IP Traceback: A New Denial-of-Service Deterrent?

The increasing frequency of malicious computer attacks on government agencies and Internet businesses has caused severe economic waste and unique social threats. IP traceback—the ability to trace IP packets to their origins—is a significant step toward identifying, and thus stopping, attackers.

HASSAN ALJIFRI
University of Miami

The Internet provides a wealth of information, convenience, and value to its users, but this accessibility makes it extremely vulnerable to motivated and well-equipped users intent on disrupting the flow of information or using it for personal gain. The tools for disruption are readily available to these Internet attackers, ranging from published operating-system weaknesses to executable software ready to exploit such vulnerabilities.

A common form of attack is denial of service (DoS). DoS attacks consume a remote host or network's resources, thereby denying or degrading service to legitimate users. Typically, adversaries conduct DoS attacks by flooding the target network and its computers with a large amount of traffic from one or (as in the case of distributed DoS, called DDoS) more computers under the attacker's control. Such attacks are among the toughest to address because they are simple to implement, hard to prevent, and difficult to trace. *IP traceback* methods provide the victim's network administrators with the ability to identify the address of the true source of the packets causing a DoS. IP traceback is vital for restoring normal network functionality as quickly as possible, preventing reoccurrences, and, ultimately, holding the attackers accountable.[1] Merely identifying the machines and networks that generate attack traffic might seem like a limited goal, but the essential clues it provides can help distinguish the actual attacker. Several efforts are under way to develop attacker-identification technologies on the Internet. This article looks at existing DDoS IP traceback methodologies and future trends.

The role of IP addresses

Ideally, the network traffic used in an attack should include information identifying its source. The Internet protocol (IP) specifies a header field in all packets that contains the source IP address, which would seem to allow for identifying every packet's origin. However, the lack of security features in TCP/IP specifications facilitates *IP spoofing*—the manipulation and falsification of the source address in the header. The Internet's current routing infrastructure is stateless and largely based on destination addresses, but no entity is responsible for ensuring that source addresses are correct. Thus, an attacker could generate offending IP packets that appear to have originated from almost anywhere. Although some network-based DoS attacks use IP spoofing by default,[2] only a small percentage of DDoS attacks use forged source addresses;[3] most attack their targets indirectly through other, previously compromised zombie systems.

To prevent this IP address manipulation, Kihong Park and Heejo Lee[4] proposed to install distributed packet filters on autonomous systems over the Internet to stop packets with spoofed IP addresses. Another solution is to set up network routers on ISP networks to ensure that the packets routed from the networks only contain valid source addresses. For both political and technical reasons, this process of blocking invalid packets at routers—called *ingress filtering*[5]—is not fully enforced today. Although there is a great deal of inter-ISP cooperation for tracing back and combating attacks, the process itself is not fully automated, and routine traffic measurements are not shared between ISPs. In addition, some ISPs refuse to install inbound filters to prevent source-address spoofing.[6] The ability to reliably trace network attacks to their sources might provide some deterrence to risk-averse individuals.[1] IP traceback methods are only the first step toward finding the true identity of the attacker who controls several compromised machines in the DDoS.

Current IP traceback approaches

Current IP traceback methods are either reactive or proactive.[6] Reactive measures initiate the traceback process in response to an attack. They must be completed while the attack is active; they're ineffective once it ceases. Input debugging[7] and controlled flooding[8] are examples of reactive measures, as described later. Most reactive measures require a large degree of ISP cooperation, which leads to extensive administrative burden and difficult legal and policy issues, thus, effective IP traceback methods should require minimal or no encroaching on ISP territory. Furthermore, reactive measures are not very effective against multipronged attacks and can't be used for post-attack analysis. Essentially, reactive measures are more effective for controlled networks than for the Internet.

In contrast, proactive measures record tracing information as packets are routed through the network. The victim can use the resulting traceback data for attack path reconstruction and subsequent attacker identification. Examples of proactive measures include logging,[6,9] messaging,[10,11] and packet marking,[12–19] as described later.

The key requirements for IP traceback methods include

- compatibility with existing network protocols,
- insignificant network traffic overhead,
- support for incremental implementation,
- compatibility with existing routers and network infrastructure,
- effectiveness against DDoS attacks, and
- minimal overhead in terms of time and resources.

ISP cooperation should not be required, and success should not depend on how long the attack lasts.

Current IP traceback methods fall into four major categories: link testing, logging, Internet control message protocol (ICMP)-based traceback, and packet marking.

Link testing

As the name implies, link-testing methods[7] (sometimes referred to as *hop-by-hop tracing*) work by testing network links between routers to determine the origin of the attacker's traffic. Most techniques start from the router closest to the victim and interactively test its incoming (upstream) links to determine which one carries the attack traffic. This process repeats recursively on the upstream routers until reaching the traffic's source. Link testing is a reactive method and requires the attack to remain active until the trace is completed (see Figure 1).

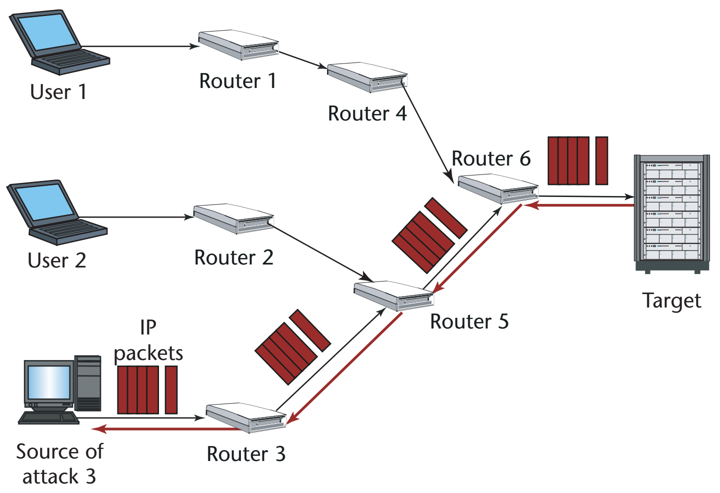

Figure 1. Link-testing traceback. The victim, or target, determines the attack signature and the process starts from the router closest to the victim. It interactively tests the upstream links to determine which one carries the attack traffic.

Input debugging[7] is one implementation of the link-testing approach. A feature already present on many routers, this feature lets the administrator determine incoming network links for specific packets. If the router operator knows the attack traffic's specific characteristics (called the *attack signature*), then it's possible to determine the incoming network link on the router. The ISP must then apply the same process to the upstream router connected to the network link and so on, until the traffic's source is identified—or until the trace leaves the current ISP's border. In the latter case, the administrator must contact the upstream ISP to continue the tracing process.

Frequently, the link-testing approach must be performed manually; recently, however, many ISPs have developed tools to automate this process and trace attacks across their own networks.[7] This technique's most severe drawback is the substantial management overhead in communicating and coordinating efforts across multiple network boundaries and ISPs. It requires time and personnel on both the victims' and ISPs' side, meaning there is no direct economic incentive for ISPs to provide such assistance. DDoS attacks compound this problem because attack traffic could

Table 1. Advantages and disadvantages of input debugging.

ADVANTAGES	DISADVANTAGES
Compatible with existing protocols	High overhead in terms of time and resources in organizations along the attack traffic path
Insignificant network traffic overhead	Communications and cooperation of ISPs along the attack path must be established
Supports incremental implementation	The attack must last long enough for a successful trace
Compatible with existing routers and network infrastructure	Less suitable for distributed denial-of-service attacks

Table 2. Advantages and disadvantages of controlled flooding.

ADVANTAGES	DISADVANTAGES
Compatible with existing protocols	Serves as a kind of denial-of-service attack
Support for incremental implementation	Requires accurate map of the network topology
Compatible with existing routers and network infrastructure	Attack must last long enough for a successful trace
	Less suitable for distributed denial-of-service attacks
	ISP cooperation might be required

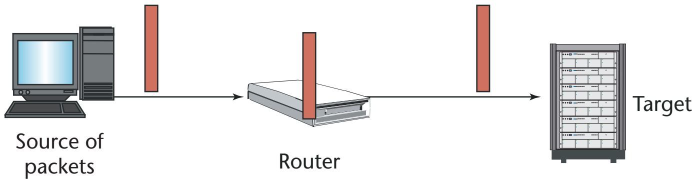

Figure 2. The router along the network path logs information about the packets that pass through it.

originate from machines under the jurisdiction of many separate ISPs. Table 1 illustrates input debugging's advantages and disadvantages.

Another technique that falls into the link-testing category is *controlled flooding*.[8] This technique works by generating a burst of network traffic from the victim's network to the upstream network segments and observing how this intentionally generated flood affects the attack traffic's intensity. Using a map of the known Internet topology around the victim, these packet floods are targeted specifically at certain hosts upstream from the victim's network; they iteratively flood each incoming network link on the routers closest to the victim's network. From changes in the attack traffic's frequency and intensity, the victim can deduce the incoming network link on the upstream router and repeat the same process on the router one level above.

The most significant problem with controlled flooding is that the technique itself is a sort of DoS attack, which can disrupt legitimate traffic on the unsuspecting upstream routers and networks. This, of course, makes it unsuitable for widespread routine usage on the Internet. Table 2 illustrates controlled flooding's advantages and disadvantages.

Logging

An obvious solution to establishing the true origin of offending Internet traffic is to log the packets at key routers throughout the Internet and then use data-mining techniques to extract information about the attack traffic's source (see Figure 2). Although this solution seems obvious and allows accurate analysis of attack traffic (even after the attack has stopped), its most significant drawbacks include the amount of processing and storage power needed to save the logs. Also, the need to save and share this information among ISPs poses logistical and legal problems as well as privacy concerns. Given today's link speeds, packet logs can grow quickly to unmanageable sizes, even over short timeframes. For example, an OC-192 link, which ISPs frequently use as their connection to the Internet's backbone, can transfer 10 Gbps (1.25 Gbytes per second) of data. Ten minutes of traffic on one of these links would require 750 Gbytes of high-speed storage.

Although logging a probabilistic sampling of the packet stream and compression can reduce resource demands somewhat, those demands are still quite significant. Alex Snoeren and colleagues[9] proposed a novel approach to logging and IP traceback called SPIE (Source Path Isolation Engine). Instead of storing the whole packet, they suggested storing only a hash digest of its relevant invariant portions in an efficient memory structure called a *Bloom filter*. To complete an IP traceback request, a network of data collection and analysis

Table 3. Advantages and disadvantages of logging.

ADVANTAGES	DISADVANTAGES
Compatible with existing protocols	Resource-intensive in terms of processing and storage requirements
Support for an incremental implementation	Sharing of the logging information among several ISPs leads to logistic and legal issues
Compatible with existing routers and network infrastructure	Less suitable for distributed denial-of-service attacks
Allows post-attack analysis	
Insignificant network traffic overhead	
Can trace a single packet[9]	

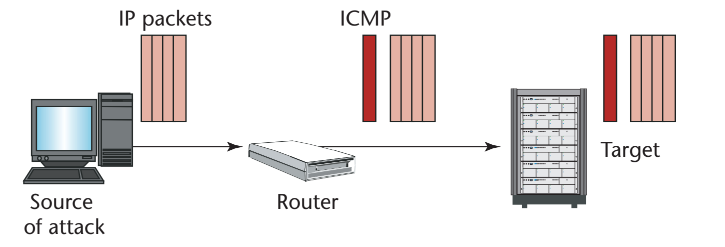

Figure 3. ICMP-based traceback. The router generates one ICMP packet for every 20,000 packets passing through it on the way to a specific target. The ICMP packet generated is then forwarded to the target.

agents spanning the different networks could use this method to extract significant packet data and generate appropriate attack graphs, thus identifying the attack traffic's origin.

Tatsuya Baba and Shigeyuki Matsuda[6] proposed an alternate and innovative logging approach. It entailed an overlay network built of sensors that could detect potential attack traffic, tracing agents (tracers) that could log the attack packets on request, and managing agents that could coordinate the sensors and tracers and communicate with each other. This approach attempts to overcome traditional logging methods' limitations and shortcomings by selectively logging traffic—after an attack is recognized and logging only certain characteristics, rather than entire packets. The approach also allows for increased speed and requires less storage.

ISPs and organizations can implement logging locally for internal purposes. Current logging-based traceback methods use a sliding time window for storing logged data to avoid excessive storage and analysis requirements in exchange for catching attacks while in progress or shortly thereafter (so that the required logging data is still available). Table 3 illustrates logging's advantages and disadvantages.

ICMP-based traceback

In July 2000, the Internet Engineering Task Force (IETF) formed a working group to develop ICMP traceback messages based on an approach called *iTrace* (www.ietf.org/html.charters/itrace-charter.html).[10] That approach used ICMP traceback router-generated messages, which the victim receives in addition to information from regular network traffic. These messages contain partial path information including: information that indicates where the packet came from, when it was sent, and its authentication.

Network managers could piece together these messages to trace a packet's path back to its origin. To limit the additional traffic this method generates, a router would generate an ICMP traceback message for only one in 20,000 packets passing through it (0.005 percent). This low probability limits additional network traffic, but still lets the victim figure out the attack traffic's actual path; in a typical DoS attack, the victim's network receives thousands of packets in a matter of seconds. Figure 3 illustrates ICMP-based traceback.

One of the iTrace scheme's weaknesses becomes apparent in a DDoS attack in which each zombie contributes only a small amount of the total attack traffic. In such cases, the probability of choosing an attack packet is much smaller than the sampling rate used. The victim probably will get many ICMP traceback messages from the closest routers but very few originating near the zombies' machines.

To overcome this drawback, researchers proposed an enhancement to iTrace called *intension-driven ICMP traceback*.[11] This technique separates the messaging function between the *decision module* and the *iTrace generation module*. A recipient network supplies specific information to the routing table to indicate it requests ICMP traceback message. On the basis of specific information provided in the routing table, the decision module would select which kind of packet to use next to generate an iTrace message. Based on this decision, the decision module will set one special bit in the packet-forwarding table. Setting that special bit indicates that the very next packet corresponding to that particular forwarding entry will be chosen to generate an iTrace message. The iTrace generation module then processes this chosen packet and sends a new iTrace message.

Table 4. Advantages and disadvantages of ICMP-based traceback.

ADVANTAGES	DISADVANTAGES
Compatible with existing protocols	Generates additional network traffic, even when using a very low frequency (1/20,000) for traceback messages
Supports incremental implementation	Unless there also is an encryption scheme with key distribution implemented, attackers could inject false ICMP traceback messages into the packet stream to mask the attack traffic's true origin
Allows post-attack analysis	ICMP traffic increasingly is filtered by organizations due to its use in several common attack scenarios
If implemented with encryption and key distribution schemes, presents a very promising and expandable technology for dealing with denial-of-service attacks	Very few ICMP traceback messages from distant routers in the case of a distributed denial-of-service attack (but can be somewhat alleviated by intention-driven) scheme
ISP cooperation is not required	
Compatible with existing routers and network infrastructure	

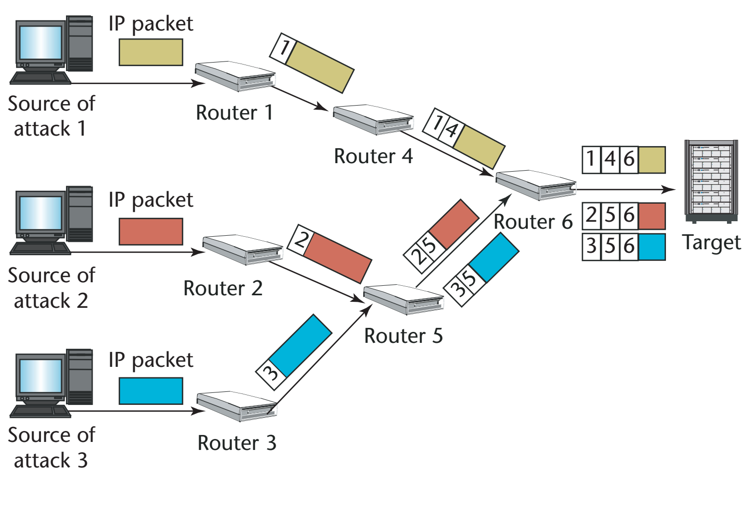

Figure 4. Packet marking. The router probabilistically marks packets as they travel through it (by inserting an indication of the router IP address). The marking process depends on the method adapted.

Intention-driven traceback also lets a recipient network signal whether it is interested in receiving iTrace packets, which increases the proportion of messages considered useful to the receiving network. This scenario also would be helpful if a given network suspects or detects that it is under attack: it could request iTrace packets from the upstream routers to identify the attack traffic's origin. Table 4 lists ICMP-based traceback's advantages and disadvantages.

Packet marking

Packet-marking methods[12–19] are characterized by inserting traceback data into the IP packet to be traced, thus marking the packet on its way through the various routers on the network to the destination host. This approach lets the host machine use markings in the individual packets to deduce the path the traffic has taken. To be effective, packet marking should not increase the packets' size (to avoid additional downstream fragmentation, thus increasing network traffic). Furthermore, packet-marking methodologies must be secure enough to prevent attackers from generating false markings. Problems also arise when we try to work within the framework of existing IP specifications. The order and length of fields in an IP header are specified, so for the packet-marking method to be effective, it must work with those settings and not alter them. Packet-marking algorithms and associated routers must be fast enough to allow real-time packet marking.

The simplest implementation of packet marking is to use the record route option (as specified in RFC 791) to store router addresses in the IP header's option field.[20] However, this method increases the packet's length at each router hop and can lead to additional fragmentation. Also, an attacker might try to fill the field reserved for the route with false data to evade traceback. Kihong Park and Heejo Lee[12] and Micah Adler[13] assessed the effectiveness and examined the trade-offs in packet-marking methods. Figure 4 shows the general concept of packet marking.

Stefan Savage and his colleagues[14] proposed algorithms for packet marking, ranging from simply appending the current router address to employing probabilistic traffic-sampling and compression methods. Traceback mechanisms that rely on probabilistic packet marking (PPM) have received widespread attention because they appear to be low cost. The method of choice that Savage described and implemented uses probabilistic sampling with a probability of 1/25 to avoid excessive overhead on the routers' packet marking. Furthermore, each packet

Table 5. Advantages and disadvantages of packet marking.

ADVANTAGES	DISADVANTAGES
Can be deployed incrementally and appears to be low cost	Requires modifications to the protocol
Works with existing routers and network infrastructure	Produces false-positive paths (that are not part of the attack paths)
Effective against distributed denial-of-service attacks	Victim must receive minimum number of packets
ISP cooperation not required	Cannot handle fragmentation
Allows post-attack analysis	Does not work with IPv6 and is not compatible with IPSec

stores information about only part of its route (an edge), rather than the full path.

Using this approach in conjunction with compression techniques and an additional field to prevent spoofing of routing information, Savage and his group used the IP header's 16-bit identification field to store the router's information. The identification field differentiates the fragments of IP packets and allows proper reassembly on the receiver side.

Compressed-edge fragment sampling (CEFS)[14] has become one of the most widely known schemes for IP traceback.[15] To perform a successful traceback, the victim must collect enough packets to reconstruct each edge of the attack path—and consequently, the full attack graph. This could prove very difficult in a DDoS attack, however, due to the difficulty in correctly grouping the fragments and encoded path edges together.

Dawn Song and Adrian Perrig[16] proposed modifications to Savages' edge-identification-based PPM method to further reduce storage requirements by storing a hash of each IP address instead of the address itself. The approach assumes the victim possesses a complete network map of all upstream routers. After edge-fragment reassembly, their method compares the resulting IP address hashes to the router IP address hashes derived from the network map (to facilitate attack path reconstruction). This modified method is supposedly more effective against DDoS attacks than previous methods have been. The authors also proposed an authentication-marking scheme that uses message authentication codes to prevent packet-content tampering by compromised routers along the attack path.

Drew Dean and his colleagues[17] proposed a modified PPM method that uses algebraic techniques from the fields of coding theory and machine learning to encode and decode path information as points on polynomials. They described schemes for full path, randomized path, and edge encoding. The encoded path information is stored in the Fragment ID field. At the victim side, algebraic methods are used to reconstruct the polynomials.

Michael Goodrich[18] proposed a PPM method called *randomize and link*. It sends a message Mx from each router in the attack path to the victim. The idea is first to break the message into a sequence of nonoverlapping word fragments w_i, and second, to compute the checksum (C) of the message. The checksum, which is called the cord of Mx, is used as both an associative address for Mx and a checksum to "link" all the pieces of Mx back together. The final step is to create a collection of blocks (b_i), where $b_i = [i, C, w_i]$. These blocks will be used to overwrite the usable (available) bits in the IP header. The reconstruction algorithm, based on b_i with the same C, tries all possible ways to arrange b_i in the right order.

On another research project,[19] my colleagues and I proposed an IP traceback approach called *Snitch*, which is a modified PPM technique that uses the space in the IP header made available by compression techniques described in RFC 2507.[21] Using these compression techniques make overcoming existing IP traceback methodologies' space limitations easier. Snitch uses an algorithm that identifies all the attack paths and reduces false-positive paths. Table 5 lists packet marking's advantages and disadvantages.

Practical solutions for IP traceback

Currently, no commercial off-the-shelf products can perform effective traceback across the Internet in real time or across multiple hops. This means that changes to existing routing protocols and hardware would be needed to implement any of the existing proposals.

Commercial packages promising a remedy for simple attacks are marketed to single corporations or ISPs and seem only to filter (or divert) the offending traffic, ensuring the corporate network's availability for legitimate traffic. Currently available commercial products can be implemented only locally in ISPs and organizations for internal purposes (see http://staff.washington.edu/dittrich/misc/ddos/lockheed.txt). Four commercial products promise to do traceback: Peakflow, FloodGuard, MANAnet, and ManHunt. (Note: inclusion of these companies in this article doesn't imply a belief they'll solve DDoS problems; these companies make that claim.)

Peakflow, marketed by Arbor Networks (www.arbornetworks.com), provides a distributed view of network-wide traffic and routing. It monitors ingress and egress traffic; graphs data by peer, router, interface, server, and port; and identifies and correlates anomalous network events with traffic disruption. It also facilitates "historical route tracking" of network traffic.

FloodGuard, from Reactive Network Solutions (www.reactivenetwork.com) is a system that detects attacks and conducts attack traceback. When multiple sources direct an attack on a protected domain, FloodGuard generates an alarm that it forwards to its upstream FloodGuard actuators. Each actuator analyzes its ingress traffic to either confirm or refute the alarm. If it confirms it (by seeing the traffic the alarm described), the actuator sends an alarm message back to the detector and then forwards the alarm to its upstream actuators (upstream relative to the protection domain of the detector that generated the alarm). These actuators repeat this process similarly. Of course, FloodGuard's traceback effectiveness depends on how widely deployed the actuators are.

Symantec's ManHunt (www.symantec.com) promises to "recognize and respond to DoS attacks in real time by automatically tracking the attack through the chain of ISPs so that the attack can be cut off at the source." For this to work, ManHunt agents must be deployed all the way back to the attacker's network. ManHunt sends and receives tracking information by communicating with the upstream routers, which must support ManHunt.

MANAnet (www.cs3-inc.com) is a system that promises to detect, defend, and traceback attacks. It works by having the routers mark the packets that pass through them in such a way that anyone looking at a packet can tell where it came from. Each router adds a few bits to a path field (currently stored in an IP option) that indicate which upstream router the packet came from. Generally, this is only meant to identify the LAN from which the packet came.

Many commercial routers have built-in features such as logging and IP accounting that can be used to characterize and track common attacks.[22] These features usually gather statistical data and offer no automated analysis or responses.

The Internet has transformed from an information repository to a vital channel for conducting business. Unfortunately, with this positive change has come an increased frequency in malicious attacks. All the proposed traceback schemes have their own specific advantages and disadvantages. Currently, no single solution could fulfill all the requirements outlined for an effective traceback method. For any of these IP traceback solutions to be effective, they would need to be deployed across corporate and administrative boundaries in a substantial portion of the Internet infrastructure. This in itself seems to be one of the biggest obstacles to a unified approach to IP traceback. Also, some measures are ineffective against DDoS attacks, are resource intensive, cause network overhead, and cannot be used for post-attack analysis. One conclusion we can draw from this is that unless IP traceback measures are deployed all over the Internet, they are only effective for controlled networks than for the Internet.

With all the ongoing research into IP traceback methods, the final challenge is convincing the disparate entities now controlling the Internet to work together and share information about traffic flowing through their networks. The technical aspects of IP traceback can be solved, as the research here proves. The question now is how to find a common denominator for industry (namely, ISPs and router manufacturers). □

References

1. S.C. Lee and C. Shields, "Tracing the Source of Network Attack: A Technical, Legal and Societal Problem," *Proc. 2001 IEEE Workshop on Information Assurance and Security*, IEEE Press, 2001, pp. 239–246.
2. *CERT Advisory CA-2000-01 Denial-of-Service Developments*, CERT, 2000; www.cert.org/advisories/CA-2000-01.html.
3. K.J. Houle and G.M. Weaver, *Trends in Denial of Service Attack Technology*, CERT Coordination Ctr., Carnegie Mellon Univ., 2001.
4. W. Lee and K. Park, "On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internets," *Proc. SIGCOMM*, ACM Press, 2001, pp. 15–26.
5. P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing," Internet Eng. Task Force RFC 2827, May 2000; www.ietf.org/rfc/rfc2827.txt.
6. T. Baba and S. Matsuda, "Tracing Network Attacks to Their Sources," *IEEE Internet Computing*, vol. 6, no. 3, 2002, pp. 20–26.
7. R. Stone, "CenterTrack: An IP Overlay Network for Tracking DoS Floods," *Proc. 9th Usenix Security Symp.*, Usenix Assoc., 2000, pp. 199–212.
8. H. Burch and B. Cheswick, "Tracing Anonymous Packets to Their Approximate Source," *Proc. 14th Conf. Systems Administration*, Usenix Assoc., 2000, pp. 313–322.
9. A.C. Snoeren et al., "Single-Packet IP Traceback," *IEEE/ACM Trans. Networking*, vol. 10, no. 6, 2002, pp. 721–734.
10. S. Bellovin, M. Leech, and T. Taylor, "ICMP Traceback Messages," Internet Draft, Internet Eng. Task Force, 2003; work in progress.
11. A. Mankin et al., "On Design and Evaluation of 'Intention-Driven' ICMP Traceback," *Proc. IEEE Int'l Conf. Computer Comm. and Networks*, IEEE CS Press, 2001. pp. 159–165.

12. W. Lee and K. Park, "On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack," *Proc. IEEE INFOCOM*, IEEE CS Press, 2001, pp. 338–347.
13. M. Adler, "Tradeoffs in Probabilistic Packet Marking for IP Traceback," *Proc. 34th ACM Symp. Theory of Computing*, ACM Press, 2002, pp. 407–418.
14. S. Savage et al., "Network Support for IP Traceback," *IEEE/ACM Trans. Networking*, vol. 9, no. 3, 2001, pp. 226–237.
15. M. Waldvogel, "GOSSIB vs. IP Traceback Rumors," *Proc. 18th Ann. Computer Security Applications Conf.* (ACSAC 2002), 2002, pp. 5–13.
16. D. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," *Proc. IEEE INFOCOM*, IEEE CS Press, 2001, pp. 878–886.
17. D. Dean, M. Franklin, and A. Stubblefield, "An Algebraic Approach to IP Traceback," *ACM Trans. Information and System Security*, vol. 5, no. 2, 2002, pp. 119–137.
18. M. Goodrich, "Efficient Packet Marking for Large-Scale IP Traceback," *Proc. 9th ACM Conf. Computer and Communication Security*, ACM Press, 2002, pp. 117–126.
19. H. Aljifri, M. Smets, and A. Pons, "IP Traceback Using Header Compression," *Computers & Security*, vol. 22, no. 2, 2003, pp. 136–151.
20. J. Postel, "Internet Protocol," Internet Eng. Task Force RFC 791, Sept. 1981; www.ietf.org/rfc/rfc0791.txt.
21. M. Degermark, B. Nordgren, and S. Pink, "IP Header Compression," Internet Eng. Task Force RFC 2507, Feb. 1999; www.ietf.org/rfc/rfc2507.txt.
22. *Characterizing and Tracing Packet Floods Using Cisco Router*, Cisco Systems, 1999; www.cisco.com/warp/public/707/22.html.

***Hassan Aljifri** is an assistant professor in the Computer Information Systems Department at the University of Miami. His research interests include computer and network security, e-commerce, and Internet computing. He received a PhD in computer engineering from the University of Miami. He is a member of the IEEE Computer Society. Contact him at the Univ. of Miami, Coral Gables, FL 33146; hassan@miami.edu.*