems, users are given more power over computer resources. For example, a normal user of *Windows XP Home Edition* is allowed to access *raw sockets*, a data structure that can be used for IP spoofing, which is only available for *root* users of Unix-like operating systems. Furthermore, both the number of Internet users and the users' bandwidth have kept increasing dramatically. Unfortunately, the average security knowledge for current Internet users is decreasing while attacks are becoming more and more sophisticated [Lipson 2002]. As a result, the attack power is expanding rapidly. On the other hand, although the security community works very hard to patch the vulnerabilities, defense effects are limited due to the lack of central control of the Internet.

In this article, we have presented a comprehensive survey of the causes of DoS attacks, and the techniques that have been proposed to detect and respond to these attacks. One important step to combat DoS attacks is to increase the reliability of global network infrastructure. More reliable mechanisms are needed to authenticate the source of Internet traffic, so that malicious users can be identified and held accountable for their activities. Having more secure computer systems on the Internet will greatly reduce attackers' power to launch large scale DDoS attacks. Another important step to combat DoS attacks is global cooperation, for example, cooperative IP traceback. However, it is a long and difficult path to achieve these goals. The main reason is that there is a lack of economic incentives for personal users or ISPs to invest money on security to mainly protect others' networks. A usage-based billing system proposed in Geng and Whinston [2000] might provide a certain level of motivation for personal users to secure their own systems. More importantly, similar problems, such as *the tragedy of the commons*[7][Hardin 1968], have been solved through legislation.

[7]The tragedy of the commons [Hardin 1968] happens when individuals try to maximize their benefits while ignoring the public interests.

Optimistically, the DoS attack problem can draw the attention of lawmakers, and global cooperation can be enforced by legislative measures.

Generally, it is expensive if not impossible to eliminate the DoS attack problem entirely. As we discussed in the previous sections, the most effective DoS defense scheme is to detect and block attack traffic close to the source. However, the implementation cost for this scheme is high, due to the difficulty in discriminating between legitimate and malicious traffic at its source. In the short term, there is a growing range of defense techniques that can be deployed close to the target and provide a reasonable level of protection. In the medium term, we expect that Internet Service Providers will begin to deploy more distributed defense mechanisms at the ingress and egress points of their networks. The longer-term challenge for defense against DoS attacks is how to achieve cooperation between ISPs, in order to block malicious traffic close to its source, before it has the chance to congest the wider Internet.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their excellent suggestions that have greatly improved the quality of this article.

REFERENCES

ABDELSAYED, S., GLIMSHOLT, D., LECKIE, C., RYAN, S., AND SHAMI, S. 2003. An efficient filter for denial-of-service bandwidth attacks. In *Proceedings of the 46th IEEE Global Telecommunications Conference* (GLOBECOM'03). 1353–1357.

ARBOR. 2005. Worldwide ISP security report. Whitepaper. Arbor Networks, Lerington, MA .

BAKER, F. 1995. Requirements for IP version 4 routers. RFC 1812. Internet Engineering Task Force (IETF). Go online to `www.ietf.org`.

BELLOVIN, S. 2000. The ICMP traceback message. IETF Internet Draft. Internet Engineering Task Force (IETF). Go online to `www.ietf.org`

BERNSTEIN, D. J. 1996. SYN cookies. Go online to `http://cr.yp.to/syncookies.html`.

BLAŽEK, R. B., KIM, H., ROZOVSKII, B., AND TARTAKOVSKY, A. 2001. A novel approach to detection of "denial-of-service" attacks via adaptive sequential and batch-sequential change-point detection methods. In *Proceedings of the 2001 IEEE Systems, Man and Cybernetics Information Assurance Workshop*.

BLOOM, B. H. 1970. Space/time tradeoffs in hash coding with allowable errors. *Commun. ACM 13*, 7 (Jul.), 422–426.

BRODSKY, B. E. AND DARKHOVSKY, B. S. 1993. *Nonparametric Methods in Change-point Problems*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

BURCH, H. AND CHESWICK, B. 2000. Tracing anonymous packets to their approximate source. In *Proceedings of the 14th Systems Administration Conference* (New Orleans, LA).

CABRERA, J. B. D., LEWIS, L., QIN, X., LEE, W., PRASANTH, R. K., RAVICHANDRAN, B., AND MEHRA, R. K. 2001. Proactive detection of distributed denial of service attacks using MIB traffic variables—a feasibility study. In *Proceedings of the 7th IFIP/IEEE International Symposium on Integrated Network Management* (Seattle, WA). 609–622.

CAIDA. 2006. Nameserver DoS attack October 2002. Go online to `http://www.caida.org/funding/dns-analysis/oct02dos.xml`.

CERT. 1996. CERT Advisory CA-1996-26: denial-of-service attack via ping. Go online to `http://www.cert.org/advisories/CA-1996-26.html`.

CERT. 1998. CERT Advisory CA-1998-01: Smurf IP denial-of-service attacks. Go online to `http://www.cert.org/advisories/CA-1998-01.html`.

CERT. 2001. CERT Advisory CA-2001-19: "Code Red" Worm exploiting buffer overflow in IIS indexing service DLL. Go online to `http://www.cert.org/advisories/CA-2001-19.html`.

CERT. 2003. CERT Advisory CA-2003-19: Exploitation of vulnerabilities in Microsoft RPC Interface. Go online to `http://www.cert.org/advisories/CA-2003-19.html`.

CERT. 2006. CERT/CC statistics. Go online to `http://www.cert.org/stats/cert_stats.html`.

CHANG, R. K. C. 2002. Defending against flooding-based distributed denial-of-service attacks: A tutorial. *IEEE Commun. Mag. 40*, 10 (Oct.), 42–51.

CHEN, E. Y. 2006. Detecting dos attacks on SIP systems. In *Proceedings of the 1st IEEE Workshop on VoIP Management and Security*. 53–58.

CHENG, C.-M., KUNG, H. T., AND TAN, K.-S. 2002. Use of spectral analysis in defense against DoS attacks. In *Proceedings of IEEE GLOBECOM 2002*. 2143–2148.

CHENG, G. 2006. Malware FAQ: Analysis on DDOS tool Stacheldraht v1.666. Go online to `http://www.sans.org/resources/malwarefaq/stacheldraht.php`.

CHEUNG, S. 2006. Denial of service against the domain name system. *IEEE Sec. Pri. 4*, 1, 40.

CLARK, D. D. 1988. The design philosophy of the DARPA Internet protocols. In *Proceedings of SIGCOMM* (Stanford, CA). 106–114.

DAVIS, M. 2006. Building better bots: Open-source processes enable production-grade malware. *Sage: Security Vision from McAfee Avert Labs 1*, 1 (Jul.), 26–35.

DEAN, D., FRANKLIN, M., AND STUBBLEFIELD, A. 2002. An algebraic approach to IP traceback. *ACM Trans. Inform. Syst. Sec. 5*, 2 (May), 119–137.

DEERING, S. AND HINDEN, R. 1998. Internet protocol, version 6 (IPv6) specification. RFC 2401. Internet Engineering Task Force (IETF). Go online to `www.ietf.org`.

DENNING, D. E. 1987. An intrusion-detection model. *IEEE Trans. Softw. Eng. 13*, 2, 222–232.

DIETRICH, S., LONG, N., AND DITTRICH, D. 2000. Analyzing distributed denial of service attack tools: The shaft case. In *Proceedings of the 14th Systems Administration Conference* (New Orleans, LA). 329–339.

EVANS, D. AND LAROCHELLE, D. 2002. Improving security using extensible lightweight static analysis. *IEEE Softw. 19*, 1, 42–51.

FERGUSON, P. AND SENIE, D. 2000. Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. RFC 2827. Internet Engineering Task Force (IETF). Go online to `www.ietf.org`.

FLOYD, S. AND JACOBSON, V. 1993. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Netw. 1*, 4 (Aug.), 397–413.

FLOYD, S. AND JACOBSON, V. 1995. Link-sharing and resource management models for packet networks. *IEEE/ACM Trans. Netw. 3*, 4 (Aug.), 365–386.

FORREST, S. AND HOFMEYR, S. 1999. Architecture for an artificial immune system. *Evolution. Computat. J. 7*, 1, 45–68.

GARBER, L. 2000. Denial-of-service attacks rip the Internet. *IEEE Comput. 33*, 4 (Apr.), 12–17.

GEMBERLING, B., MORROW, C., AND GREENE, B. 2001. ISP security-real world techniques. Presentation, NANOG. Go online to `www.nanog.org`

GENG, X. AND WHINSTON, A. 2000. Defeating distributed denial of service attacks. *IEEE IT Profess. 2*, 4 (Jul./Aug.), 36–41.

GIBSON, S. 2002. Distributed reflection denial of service. Go online to `http://grc.com/dos/drdos.htm`.

GIL, T. M. AND POLETTO, M. 2001. MULTOPS: A data-structure for bandwidth attack detection. In *Proceedings of the 10th USENIX Security Symposium*.

GLIGOR, V. D. 1984. A note on denial-of-service in operating systems. *IEEE Trans. Softw. Eng. 10*, 3, 320–324.

GORDON, L. A., LOEB, M. P., LUCYSHYN, W., AND RICHARDSON, R. 2005. *2005 CSI/FBI Computer Crime and Security Survey*. Available online at `www.GCSI.com`.

HANDLEY, M. 2005. Internet Architecture WG: DoS-resistant Internet subgroup report. Available online at `http://www.communications.net/object/download/1543/doc/mjh-dos-summary.pdf`.

HARDIN, G. 1968. The tragedy of the commons. *Science*, 1243–1248.

HONEYNET. 2005. Know your enemy:tracking botnets. Whitepaper. The Honeynet Project & Research Alliance. Feb. Go online to `www.honeynet.org/index.html`.

HUSSAIN, A., HEIDEMANN, J., AND PAPADOPOULOS, C. 2003. A framework for classifying denial of service attacks. In *Proceedings of the ACM SIGCOMM Conference* (Karlsruhe, Germany). 99–110.

KANDULA, S., KATABI, D., JACOB, M., AND BERGER, A. W. 2005. Botz-4-Sale: Surviving organized DDoS attacks that mimic flash crowds. In *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation* (NSDI), (Boston, MA).

KARGL, F., MAIER, J., AND WEBER, M. 2001. Protecting web servers from distributed denial of service attacks. In *Proceedings of the 10th International World Wide Web Conference*. 130–143.

KENT, S. AND ATKINSON, R. 1998. Security architecture for the Internet protocol. RFC 2401. Internet Engineering Task Force (IETF). Go online to `www.ietf.org`.

KEROMYTIS, A. D., MISRA, V., AND RUBENSTEIN, D. 2002. SOS: Secure overlay services. In *Proceedings of the 2002 ACM SIGCOMM Conference*. 61–72.

KOMPELLA, R. R., SINGH, S., AND VARGHESE, G. 2004. On scalable attack detection in the network. In *IMC '04: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*. ACM Press, New York, NY, 187–200.

KUHN, D., WALSH, T. J., AND FRIES, S. 2005. Security considerations for voice over IP systems. NIST Special Publication 800-58. National Institute of Science and Technology, Gaithersburg, MD.

KULKARNI, A., BUSH, S., AND EVANS, S. 2001. Detecting distributed denial-of-service attacks using Kolmogorov complexity metrics. Tech. rep. 2001CRD176. GE Research & Development Center. Schectades, NY.

LAU, F., RUBIN, S. H., SMITH, M. H., AND TRAJKOVIĆ, L. 2000. Distributed denial of service attacks. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*. Vol. 3. 2275–2280.

LI, J., MIRKOVIC, J., WANG, M., REITHER, P., AND ZHANG, L. 2002. Save: Source address validity enforcement protocol. In *Proceedings of IEEE INFOCOM 2002*. 1557–1566.

LIPSON, H. F. 2002. Tracking and tracing cyber-attacks: Technical challenges and global policy issues. Special rep. CMU/SEI-2002-SR-009. CERT Coordination Center. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

MAHAJAN, R., BELLOVIN, S. M., FLOYD, S., IOANNIDIS, J., PAXSON, V., AND SHENKER, S. 2002. Controlling high bandwidth aggregates in the network. *ACM Comput. Commun. Rev. 32*, 3 (Jul.), 62–73.

MANIKOPOULOS, C. AND PAPAVASSILIOU, S. 2002. Network intrusion and fault detection: A statistical anomaly approach. *IEEE Commun. Mag. 40*, 10 (Oct.), 76–82.

MEASUREMENT. 2005. The measurement factory DNS survey. Go online to `http://dns.measurement-factory.com/surveys/sum1.html`.

MILLEN, J. K. 1992. A resource allocation model for denial of service. In *Proceedings of the IEEE Symposium on Security and Privacy*. 137–147.

MIRKOVIC, J., DIETRICH, S., DITTRICH, D., AND REIHER, P. 2005. *Internet Denial of Service: Attack and Defense Mechanisms*. Prentice Hall, Engle Wood Cliffs, NJ.

MIRKOVIĆ, J., PRIER, G., AND REIHER, P. 2002. Attacking DDoS at the source. In *Proceedings of ICNP 2002* (Paris, France). 312–321.

MIRKOVIC, J. AND REIHER, P. 2004. A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Comput. Commun. Rev. 34*, 2, 39–53.

MIRKOVIC, J., ROBINSON, M., REIHER, P., AND KUENNING, G. 2003. Forming alliance for DDoS defenses. In *Proceedings of the New Security Paradigms Workshop* (NSPW 2003). ACM Press, New York, NY, 11–18.

MOCKAPETRIS, P. 1987a. Domain names—concepts and facilities. RFC 1034. Internet Engineering Task Force (IETF). Go online to `www.ietf.org`.

MOCKAPETRIS, P. 1987b. Domain names—implementation and specification. RFC 1035, the Internet Engineering Task Force (IETF). Go online to `www.ietf.org`.

MOREIN, W. G., STAVROU, A., COOK, D. L., KEROMYTIS, A. D., MISRA, V., AND RUBENSTEIN, D. 2003. Using graphic turing tests to counter automated ddos attacks against web servers. In *Proceedings of the 10th ACM International Conference on Computer and Communications Security* (CCS), (Washington, DC).

MORROW, C. AND GEMBERLING, B. 2001. Blackhole route server and tracking traffic on an IP network. Go online to `http://www.secsup.org/Tracking/`.

NEEDHAM, R. M. 1994. Denial of service: an example. *Commun. ACM 37*, 11, 42–46.

PAPADOPOULOS, C., LINDELL, R., MEHRINGER, J., HUSSAIN, A., AND GOVINDAN, R. 2003. Cossack: Coordinated suppression of simultaneous attacks. In *Proceedings of the 3rd DARPA Information Survivability Conference and Exposition* (DISCEX 2003). Vol. 2. 94–96.

PARK, K. AND LEE, H. 2001a. On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack. In *Proceedings of IEEE INFOCOM 2001*. 338–347.

PARK, K. AND LEE, H. 2001b. On the effectiveness of router-based packet filtering for distributed DoS attack prevention in power-law Internets. In *Proceedings of the 2001 ACM SIGCOMM Conference* (San Diego, California, CA). 15–26.

PAXSON, V. 2001. An analysis of using reflectors for distributed denial-of-service attacks. *ACM Comput. Commun. Rev. 31*, 3 (Jul.), 38–47.

PENG, T., LECKIE, C., AND KOTAGIRI, R. 2004. Proactively detecting distributed denial of service attacks using source ip address monitoring. In *Proceedings of the Third International IFIP-TC6 Networking Conference* (Networking 2004). 771–782.

PENG, T., LECKIE, C., AND RAMAMOHANARAO, K. 2002a. Adjusted probabilistic packet marking for IP traceback. In *Proceedings of the Second IFIP Networking Conference* (Networking 2002). (Pisa, Italy). 697–708.

PENG, T., LECKIE, C., AND RAMAMOHANARAO, K. 2002b. Defending against distributed denial of service attack using selective pushback. In *Proceedings of the 9th IEEE International Conference on Telecommunications* (ICT 2002) (Beijing, China). 411–429.

PENG, T., LECKIE, C., AND RAMAMOHANARAO, K. 2003. Prevention from distributed denial of service attacks using history-based IP filtering. In *Proceeding of the 38th IEEE International Conference on Communications* (ICC 2003) (Anchorage, Alaska). 482–486.

REKHTER, Y. AND LI, T. 1995. A border gateway protocol 4 (BGP-4). RFC 1771. Internet Engineering Task Force (IETF). Go online to `www.ietf.org`.

ROCHLIS, J. A. AND EICHIN, M. W. 1989. With microscope and tweezers: The worm from MIT's perspective. *Commun. ACM 32*, 6, 689–698.

ROSENBERG, J., SCHULZRINNE, H., CAMARILLO, G., JOHNSTON, A., PETERSON, J., SPARKS, R., HANDLEY, M., AND SCHOOLER, E. 2002. SIP: Session initiation protocol. RFC 3261. Internet Engineering Task Force (IETF). Go online to `www.ietf.org`.

SAVAGE, S., WETHERALL, D., KARLIN, A., AND ANDERSON, T. 2000. Practical network support for IP traceback. In *Proceedings of the 2000 ACM SIGCOMM Conference*. 295–306.

SCALZO, F. 2006. Recent dns reflector attacks. VeriSign. Go online to `http://www.nanog.org/mtg-0606/pdf/frank-scalzo.pdf`.

SCHUBA, C. L., KRSUL, I. V., KUHN, M. G., SPAFFORD, E. H., SUNDARAM, A., AND ZAMBONI, D. 1997. Analysis of a denial of service attack on TCP. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*. IEEE Computer Society, IEEE Computer Society Press, Los Alamitos, CA, 208–223.

SISALEM, D., EHLERT, S., GENEIATAKIS, D., KAMBOURAKIS, G., DAGIUKLAS, T., MARKL, J., ROKOS, M., BOTRON, O., RODRIGUEZ, J., AND LIU, J. 2005. Towards a secure and reliable VoIP infrastructure. Tech. rep. D2.1. SNOCER. May.

SNOEREN, A. C., PARTRIDGE, C., SANCHEZ, L. A., JONES, C. E., TCHAKOUNTIO, F., KENT, S. T., AND STRAYER, W. T. 2001. Hash-based IP traceback. In *Proceedings of the 2001 ACM SIGCOMM Conference* (San Diego, CA). 3–14.

SONG, D. X. AND PERRIG, A. 2001. Advanced and authenticated marking schemes for IP traceback. In *Proceedings of IEEE INFOCOM 2001*. 878–886.

SPATSCHECK, O. AND PETERSEN, L. L. 1999. Defending against denial of service attacks in Scout. In *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation*.

STONE, R. 1999. Centertrack: An IP overlay network for tracking DoS floods. In *Proceedings of the 9th USENIX Security Symposium* (Denver, CO).

TUPAKULA, U. AND VARADHARAJAN, V. 2003. A practical method to counteract denial of service attacks. In *Proceedings of the Twenty-Sixth Australasian Computer Science Conference* (ACSC2003) (Adelaide, Australia). 275–284.

US-CERT. 2005. Technical cyber security alert TA05-210A. Cisco IOS IPv6 vulnerability. Go online to `http://www.us-cert.gov/cas/techalerts/TA05-210A.html`.

VAUGHN, R. AND EVRON, G. 2006. DNS amplification attacks. Go online to `http://www.isotf.org/news/DNS-Amplification-Attacks.pdf`.

VIXIE, P. 1999. Extension mechanisms for DNS (EDNS0). RFC 2671. Internet Engineering Task Force (IETF). Go online to `www.ietf.org`.

VIXIE, P., SNEERINGER, G., AND SCHLEIFER, M. 2002. Events of 21-Oct-2002. Go online to `www.isc.org/ops/f-root/october21.txt`.

WALDVOGEL, M. 2002. Gossib vs. IP traceback rumors. In *Proceedings of the 18th Annual Computer Security Applications Conference* (ACSAC 2002).

WANG, H., ZHANG, D., AND SHIN, K. G. 2002. Detecting SYN flooding attacks. In *Proceedings of IEEE INFOCOM 2002*. 1530–1539.

WANG, J. 1999. A survey of Web caching schemes for the internet. *SIGCOMM Comput. Commun. Rev. 29*, 5, 36–46.

WILLIAMS, P. D., ANCHOR, K. P., BEBO, J. L., GUNSCH, G. H., AND LAMONT, G. B. 2001. CDIS: Towards a computer immune system for detecting network intrusions. In *Proceedings of the International Symposium on Recent Advances in Intrusion Detection*. 117–133.

WRIGHT, G. R. AND STEVENS, W. R. 1995. *TCP/IP Illustrated, The Implementation*. Vol. 2. Addison-Wesley, Reading, MA.

WU, S. F., ZHANG, L., MASSEY, D., AND MANKIN, A. 2001. *Intension-Driven ICMP Trace-Back*. IETF Internet Draft. Go online to `www.ietf.org`.

YAU, D. K. Y., LUI, J. C. S., AND LIANG, F. 2002. Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. In *Proceedings of the IEEE International Workshop on Quality of Service* (IWQoS) (Miami Beach, FL). 35–44.

ZHANG, Z., LI, J., MANIKOPOULOS, C., JORGENSON, J., AND UCLES, J. 2001. HIDE: A hierarchical network intrusion detection system using statistical preprocessing and neural network classification. In *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security* (United States Military Academy, West Point, NY).

Received August 2004; revised March 2006, August 2006; accepted November 2006