Chapter 8

Denial-of-Service Attacks

Aikaterini Mitrokotsa and Christos Douligeris

8.1 INTRODUCTION

Availability requires that computer systems function normally without loss of resources to legitimate users. One of the most challenging issues to availability is the denial-of-service (DoS) attack. DoS attacks constitute one of the major threats and among the hardest security problems in today's Internet. The main aim of a DoS is the disruption of services by attempting to limit access to a machine or service. Depending on the attackers' strategy, the target resources may be the file system space, the process space, the network bandwidth, or the network connections. These attacks achieve their goal by sending at a victim a stream of packets in order to exhaust the bandwidth of its network traffic or its processing capacity denying or degrading service to legitimate users. There have been some large-scale attacks targeting high-profile Internet sites [1–3].

Distributed denial-of-service (DDoS) attacks add the many-to-one dimension to the DoS problem, making the prevention and mitigation of such attacks more difficult and the impact proportionally severe. These attacks use many Internet hosts in order to exhaust the resources of the target and cause DoS to legitimate clients. The traffic is usually so aggregated that it is difficult to distinguish legitimate packets from attack packets. More importantly, the attack volume can be larger than the system can handle. There are no apparent characteristics of DDoS streams that could be directly and wholesalely used for their detection and filtering. The attacks achieve their desired effect by sending large amounts of network traffic and by varying packet fields in order to avoid characterization and tracing. Extremely sophisticated, "user-friendly," and powerful DDoS toolkits are available to potential attackers, increasing the danger of becoming a victim in a DoS or a DDoS attack, as essential systems are ill prepared to defend themselves.

The consequences of DoS attacks are extremely serious and financially disastrous, as can be seen by frequent headlines naming the most recent victim of a DoS attack. In February 2001, University of California at San Diego (UCSD) [3] network researchers from the San Diego Supercomputer Center (SDSC) and the Jacobs School of Engineering analyzed the pattern of DoS attacks against the computers of corporations, universities, and private individuals. They proposed a new technique, called "backscatter analysis." This technique estimates the worldwide DoS activity. This research provided the only data

Network Security: Current Status and Future Directions, Edited by C. Douligeris and D. N. Serpanos

quantifying DoS attacks that are available to the public in the Internet and enabled the understanding of the nature of DoS attacks.

The researchers [3] used data sets that were collected and analyzed in a three-week-long period. They assessed the number, duration, and focus of attacks and observed more than 12,000 attacks against more than 5000 targets. The targets of the attacks ranged from well-known e-commerce companies to small ISPs (Internet service providers) and individual personal computers.

In this chapter, we present the state of the art in the DoS field using various types of DoS/DDoS attacks and the defense mechanisms that can be used to combat these attacks.

Following this introduction, the chapter is organized as follows. Section 8.2 investigates first the problem of DoS attacks and then the motivation and defense problems. Section 8.3 introduces the problem of DDoS attacks, gives the basic characteristics of well-known DDoS tools, and presents the various types of DDoS attacks. Section 8.4 presents the various DDoS defense mechanisms, and Section 8.5 concludes the chapter.

8.2 DOS ATTACKS

8.2.1 Basic Characteristics of DoS Attacks

According to the World Wide Web (WWW) Security FAQ [4] a DoS attack can be described as an attack designed to render a computer or network incapable of providing normal services. A DoS attack is considered to take place only when access to a computer or network resource is intentionally blocked or degraded as a result of malicious action taken by another user. These attacks do not necessarily damage data directly or permanently, but they intentionally compromise the availability of the resources.

The most common DoS attacks target the computer network's bandwidth or connectivity. In bandwidth attacks the network is flooded with a high volume of traffic leading to the exhaustion of all available network resources, so that legitimate requests cannot get through, resulting in degraded productivity.

In connectivity attacks a computer is flooded with a high volume of connection requests leading to the exhaustion of all available operating system resources, thus rendering the computer unable to process legitimate user requests.

8.2.2 Types of DoS Attacks

DoS attacks can be divided into five categories based on the attacked protocol level, as illustrated in Figure 8.1 [5]:

1. DoS attacks at the *network device level* include attacks that might be caused either by taking advantage of bugs or weaknesses in software or by exhausting the hardware resources of network devices. One example is caused by a buffer overrun error in the password checking routine. Using this, certain routers [5] could crash if the connection to the router is performed via telnet and extremely long passwords are entered.
2. At the *operating system (OS) level* DoS attacks [5] take advantage of the ways protocols are implemented by OSs. One example in this category is the ping-of-death attack [6]. In this attack, Internet Control Message Protocol (ICMP)

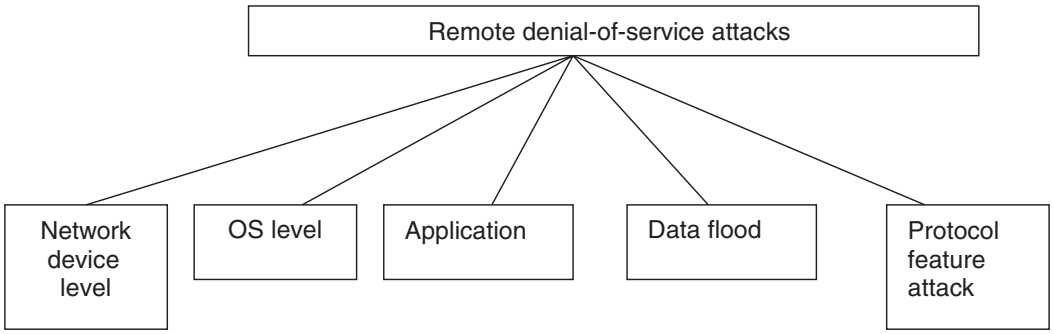

Figure 8.1 Classification of remote DoS attacks.

echo requests having data sizes greater than the maximum Internet Protocol (IP) standard size are sent to the victim. This attack often has the effect of crashing the victim's machine.

3. *Application-based attacks* try to settle a machine or a service out of order either by exploiting bugs in network applications that are running on the target host or by using such applications to drain the resources of their victim. It is also possible that the attacker may have found points of high algorithmic complexity and exploits them in order to consume all available resources on a remote host. One example of an application-based attack is the finger bomb [7]. A malicious user could cause the finger routine to be recursively executed on the victim in order to drain its resources.
4. In *data flooding attacks*, an attacker attempts to use the bandwidth available to a network, host, or device at its greatest extent by sending it massive quantities of data to process. An example is flood pinging. Simple flooding is commonly seen in the form of DDoS attacks, which will be discussed later.
5. DoS attacks *based on protocol features* take advantage of certain standard protocol features. For example, several attacks exploit the fact that IP source addresses can be spoofed. Moreover, several types of DoS attacks attempt to attack the domain name system (DNS) cache on name servers. A simple example of attacks exploiting DNS is when an attacker owning a name server traps a victim name server into caching false records by querying the victim about the attacker's own site. A vulnerable victim name server would then refer to the malicious server and cache the answer [8].

8.2.3 DoS Motivation and Defense Problems

There are several motivations for DoS attacks. Individuals often launch DoS attacks in order to be noticeable and generate publicity. Other attacks are politically motivated. Websites belonging to controversial entities such as government sites have frequently been the targets of DoS attacks. Personal reasons are another motivation for DoS attacks. Individuals may launch attacks based on perceived slights or simply as jokes. Those attacks are generally not very intense and are usually not maintained for very long. DoS attacks have some characteristics that make them very difficult to combat. In the following we present some issues that make protection from DoS attacks very difficult.

1. ***Highly Interdependent Internet Security [9].*** The Internet has few built-in protection mechanisms to deal with DoS attacks. Its design opens security issues that can be exploited by attackers. It is important to note that no matter how secure a host is, it is always under threat while the rest of the Internet is insecure.

2. ***Inherently Difficult to Detect DoS Attacks [10].*** Detecting the origin of DoS attacks is quite difficult. Taking advantage of the stateless nature of the Internet, attackers use IP source address spoofing to hide the identity of the attacking machines and hide their identity behind handler machines. Furthermore, DoS streams do not present any common characteristics of DoS streams that we may use to detect DoS attacks [10]. So the distinction of attack packets from legitimate packets becomes extremely difficult [9].

3. ***Limited Resources [10].*** The large number of packet streams that need to be generated in massive DoS attacks require large amounts of resources. The systems and networks that comprise the Internet are composed of limited resources that can be easily exhausted during the detection of DoS attacks.

4. ***Automated Tools.*** DoS tools are available on the Internet accompanied with instructions that allow easy and effective use even from nontechnically skilled users. The attackers always try to develop more efficient tools in order to bypass security systems developed by system managers and researchers.

5. ***Target-Rich Environment [9].*** There are many hosts and networks in the Internet that are vulnerable and may be exploited and provide fertile ground to launch DoS attacks. Many Internet users do not have sufficient technical skills or are not security conscious and cannot protect their systems against DoS attacks. Moreover the design of an effective DoS system is a difficult task that faces many challenges. The requirements for an effective response to a DoS attack are the following [11]:

- One of the main characteristics of a DoS defense system is the high security. It must be ensured that a DoS defense system cannot be used as a victim of a DoS attack.
- A DoS defense system should be reliable in detecting DoS attacks and have no false positives. However, because this may come at a high cost, we may not be very strict with this requirement.
- A DoS defense system should be efficient in detecting and responding to a DoS attack in order to mitigate the effectiveness of the attack.
- A DoS defense mechanism should be realistic in design and applicable in existing security infrastructures without requiring important changes in the Internet infrastructure.
- A DoS defense mechanism should not require many resources and should have low performance cost to avoid the degradation of the performance of the attacked network.

8.3 DDOS ATTACKS

8.3.1 Defining DDoS Attacks

According to the WWW Security FAQ [4, Section 8, Question 1] on DDoS attacks: "A DDoS attack uses many computers to launch a coordinated DoS attack against one or more targets. Using client/server technology, the perpetrator is able to multiply the effectiveness of the DoS significantly by harnessing the resources of multiple unwitting accomplice

computers, which serve as attack platforms." DDoS attacks deploy in a "distributed" way over the Internet and do not break the victim's system, thus making any traditional security defense mechanism inefficient.

8.3.2 DDoS Strategy

A DDoS attack is composed of four elements, as shown in Figure 8.2 [11]:

- The real attacker.
- The handlers or masters, which are compromised hosts with a special program running on them that makes them capable of controlling multiple agents.
- The attack demon agents or zombie hosts, which are compromised hosts that are running a special program and are responsible for generating a stream of packets toward the intended victim. Attack demons usually are external to both the victim's and attacker's networks in order to avoid both an efficient response that might stop the attack and the traceback of the attacker, respectively.
- A victim or target host.

The following steps take place while preparing and conducting a DDoS attack:

1. ***Selection of Agents.*** The attacker chooses the agents that will perform the attack. The selection of the agents is based on the existence of vulnerabilities in those machines that can be exploited by the attacker in order to gain access to them. It is important that the agents should have enough resources to be able to generate powerful attack streams.

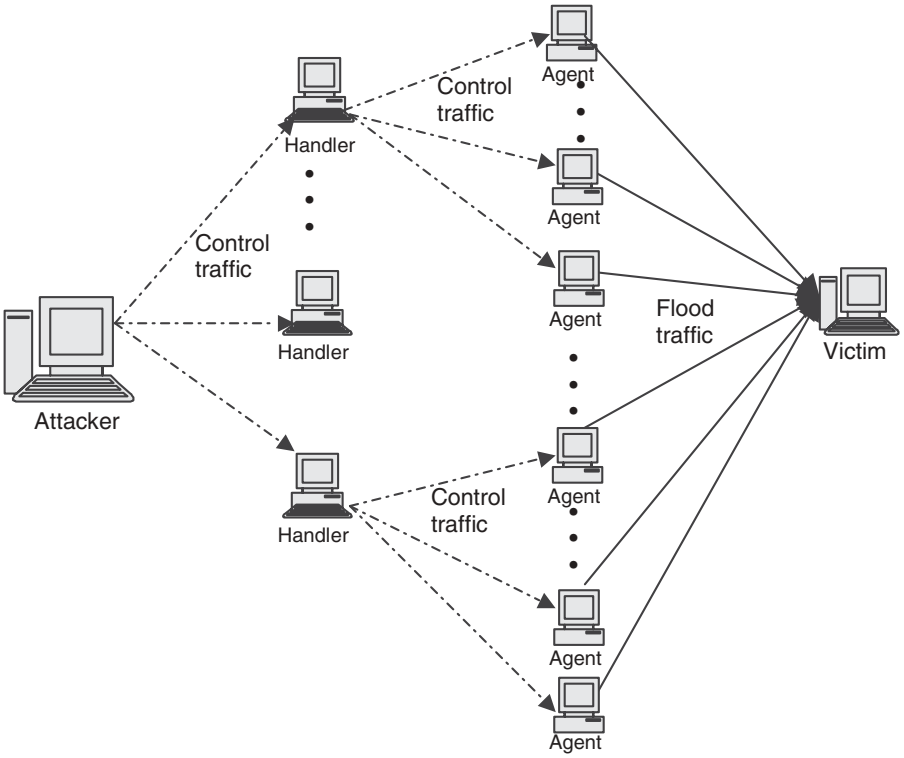

Figure 8.2 Architecture of DDoS attacks.

At first agent selection was a manual procedure but with automatic scanning tools this has become much easier.

2. ***Compromise.*** The attacker exploits the security holes and vulnerabilities of the agent machines and plants the attack code. Furthermore the attacker tries to protect the code from discovery and deactivation. Self-propagating tools such as the Ramen worm and Code Red soon automated this phase. The people who use the agent systems do not know that their systems are compromised and used for the launch of a DDoS attack [12]. When participating in a DDoS attack, agent programs consume little resources, which means the users of computers experience minimal change in performance.

3. ***Communication [12].*** Before the attacker initiates the attack, he or she communicates with the handlers to find out which agents can be used in the attack, if it is necessary to upgrade the agents, and the best time to schedule the attack. Agents are able to communicate either with one or multiple handlers depending on the configuration of the DDoS attack network. The protocols that are used for communication between handlers and agents are Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and ICMP.

4. ***Attack.*** At this step the attacker commands the onset of the attack. The victim, the duration of the attack, and special features of the attack, such as the type, length, time to live (TTL), and port numbers, can be adjusted. The variety of the properties of attack packets can help the attacker in avoiding detection.

The latest generation of DDoS attacks do not wait for a trigger from the aggressor but instead monitor a public location on the Internet. For example, a chat room could be monitored and the attack may start automatically as soon as a particular key word or phrase is typed. In this way the aggressor is more or less untraceable. More frighteningly still the trigger word or phrase could be set as one commonly used and then the aggressor would need to take no action for an attack to take place.

Recently a multiuser, online chatting system known as Internet Relay Chat (IRC) channels are used to achieve communication between the agents and the attacker [13], since IRC networks can be used so that their users will be able to communicate in public, private, and secret channels. IRC-based DDoS attack network and agent–handler DDoS attack models have many similarities. According to [12] the distinguishing difference between them is that instead of a handler program an IRC server is used to learn the addresses of connected agents and handlers. The main advantage of IRC-based attack networks over agent–handler attack models is that, although the identity of a singe participant may be discovered and this would lead to the discovery of the communication channels, the identities of the other participants will not be revealed. Moreover IRC-based DDoS attack models provide anonymity, making even more difficult the detection of the agents and the source of the attack. Furthermore, the attacker is notifed by the agent software about the situation of the idle or running agents as well as of all lists of available agents.

8.3.3 DDoS Tools

There are several known DDoS attack tools. The architecture of these tools is very similar, and in fact some tools have been constructed through minor modifications of other tools. In this section, we present the functionality of some of these tools. For presentation purposes we divide them into agent-based and IRC-based DDoS tools.

Agent-based DDoS tools are based on the agent–handler DDoS attack model, which consists of handlers, agents, and victims as described in Section 8.3.2. Some well-known agent-based DDoS tools are *Trinoo*, *Tribe Flood Network (TFN)*, *TFN2K*, *Stacheldraht*, *mstream*, and *Shaft*.

Trinoo [14] is the most known and mostly used DDoS attack tool. It has been able to achieve bandwidth depletion and can be used to launch UDP flood attacks against one or many IP addresses. Shaft [15] is a DDoS tool similar to Trinoo and is able to launch packet flooding attacks. Shaft has the ability to control the duration of the attack as well as the size of the flooding packets [16].

TFN [17] is a DDoS attack tool that is able to perform bandwidth depletion and resource depletion attacks. It is able to implement Smurf, UDP flood, TCP SYN flood, ICMP echo request flood, and ICMP directed broadcast. TFN2K [15] is a derivative of TFN and is able to implement Smurf, SYN, UDP, and ICMP flood attacks. TFN2K has the special feature of being able to add encrypted messages between all attack components. *Stacheldraht* [18] (German for "barbed wire") is based on early versions of TFN attempts to eliminate some of its weak points and implement Smurf, SYN flood, UDP flood, and ICMP flood attacks. Mstream [19] is a simple point-to-point TCP ACK flooding tool that is able to overwhelm the tables used by fast routing routines in some switches.

IRC-based DDoS attack tools were developed after the appearance of the agent–handler attack tools. This resulted in many more sophisticated IRC-based tools which include some important features that can be found in many agent–handler attack tools. One of the best known IRC-based DDoS tools is *Trinity*. In addition to the now well-known UDP, TCP SYN, TCP ACK, and TCP NUL packet floods, Trinity v3 [20] introduces TCP random flag packet floods, TCP fragment floods, TCP established floods, and TCP RST packet floods. In the same generation as Trinity are myServer [15], which relies on external programs to provide DoS, and Plague [15], which provides TCP ACK and TCP SYN flooding. Knight [21] is another IRC-based DDoS attack tool that is very light-weight and powerful and able to perform UDP flood attacks, SYN attacks, and an urgent pointer flooder [12]. An IRC-based DDoS tool that is based on Knight is Kaiten [15], which includes UDP, TCP flood attacks, SYN, and PUSH+ACH attacks.

8.3.4 Types of DDoS Attacks

To understand DDoS attacks it is necessary to understand the various types of DDoS attacks. Figure 8.3 illustrates the various types of DDoS attacks in a two-level structure. In the first level, attacks are divided according to their degree of automation, exploited vulnerability, attack rate dynamics, and impact. In the second level specific characteristics of each first-level category are recognized. A more detailed classification of DDoS attacks can be found in [22, 23].

8.3.4.1 DDoS Attacks by Degree of Automation

Based on the degree of automation of the attack, DDoS attacks can be divided into manual, semiautomatic, and automatic attacks.

The early DDoS attacks were *manual*. This means that the DDoS strategy included the scanning of remote machines for vulnerabilities, breaking into them, and installing the attack code. All of these steps were later automated by the use of semiautomatic DDoS attacks and automatic DDoS attacks.

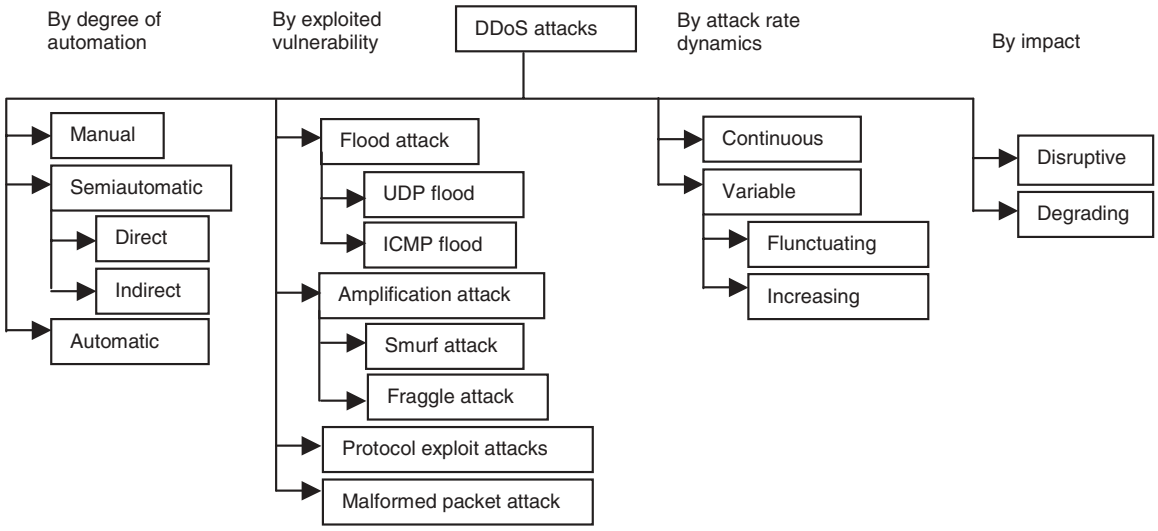

Figure 8.3 DDoS attacks.

Semi automatic attacks belong in the agent–handler attack model, and the attacker scans and compromises the handlers and agents by using automated scripts. The attack type, the victim's address, and the onset of the attack are specified by the handler machines. Semiautomatic attacks can be divided further into attacks with direct communication and attacks with indirect communication. Attacks with direct communication include attacks during which it is necessary for the agent and handler to know each other's identity in order to communicate. This approach includes the hard coding of the IP address of the handler machines. The main drawback of this approach is that if the identity of one compromised host is revealed the whole DDoS network may be exposed. In contrast, attacks with indirect communication achieve greater survivability. Examples of this kind of attack are the IRC-based DDoS attacks discussed in the previous section.

In *automatic DDoS attacks* the attacker and agent machines do not need to communicate. In most cases the attack phase is limited to a single command. All the features of the attack (e.g., the attack type, the duration, and the victim's address) are preprogrammed in the attack code. This way, the attacker has minimal exposure and the possibility of revealing his or her identity is small. The drawback of this approach is that the propagation mechanisms may leave the compromised machine vulnerable, making possible the gain of access and modification of the attack code.

8.3.4.2 DDoS Attacks by Exploited Vulnerability

DDoS attacks according to exploited vulnerability can be divided into the following categories: flood attacks, amplification attacks, protocol exploit attacks, and malformed packet attacks.

In a *flood attack*, the agents send a vast amount of IP traffic to a victim system in order to congest the victim system's bandwidth. The impact of packet streams sent by the agents to the victim varies from slowing it down or crashing the system to saturation of the network bandwidth. Some of the well-known flood attacks are UDP flood attacks and ICMP flood attacks:

A UDP flood attack is possible when a large number of UDP packets are sent to a victim system. This results in saturation of the network and depletion of available bandwidth for valid service requests to the victim. A UDP flood attack is possible when an attacker sends a UDP packet to a random port on the victim system. When the victim system receives a UDP packet, it determines the application waiting on the destination port. When it realizes that there is no application waiting on the port, it will generate an ICMP packet of "destination unreachable" [24] to the forged source address. If enough UDP packets are delivered to the ports of the victim, the system will go down. Using a DDoS tool the source IP address of the packets sent by the attacker can be spoofed, the true identity of the secondary victim is prevented from exposure, and the packets returned from the victim system are not sent back to the agents of the attacker.

ICMP flood attacks exploit the ICMP, which enables users to send an echo packet to a remote host to check whether it is alive. More specifically during a DDoS ICMP flood attack the agents send a large number of ICMP_ECHO_REPLY packets ("ping") to the victim. These packets request a reply from the victim, which results in the saturation of the bandwidth of the victim's network connection [14]. During an ICMP flood attack the technique of IP spoofing is used.

In *amplification attacks* the attacker or the agents exploit the broadcast IP address feature that most routers have. This feature is exploited in order to achieve amplification and reflection of attacks by sending messages to broadcast IP addresses. This results in all the routers that are in the network sending the packets to all the IP addresses that are in the broadcast range [12]. This way the malicious traffic that is produced reduces the victim system's bandwidth. In this type of DDoS attack, the broadcast message can be sent directly or by the use of agents, so that the attack traffic generated will have greater volume. If the broadcast message is sent directly, the attacker can use the hosts that belong in the broadcast network as agents without filtering or installing attack code in agents. Some well-known amplification attacks are Smurf and Fraggle.

The intermediary nodes that are used as attack launchers in amplification attacks are called reflectors [25]. A reflector is any IP host that will return a packet if it has received a packet. Web servers, DNS servers, and routers are reflectors because they return SYN ACKs or Reset connection (RST) after receiving a SYN or other TCP packets.

During an amplification attack the attacker sends spoofed packets that require responses to the reflectors. The source addresses of the packets are spoofed with the address of the victim. After receiving the spoofed packets, the reflectors respond to the victim accordingly. The attack packets are essentially reflected in the normal packets toward the victim. Apparently, if the number of reflected packets is extremely large, then the victim's link can be flooded. We should note that the reflectors are identified as the origin of the reflected packets that flood the victim. Moreover, it is extremely difficult for the operator of a reflector to locate the compromised slave that is exploiting the reflector because the traffic sent to the reflector does not include the origin (source address) of the slave but rather includes the origin of the victim. The main characteristics that differentiate an amplification attack from a direct one are the following [26]:

- In an amplification attack some predetermined reflectors are necessary.
- The reflectors may be dispersed on the Internet because it is not necessary for the attacker to install any agent software.

- The packets sent from the reflectors are normal packets with legitimate origin and thus cannot be captured and eliminated through filtering and route-based mechanisms.

Smurf attacks send ICMP echo request traffic with a spoofed origin [27] of the victim to some IP broadcast addresses. On an IP network most hosts accept ICMP echo requests and reply to the origin of these requests. In the case of DoS attacks the source address is the address of the target victim. In the case of a broadcast network there could be hundreds of replies to each ICMP packet. The use of a network to send many responses to a packet is called "amplifier" [28]. In this type of attack the party that is hurt is not only the victim but also reflectors [29]. Fraggle attacks are similar to Smurf attacks except that instead of using ICMP echoes they use UDP echo packets. Fraggle attacks can have even more severe impact than Smurf attacks.

Protocol exploit attacks [22] exploit a specific feature or implementation bug of some protocol that has been installed in the victim's system to achieve the exhaustion of available resources. An example of protocol exploit attacks is the TCP SYN attack.

TCP SYN attacks exploit the weakness of the three-way handshake in the TCP connection setup. A server, after receiving an initial SYN request from a client, responds with a SYN/ACK packet and waits for the final ACK of the client. A SYN flooding attack is initiated by sending a large number of SYN packets and never acknowledging any of the replies, while the server is waiting for the nonexistent ACKs. The server has a limited buffer queue for new connections; this results in a server with a full buffer queue that is unable to process legitimate connections [29].

Malformed packet attacks [12] rely on incorrectly formed IP packets that are sent from agents to the victim that will lead to the crash of the victim's system. Malformed packet attacks can be divided into *IP address attack* and *IP packet options attack*. In an IP address attack, the packet has the same source and destination IP addresses. This results confusing the OS of the victim and the system crashes. A special characteristic of malformed packets that is exploited for the launch of IP packet options attacks is that it is able to randomize the optional fields of an IP packet and make all quality-of-service bits equal to 1. This results in the need for additional processing time by the victim to analyze the traffic. The combination of this attack with the use of multiple agents, could lead to the crash of the victim's system.

8.3.4.3 DDoS Attacks by Attack Rate Dynamics

Depending on the attack rate dynamics DDoS attacks can be divided into continuous-rate and variable-rate attacks [22]:

Continuous-rate attacks comprise attacks that after the onset of the attack are executed with full force and without a break or decrement of force. The impact of such an attack is very quick.

Variable-rate attacks, as their name indicates, "vary the attack rate" and thus avoid detection and immediate response. Variable-rate attacks may be divided into fluctuating-rate and increasing-rate attacks. Fluctuating-rate attacks have a wavy rate that is defined by the victim's behavior and response to the attack, at times decreasing the rate to avoid detection. Increasing-rate attacks gradually lead to the exhaustion of a victim's resources, something that may delay detection of the attack.

8.3.4.4 DDoS Attacks by Impact

Based on the impact of a DDoS attack, we can divide DDoS attacks into disruptive and degrading attacks [22]:

Disruptive attacks lead to complete denial of the victim's service to its clients.

In *degrading attacks* the main goal of the attacker is not to exhaust but only to consume some portion of a victim's resources. This results in delay of the detection of the attack and much damage to the victim's system.

8.4 DDoS DEFENSE MECHANISMS

There are many DDoS defense mechanisms. We present them using two criteria: the activity deployed by the attacked and the location deployment of the attack. DDoS defense mechanisms according to the activity deployed can be divided into the following four categories:

- Intrusion prevention
- Intrusion detection
- Intrusion response
- Intrusion tolerance and mitigation

The second criterion, location deployment of the attack, results in the following three categories of defense mechanisms:

- Victim network
- Intermediate network
- Source network

DDoS mechanisms are illustrated in Figure 8.4. In the following, we discuss the techniques used in each DDoS defense mechanism category.

8.4.1 DDoS Defense Mechanisms by Activity

8.4.1.1 Intrusion Prevention

The best mitigation strategy against any attack is to completely prevent the attack. In this stage we try to stop DDoS attacks from being launched in the first place. There are many DDoS defense mechanisms that try to prevent systems from attacks:

- *Using globally coordinated filters*, attacking packets can be stopped before they cause serious damage. There are many filtering mechanisms that can be used, including *ingress filtering*, *egress filtering*, *route-based distributed packet filtering, history-based IP* (HIP) *filtering*, and *secure overlay services (SOSs).*

  In ingress filtering [30] a router is set up to block out of the network incoming packets with illegitimate origin.

  Egress filtering [31] is a filtering method on outbound traffic, which allows packets only from a specific set of IP addresses to leave the network.

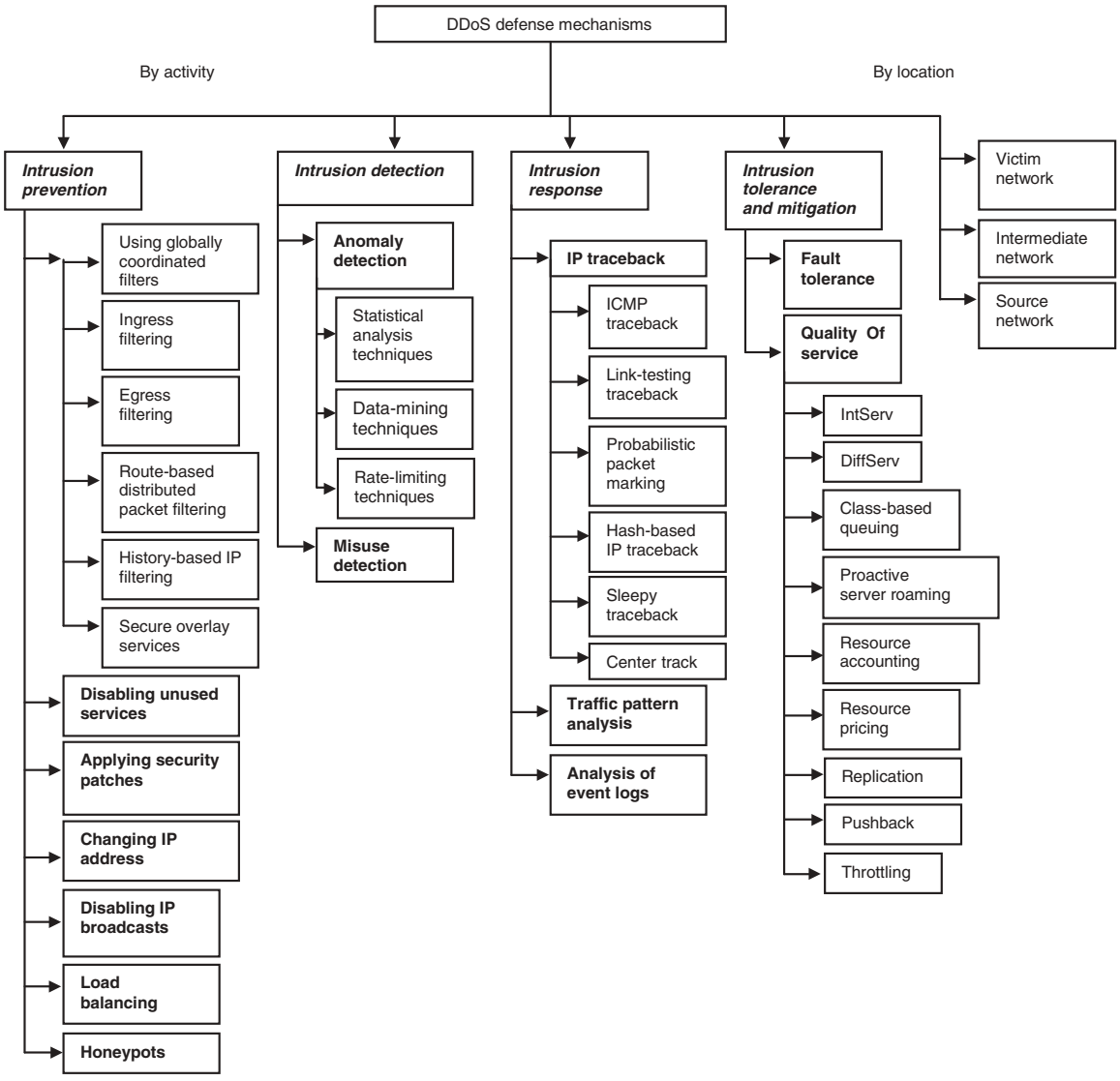

Figure 8.4 DDoS defense mechanisms.

Route-based distributed packet filtering [32] is an approach capable of filtering out a large portion of spoofed IP packets and preventing attack packets from reaching their targets as well as to help in IP traceback.

In (HIP) filtering [33] the edge router admits the incoming packets according to a prebuilt IP address database.

SOS [34] is an architecture in which only packets coming from a small number of nodes, called servlets, are assumed to be legitimate client traffic that can reach the servlets through hash-based routing inside an overlay network.

- *Disabling unused services* [35] is another approach to prevent DDoS attacks. If network services are not needed or unused, the services should be disabled to prevent attacks. For example, if UDP echo is not required, disabling this service will make the system more secure against this kind of attack.

- *Applying security patches* [35] can armor the hosts against DDoS attacks. Every computer host should update according to the latest security patches and use all the available security mechanisms to combat DDoS attacks.
- *Changing the IP address* [35] is a simple way to guard against a DDoS attack. This technique is called "moving the target defense." All Internet and edge routers are informed when the IP address is changed in order to drop malicious packets. This option can be used only for local DDoS attacks based on IP addresses. However, attackers can render this technique useless by adding a DNS tracing function to the DDoS tool.
- By *disabling IP broadcasts* [29], we can prevent the use of host computers as reflectors in Smurf and ICMP flood attacks. We should make clear that this intrusion prevention mechanism can be effective only if all the neighboring networks have also disabled IP broadcasts.
- *Load balancing* [12] is a simple approach that enables network providers to increase the provided bandwidth on critical connections and prevent their crash in case an attack is launched against them. Additional failsafe protection can be the replication of servers in case some crash during a DDoS attack.
- *Honeypots* [36] can be used to prevent DDoS attacks. Honeypots are not very secure systems and can be used to trick the attacker to attack the honeypot instead of the system being protected. Honeypots may be used not only for the protection of systems but also to gain some extra information about the actions of the attackers [11]. Honeypots are based on the idea of luring the attacker into believing that he or she has successfully compromised the system (e.g., honeypot), causing the attacker to install either the handler or the agent code that is in the honeypot. Thus, systems can be protected from possible DDoS attacks.

Prevention approaches offer a first line of defense against DDoS attacks. A second line of defense, intrusion detection, will be discussed in the next section.

8.4.1.2 Intrusion Detection

Intrusion detection can be used to guard a host computer or network against being a source or a victim of an attack. Intrusion detection systems detect DDoS attacks either by using a priori knowledge of the types of known attacks (signatures) or by recognizing deviations from normal system behaviors.

Anomaly detection relies on detecting behaviors that are abnormal with respect to some normal standard. Many anomaly detection systems and approaches have been developed to detect the faint signs of DDoS attacks. NOMAD [37] is a scalable network monitoring system that is able to detect network anomalies by making statistical analysis of IP packet header information. Other anomaly-based detection mechanisms use Management Information Base (MIB) data from routers [38], congestion-triggered packet sampling and filtering [39], data mining techniques and rate limiting techniques like D-WARD and MULTOPS.

Misuse detection uses a priori knowledge on intrusions and tries to detect attacks based on specific patterns or signatures of known attacks. These patterns are defined as intrusion signatures. Although misuse detection systems are very accurate in detecting known attacks, their basic drawback is that attacks are under continuous evolution and this leads to the need for an up-to-date knowledge base of attacks. Several popular network monitors perform signature-based detection, such as CISCO's NetRanger, NID, SecureNet

PRO, RealSecure, NFRNID, and Snort. Intrusion detection systems are discussed in Chapter 6.

8.4.1.3 Intrusion Response

Once an attack is identified, the next step is to identify the origin and block its traffic accordingly. The blocking part is usually performed under manual control (e.g., by contacting the administrators of upstream routers and enabling access control lists) since an automated response system might cause further service degradation in response to a false alarm. Automated intrusion response systems are deployed only after a period of self-learning (for the ones that employ neural computation in order to discover the DDoS traffic) or testing (for the ones that operate on static rules). There are many approaches that target the tracing and identifying of the real attack source.

IP traceback traces the attacks back to their origin, so one can find the true identity of the attacker and achieve detection of asymmetric routes as well as path characterization. Some factors that render IP traceback difficult is the stateless nature of Internet routing and the lack of source accountability in TCP/IP. For efficient IP traceback, it is necessary to compute and construct the attack path. At a very basic level, IP traceback can be thought of as a process that is performed manually in which the administrator of the network that is the victim of an attack calls the ISP in order to be informed of the direction from which the packets are coming. Because of the difficulty of the manual traceback, there have been many proposals that try to make this process easier and automatic, for example, ICMP traceback [40], link-testing traceback [41], probabilistic packet marking (PPM) [42], hash-based IP traceback [43], Sleepy Traceback [44], and CenterTrack [45].

Traffic pattern analysis [12] is another way to respond to DDoS attacks. During a DDoS attack, traffic pattern data can be stored and then analyzed after the attack in order to find specific characteristics and features that may indicate an attack. The results from this analysis of data can be used to update load balancing and throttling techniques as well as develop new filtering mechanisms that prevent DDoS attacks.

Analysis of event logs [12] is another good approach that targets the response to DDoS attacks. The selection of event logs recorded during the setup and the execution of the attack can be used to discover the type of DDoS attacks and do a forensic analysis. Network equipment such as firewalls, packet sniffers, server logs, and honeypots [36] can be used in the selection of event logs.

8.4.1.4 Intrusion Tolerance and Mitigation

Research on intrusion tolerance accepts that it is impossible to prevent or stop DDoS attacks completely and focuses on minimizing the attack impact and maximizing the quality of its services. Intrusion tolerance can be divided into two categories: fault tolerance and quality of service (QoS).

Fault tolerance is a research area whose designs are built in critical infrastructures and applied in three levels: hardware, software, and system [46]. The idea of fault tolerance is that by duplicating the network's services and employing different access points, the network can continue offering its services when flooding traffic congests one network link.

QoS describes the ability of a network to deliver predictable results for some applications. Many intrusion-tolerant QoS techniques and intrusion-tolerant QoS systems have been developed to mitigate DDoS attacks.

Among intrusion-tolerant QoS techniques integrated (IntServ) and differentiated services (DiffServ) represent the principal architectures [47]. Queuing techniques are also employed to combat DDoS attacks. The oldest and most widely applied queuing technique is class-based queuing (CBQ). CBQ [48] sets up different traffic queues for different types of packets. An amount of outbound bandwidth can then be assigned to each queue. Other intrusion-tolerant QoS systems are VIPnets [49], proactive server roaming, resource accounting, resource pricing, pushback, and throttling.

8.4.2 DDoS Defense Mechanisms by Deployment Location

Based on the deployment location, DDoS defense mechanisms are divided into those deployed at the victim, intermediate, and source networks.

Most systems deployed to defend against DDoS attacks have been designed to work on the victim's network, since this will suffer the most from an attack. The victim is the one that needs to be protected against a DDoS attack so it is the one that should deploy a DDoS defense system [11]. An example is EMERALD [50]. Such a system will increase a victim's ability to recognize that it is the target of an attack as well as to gain more time to respond. Note, however, that to achieve increased security the victim's network will sacrifice some of its performance and resources.

DDoS defense mechanisms deployed at the intermediate network are more effective than a victim network mechanisms since the attack traffic can be handled easily and find the origin of the attack. An example is WATCHERS [51]. However, these defense mechanisms present several disadvantages that prevent their wide deployment, such as the increase of the intermediate network's performance and the greater difficulty to detect the attack since the intermediate network usually is not affected.

DDoS defense mechanisms deployed at the source network may stop attack flows before they enter the Internet core. This means that it is easier to defend against them before they aggregate with other attack flows. Moreover, being close to the source makes it easier to trace back to the origin of the attack. A source network mechanism has the same disadvantage as the intermediate network mechanism of detecting the occurrence on an attack, since it does not experience any difficulties. This disadvantage can be balanced by its ability to sacrifice some of its resources and performance in order to achieve better DDoS detection. However, the main disadvantage of such a system is that in the case of unreliable attack detection legitimate traffic might be restricted.

8.5 CONCLUSIONS

Undoubtedly, DoS attacks are a serious problem on the Internet and their rate of growth and wide acceptance challenge the general public, a skeptical government, and businesses. It is clear that the wave of DoS attacks will continue to pose a significant threat; as new countermeasures are developed, new DoS attack modes will emerge. Since DoS attacks are complex and difficult to combat, there is no single-point solution; everyone is vulnerable and everyone's security is intertwined. A network infrastructure must be both robust

enough to survive direct DoS attacks and extensible enough to adapt and embrace new defenses against emerging and unanticipated attack modes.

We need to confront DoS attacks as a problem that requires a long-term effort in order to implement effective solutions. *Consensus Roadmap for Defeating Distributed Denial of Service Attacks* [52] identifies some actions that will help us defend against DoS attacks more effectively in the distant future. Between them is included the accelerated adoption of the IPsec components of Internet Protocol Version 6 and Secure Domain Name System. Furthermore, increased emphasis should be given on security in the research and development of Internet II. Moreover, we should encourage vendors to automate security updating for their clients. Thus it will be easier to be up to date in security issues. Furthermore, the research and development of safer operating systems are necessary as well as continued research in anomaly based and other forms of intrusion detection. In addition, we should not forget to consider changes in government procurement policy that will emphasize the security and safety of information systems.

REFERENCES

1. CERT Coordination Center, Denial of service attacks, http://www.cert.org/tech_tips/denial_of_service.html.
2. Computer Security Institute and Federal Bureau of Investigation, CSI/FBI computer crime and security survey 2001, http://www.gocsi.com, Mar. 2001.
3. D. Moore, G. Voelker, and S. Savage, Inferring Internet denial of service activity, in *Proceedings of the USENIX Security Symposium*, Washington, DC, 2001, pp. 9–22.
4. L. D. Stein and J. N. Stewart, The World Wide Web Security FAQ, version 3.1.2, http://www.w3.org/Security/Faq, Feb. 4, 2002.
5. D. Karig and R. Lee, Remote denial of service attacks and countermeasures, Technical Report CE-L2001-002, Department of Electrical Engineering, Princeton University, Princeton, NJ, October 2001.
6. M. Kenney, Malachi, ping of death, http://www.insecure.org/sploits/ping-o-death.html, Jan. 1997.
7. Finger bomb recursive request, http://xforce.iss.net/static/47.php.
8. D. Davidowicz, Domain name system (DNS) security, http://compsec101.antibozo.net/papers/dnssec/dnssec.html, 1999.
9. P. Zaroo, A survey of DDoS attacks and some DDoS defense mechanisms, Advanced Information Assurance (CS 626), http://www.cs.uidaho.edu/~visakhr/ddos_paper.pdf. 2002.
10. D. Xuan, R. Bettati, and W. Zhao, A gateway-based defense system for distributed DoS attacks in high-speed networks, in *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, W1A2 0900, United States Military Academy, West Point, NY, June 5–6, 2001.
11. J. Mirkovic, D-WARD: DDoS network attack recognition and defense, PhD disseration prospectus, UCLA, Jan. 23, 2002.
12. S. Specht and R. Lee, Taxonomies of distributed denial of service networks, attacks, tools and countermeasures, Technical Report CE-L2003-03, *Princeton University*, http://www.princeton.edu/~rblee/ELE572_F04Readings.html.
13. J. Lo et al., An IRC tutorial, http://www.irchelp.org/irchelp/irctutorial.html, 1998.
14. P. J. Criscuolo, Distributed denial of service Trin00, Tribe Flood Network, Tribe Flood Network 2000, and Stacheldraht CIAC-2319, Department of Energy Computer Incident Advisory (CIAC), UCRL-ID-136939, Rev. 1, Lawrence Livermore National Laboratory, http://ftp.se.kde.org/pub/security/csir/ciac/ciacdocs/ciac2319.txt, Feb. 14, 2000.
15. S. Dietrich, N. Long, and D. Dittrich, Analyzing distributed denial of service tools: The Shaft case, in *Proceedings of the Fourteenth Systems Administration Conference (LISA 2000)*, New Orleans, LA, Dec. 3–8, 2000, pp. 329–339.
16. G. C. Kessler, Defenses against distributed denial of service attacks, http://www.garykessler.net/library/ddos.html, Nov. 2000.
17. D. Dittrich, The Tribe Flood Network distributed denial of service attack tool, University of Washington, http://staff.washington.edu/dittrich/misc/trinoo.analysis.txt, Oct. 21, 1999.
18. D. Dittrich, The "Stacheldraht" distributed denial of service attack tool, University of Washington, http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt, Dec. 1999.
19. R. Farrow, DDoS is neither dead nor forgotten, *Network Magazine*, http://www.networkmagazine.com/shared/article/showArticle.jhtml?articleId=8703018&pgno=1, 2001.
20. B. Hancock, Trinity v3, A DDoS tool, hits the streets, *Computers Security*, 19(7):574, 2000.

21. Bysin, Knight.c sourcecode, PacketStormSecurity.nl, http://packetstormsecurity.nl/distributed/knight.c, July 11, 2001.
22. J. Mirkovic and P. Reiher, A taxonomy of DDOS attacks and defense mechanisms, *ACM SIGCOMM Computer Communication Review*, 34(2):39–53, Apr. 2004.
23. C. Douligeris and A. Mitrokotsa, DDoS attacks and defense mechanisms: Classification and state-of-the-art, *Computer Networks*, 5(44):643–666, Apr. 2004.
24. K. J. Houle and G. M. Weaver, Trends in denial of service attack technology, CERT and CERT Coordination Center, Carnegie Mellon University, http://www.cert.org/archive/pdf/DoS_trends.pdf, Oct. 2001.
25. V. Paxson, An analysis of using reflectors for distributed denial of service attacks, *ACM Computer Communication Review*, 31(3):38–47, 2001.
26. R. K. C. Chang, Defending against flooding-based, distributed denial of service attacks: A tutorial, *IEEE Communications Magazine*, 40(10):42–51, 2002.
27. Daemon9, route, infinity, IP-spoofing demystified: Trust relationship exploitation, *Phrack Magazine*, Guild Productions, http://www.citi.umich.edu/u/provos/security/ph48.txt, June 1996.
28. C. A. Huegen, The latest in denial of service attacks: Smurfing description and information to minimize effects, http://www.pentics.net/denial-ofservice/white-papers/smurf.cgi, 2000.
29. F. Lau, S. H. Rubin, M. H. Smith, and L. Trajkovic, Distributed denial of service attacks, in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, Nashville, TN, 2000.
30. P. Ferguson and D. Senie, Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing, RFC 2827, Internet Engineering Task Force, www.ietf.org, 2001.
31. Global Incident Analysis Center, Special notice—Egress filtering, http://www.sans.org/y2k/egress.htm.
32. K. Park and H. Lee, On the effectiveness of route-based packet filtering for Distributed DoS attack prevention in power law Internets, in *Proceedings of the ACM SIGCOMM_01 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ACM Press, New York, 2001, pp. 15–26.
33. T. Peng, C. Leckie, and K. Ramamohanarao, Protection from distributed denial of service attack using history-based IP filtering, in *Proceedings of IEEE International Conference on Communications (ICC 2003)*, Anchorage, AL, 2003.
34. A. Keromytis, V. Misra, and D. Rubenstein, SoS: Secure overlay services, in *Proceedings of the ACM SIGCOMM _02 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ACM Press, New York, 2002, pp. 61–72.
35. X. Geng and A. B. Whinston, Defeating distributed denial of service attacks, *IEEE IT Professional*, 2(4):36–42, 2000.
36. N. Weiler, Honeypots for distributed denial of service, in *Proceedings of the Eleventh IEEE International Workshops Enabling Technologies: Infrastructure for Collaborative Enterprises 2002*, Pitsburgh, PA, June 2002, pp. 109–114.
37. R. R. Talpade, G. Kim, and S. Khurana, NOMAD: Traffic based network monitoring framework for anomaly detection, in *Proceedings of the Fourth IEEE Symposium on Computers and Communications*, Athens, 1998.
38. J. B. D. Cabrera, L. Lewis, X. Qin, W. Lee, R. K. Prasanth, B. Ravichandran, and R. K. Mehra, Proactive detection of distributed denial of service attacks using MIB traffic variables—A feasibility study, in *Proceedings of the Seventh IFIP/IEEE International Symposium on Integrated Network Management*, Seattle, WA, May 14–18, 2001.
39. Y. Huang and J. M. Pullen, Countering denial of service attacks using congestion triggered packet sampling and filtering, in *Proceedings of the Tenth International Conference on Computer Communiations and Networks*, Scottsdale, Arizoua, 2001.
40. S. Bellovin, The ICMP traceback message, Network Working Group, Internet draft, http://www.research.att.com/~smb/papers/draft-bellovin-itrace-00.txt, Mar. 2000.
41. H. Burch and H. Cheswick, Tracing anonymous packets to their approximate source, in *Proceedings of USENIX LISA (New Orleans) Conference*, New Orleans, 2000, pp. 319–327.
42. S. Savage, D. Wetherall, A. Karlin, and T. Anderson, Network support for IP traceback, *IEEE/ACM Transactions on Networking*, 9(3):226–237, 2001.
43. A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer, Hash-based IP traceback, in *Proceedings of the ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, ACM Press, New York, 2001, pp. 3–14.
44. X. Wang, D. S. Reeves, S. F. Wu, and J. Yuill, Sleepy watermark tracing: An active network-based intrusion response framework, in *Proceedings of the Sixteenth International Conference of Information Security (IFIP/SEC_01)*, Paris, June 2001.
45. R. Stone, CenterTrack: An IP overlay network for tracking DoS floods, in *Proceedings of the Ninth USENIX Security Symposium*, Denver, CO, Aug. 14–17, 2000, pp. 199–212.
46. National Institute of Standards and Technology, A conceptual framework for system fault tolerance, http://hissa.nist.gov/chissa/SEI_Framework/framework_1.html, 1995.
47. W. Zhao, D. Olshefski, and H. Schulzrinne, Internet quality of service: An overview, Technical Report CUCS-003-00, Columbia University, New York, 2000.

48. F. KARGL, J. MAIER, and M. WEBER, Protecting web servers from distributed denial of service attacks, in *Proceedings of the Tenth International Conference on World Wide Web*, Hong Kong, May 1–5, 2001, pp. 514–524.
49. J. BRUSTOLONI, Protecting electronic commerce from distributed denial of service attacks, in *Proceedings of the Eleventh International World Wide WebConference*, ACM, Honolulu, HI, 2002, pp. 553–561.
50. P. A. PORRAS and P. G. NEUMANN, EMERALD: Event monitoring enabling responses to anomalous live disturbances, in *Proceedings of the Nineteenth National Computer Security Conference*, Baltimore, MD, Oct. 22–25, 1997, pp. 353–365.
51. K. A. BRADLEY, S. CHEUNG, N. PUKETZA, B. MUKHERJEE, and R. A. OLSSON, Detecting disruptive routers: A distributed network monitoring approach, in *Proceedings of the 1998 IEEE Symposium on Security and Privacy*, IEEE Press, New York, 1998, pp. 115–124.
52. *Consensus Roadmap for Defeating Distributed Denial of Service Attacks*, Version 1.10, SANS Institute, Bethesda, MD, Feb. 23, 2000.