On IP Traceback

Andrey Belenky and Nirwan Ansari, New Jersey Institute of Technology

ABSTRACT

In this article we present the current state of the art in IP traceback. The rising threat of cyber attacks, especially DDoS, makes the IP traceback problem very relevant to today's Internet security. Each approach is evaluated in terms of its pros and cons. We also relate each approach to practical deployment issues on the existing Internet infrastructure. The functionality of each approach is discussed in detail and then evaluated. We conclude with a discussion on some legal implications of IP traceback.

INTRODUCTION

In recent years much interest and consideration have been paid to the topic of securing the Internet infrastructure as it continues to become a medium for a broad range of transactions. Secured data transmission, reception, and storage are of the utmost concern to facilitate various rapidly growing e-commerce applications. A number of approaches to security have been proposed, each attempting to mitigate a specific set of concerns. Since several high-profile distributed denial of service (DDoS) attacks on major U.S. Web sites in 2000, numerous approaches have been suggested to identify the attacker(s). A good introduction to the subject of DDoS attacks and defending against them can be found in [1].

The focus of this article are approaches dealing with the *IP traceback* problem, which is defined in [1] as identifying the actual source of any packet sent across the Internet. Most of the approaches discussed in this article were inspired by denial of service (DoS) and DDoS attacks. The purpose of both these attacks is to make a certain server inaccessible by exhausting its physical or logical resources. DoS attacks are composed of a single stream of attack traffic, DDoS attacks multiple streams of attack traffic. Readers are referred to [1] for a more complete discussion of (D)DoS attacks and defense against them. In general, IP traceback is not limited only to DoS and DDoS attacks. The task of identifying the actual source of the packets is complicated by the fact that the IP address can be forged or spoofed. In such instances, conventional methods of determining the location of the system with a given IP address on the Internet (e.g., **`traceroute`**) no longer work because the source address used for tracing can be spoofed. Therefore, more advanced methods of identifying the source of attacking packets are needed.

Identifying the source of the offending packets does not necessarily identify the actual attacker. The source of these packets may be a host in a stepping stone chain, a reflector, a zombie, or a device compromised by the attacker in some other way. IP traceback techniques neither prevent nor stop the attack; they are used *only* for identification of the source(s) of the offending packets during and after the attack. Furthermore, it may be impossible to precisely identify the source of the attack packets since it may be behind a firewall or have a private IP address. Consequently, IP traceback may be limited to identifying the point where the packets constituting the attack entered the Internet.

The rest of this article is structured in the following way. The next section describes current ways of dealing with anonymous attacks and discusses the motivation behind IP traceback. We then introduce the framework and metrics for evaluating the discussed schemes. The actual schemes and evaluations are presented, and implications and challenges associated with IP traceback technology are discussed. The final section provides a comprehensive comparison of all the methods described and touches on some legal implications of IP traceback.

AVAILABLE EXISTING TECHNOLOGIES AGAINST ANONYMOUS ATTACKS

This section provides some background on what methods are currently available for protection against anonymous attacks.

FILTERING AND ACCESS CONTROL

Filtering is the simplest mechanism to prevent anonymous attacks and has been available for many years. In essence, a router or firewall facing the Internet is configured to only accept traffic from certain addresses into the local network, and to only let traffic from certain addresses out. Filtering inbound traffic provides some limited protection from a network known to have been offensive in the past. Also, packets with a source address that belongs to the network itself or is from the private address space are also generally filtered. Unfortunately, this technique is unsuitable for Web site companies, which need to accept packets from as many sources as possible, and restricting those to a set of known trusted addresses is not acceptable.

Filtering outbound traffic ensures that packets with source addresses that do not belong to the preconfigured address range do not enter the Internet. If a network institutes outbound packet filtering, initiating an anonymous attack from this network becomes impossible. This technique can only be used close to the edge of the network where addressing rules are well defined. For transit networks where packets with a different source address can enter the network in multiple locations, source address filtering becomes complex and ineffective.

Both inbound and outbound filtering are configured manually and present considerable overhead for the routers in terms of processing each packet. These shortcomings are addressed in the recently proposed Source Address Validity Enforcement Protocol (SAVE) described in [2].

SYN Flood Protection

SYN flood protection is perhaps the most effective approach currently available in prevention of SYN flood attacks. SYN flood protection addresses the very cause of an attack, which is the TCP three-way handshake. Basically, the scheme keeps track of half-opened TCP connections. A connection to a host is considered half-opened when a TCP SYN request and a TCP SYN/ACK have been exchanged, but the third handshake message, TCP ACK, has not been received. The attacked host keeps allocating resources to the half-opened connections, which are never fully opened by the attacker. A large number of these half-opened connections may potentially exhaust the server of resources and even bring it down. SYN flood protection limits the number of these half-opened connections. When the number of half-opened TCP connections exceeds a certain threshold, either old half-opened connections will be closed in order of their age, or new ones will not be allowed to even reach the server.

This functionality is a commodity and is available on most routers, firewalls, and servers. It provides good protection against SYN flood attacks but is unable to provide protection against any other type of attack. In addition, this method does not provide any traceback information.

Tracing Backscatter with BlackHole Route Server

This method was introduced in [3] and is currently used by some Internet service providers (ISPs) to block DoS attack traffic to a given victim host as well as to determine where this traffic entered the ISP. When attack traffic is detected, the routing in the ISP network is manipulated in such a way that the attack packets are directed toward a so-called BlackHole route server, which is set up in advance. This redirection is enabled by certain features of the Border Gateway Protocol (BGP) configured on the border routers of a given autonomous system (AS). The BlackHole router server generates replies on behalf of the attacked destination. This diverts the attack packets from the victim. When replies are generated, they may reach only as far as the border router. The border router does not know where to forward the packet because the destination address, which is the source address in the attack packets, is spoofed, and in many cases does not exist; thus, Internet Control Message Protocol (ICMP) Destination Unreachable messages are generated. These messages can be intercepted. By examining the origin of these intercepted messages it is possible to determine the interface where the attack packets entered the network since they originate from the same interfaces through which the original attack packets entered the network.

This method is a first step in implementing a real IP traceback solution. However, it requires human intervention, and traceback is limited by administrative boundaries and only capable of tracing DoS attacks. Also, only DoS attacks with random nonexistent addresses may be traced.

The currently available methods of dealing with anonymous attacks are not comprehensive. They either deal with a very limited set of problems or are too expensive to implement and enforce. While it may be simply impossible to prevent attackers from attempting an attack, it might be possible to lessen, or even completely eliminate, the effects of the attack by not allowing the packets to reach the victim(s). This is the proactive approach discussed in detail in [1]. The reality, however, is that prevention of all attacks on the Internet is far from reality. When prevention fails, a mechanism to identify the source(s) of the attack is needed to at least ensure accountability for these attacks. This is the motivation for designing IP traceback schemes.

When the number of half-opened TCP connections exceeds a certain threshold, either old half-opened connections will be closed in order of their age, or new ones will not be allowed to even reach the server.

Framework and Evaluation Metrics

The main objective of this article is the evaluation of proposed IP traceback techniques within a framework defined later in this section. It is worth mentioning that most of the schemes presented here remain theoretical and have not been implemented in the industry except in trials and simulations. In addition, the following discussion of different traceback methods provides insight for evaluating IP traceback solutions that may be proposed in the future.

Following are metrics essential in comparing IP traceback approaches.

ISP Involvement — Tracking an anonymous attack is not a trivial task. An individual or organization would find this task difficult, if not impossible, without involving their upstream ISPs. Today, tracing an anonymous attack even within a single ISP remains a manual task. ISPs and enterprise networks do not have incentives to monitor for attack packets according to [1]. The lack of incentives comes from the fact that monitoring for such packets has no immediate benefit to the ISP itself or its subscribers. Furthermore, participating in traceback may mean disclosure of internal topology, investment in additional equipment, upgrades to existing equipment, and additional operational costs for the ISP. Consequently, IP traceback solutions should not assume complete cooperation of ISPs. A desirable quality of an IP traceback scheme should be low ISP involvement, which implies that the scheme should be easily built or inserted with little infrastructure or operational change. Most schemes assume that ISPs will provide limited facility to enable IP traceback, but the burden of the actual traceback process will be either shared between the subscriber and the ISP, or the sole responsibility of the victim. An ideal traceback scheme would have a very low level of ISP involvement.

An important evaluation criterion of an IP traceback scheme is the ability of the scheme to determine the source of an attack based on as few packets as possible once the attack has been identified.

Number of Attacking Packets Needed for Traceback — Attacks can consist of as few as one packet or many thousands of packets. An important evaluation criterion of an IP traceback scheme is the ability of the scheme to determine the source of an attack based on as few packets as possible once the attack has been identified. This will enable the scheme to successfully traceback more attacks. Ideally, the scheme should be able to trace the attacker with a single packet.

The Effect of Partial Deployment — Clearly, if any scheme is adopted in the Internet, not all ISPs will simultaneously implement this function. Any scheme can perform the traceback only within the ISPs that deploy it. It is expected that a given ISP would deploy the scheme on all of its routers; however, it is important that the scheme can produce meaningful results when deployed partially within a single ISP. This will allow for partial, gradual deployment on the Internet, making the scheme more practical. The effects of partial deployment can vary from inability to perform tracing altogether to producing meaningful traces limited to the range of deployment, which should be the case for the ideal scheme.

Processing Overhead — There are two considerations for processing overhead: *where* and *when* it is incurred. Additional processing associated with the traceback scheme can occur on the devices of the ISP network and/or at the subscribers, the potential victims of the attacks. For most methods, additional processing will occur in both places. Processing overhead on the ISP routers is especially undesirable since it may result in the need to upgrade or buy more equipment. Therefore, a scheme with less processing overhead incurred on the network will most likely be accepted by an ISP. For organizations, additional processing overhead is not as critical. Organizations are usually concerned about security and willing to invest in dedicated intrusion detection system (IDS) servers that would incur most of the processing associated with IP traceback. Another consideration is when the processing overhead is incurred. Processing overhead can be incurred for every packet and during traceback. Preferably it should be incurred only during traceback, which hopefully will be a relatively infrequent operation. An ideal scheme would incur minimal processing overhead during traceback only.

Bandwidth Overhead — Additional traffic that the network has to carry for traceback is considered bandwidth overhead. Large bandwidth overhead is undesirable since it may exhaust the capacity of links and routers, forcing the ISP to introduce additional capacity and possibly upgrade or purchase new devices. The scheme should not assume availability of infinite bandwidth. As with processing overhead, bandwidth overhead can be incurred either constantly, for every packet, or only during the process of traceback once an attack is identified. It would be preferable to incur bandwidth overhead only during the traceback process, if at all.

Memory Requirements — Additional memory may be required on routers, or dedicated traceback servers located at either the ISP network or the client site. Additional memory on routers is highly undesirable since it may result in upgrades. Additional memory on dedicated servers is tolerable. Therefore, the important metric of a traceback scheme is the amount of additional memory required on routers. An ideal scheme would have a limited amount of additional memory required at the dedicated server, and no additional memory requirements on network equipment.

Ease of Evasion — The scheme is said to be easy to evade if the attacker, who is aware of the scheme, can easily orchestrate an attack that will be untraceable. Clearly, this quality is not desirable in a traceback scheme, and the ease of evasion should be as low as possible for an ideal scheme.

Protection — Protection refers to the ability of the traceback scheme to produce meaningful traces if a limited number of network elements involved in a traceback have been subverted. A traceback scheme with good protection would be able to produce valid traces even if this happens. Taking over a router or a well protected server is an extremely difficult task, and can be accomplished most often due to errors in configuration or improper patching of software. It is assumed that the devices involved in traceback will be properly managed and protected, minimizing the chance of subversion. A high level of protection is preferred in a traceback scheme; however, it is assumed that the probability of an attacker actually taking over a device is very small. An ideal scheme should act as if a device is not part of the scheme when the device becomes subverted.

Scalability — Scalability relates to the amount of additional configuration on other devices needed to add a single device to the scheme. It also measures the ability of the scheme to perform as network size increases. Features that depend on configuration of other devices deteriorate scalability. If only newly added devices require configuration, scalability is good. If, on the other hand, introducing another device to the scheme requires configuration of other devices, scalability is poor. Also, scalability measures how easily the scheme can expand. An ideal scheme should be scalable, and configuration of the devices involved should be totally independent of each other.

Number of Functions Needed to Implement — This metric reflects how many different functions a vendor of equipment needs to implement for a given scheme. It is easier for a vendor to implement fewer functions. Ideally only a single function should need to be implemented. The amount of effort required to implement each function is not discussed in this article. Most of the functions described are straightforward to implement. It is worth mentioning that historically vendors implement features on equipment far ahead of their wide deployment.

Ability to Handle Major DDoS Attacks — This is an extremely important metric that reflects how well the scheme can perform the traceback of DDoS attack under severe circumstances (e.g., a large number of attackers using reflectors or random address spoofing). Many schemes are not able to cope with all types of attacks. Being able to trace any attack, especially a DDoS attack, is a necessary quality of a traceback scheme. An ideal scheme would be able to trace back *all* attacks.

Ability to Trace Transformed Packets — A packet transformation is a modification of the packet during the forwarding process. Most common transformations include Network Address Translation (NAT), where source and/or destination IP address are changed; and tunneling, where a given packet is encapsulated inside another packet. Another type of transformation is packet generation, the most common examples of which are Internet Control Message Protocol (ICMP) packets and duplications of a packet in multicast. It is essential for a traceback scheme to handle transformations; otherwise, attacks that use packet transformations cannot be traced. An ideal scheme would correctly trace back attacks consisting of packets that undergo any number of transformations of any type.

Evaluation of Schemes

This section provides an overview on current state-of-the-art approaches to IP traceback, and evaluate them against the metrics established above. The traceback schemes discussed below fall into four general categories:

End-host storage
- Probabilistic packet marking (PPM)
- ICMP traceback (iTrace)

Packet logging
- Hash-based IP traceback

Specialized routing
- Overlay network
- IP traceback with IPSec

State of the network inference
- Controlled flooding

Probabilistic Packet Marking

This scheme is based on the idea that routers mark packets that pass through them with their addresses or a part of their addresses. Packets for marking are selected at random with some fixed probability of being selected. As the victim gets the marked packets, it can reconstruct the full path, even though the IP address of the attacker is spoofed. Originally introduced in [4], this scheme was improved in several different ways, among which [5] introduced improved coding methods and security. This scheme is aimed primarily at DoS and DDoS attacks as it needs many attack packets to reconstruct the full path.

Figure 1 depicts a schematic illustration of the approach. Attacker A initiates an attack to victim V. Assume that the path the packets take is **R1-R2-R4-R9-R12**. (This path will also be adopted for illustrating other schemes in this article.) Each router implementing PPM accepts the stream of packets, and before routing them probabilistically marks them with its partial address information (i.e., puts the router's partial address in the packet headers). Packets are marked with a marking probability *p*, which is suggested to be 0.04 in [4]. When the victim receives enough such packets, it can reconstruct the addresses of all the PPM-enabled routers along the attack path. Clearly, in order to reconstruct the full path the flow must contain a large number of packets.

To deploy the scheme, the vendors need to implement two functions: marking and reconstruction. Once the marking function is available, the software on all routers would have to be upgraded. Upgrade of the software on the routers is a straightforward task.

To deploy the scheme, vendors need to implement two functions: marking and reconstruction. Once the marking function is available, the software on all routers must be upgraded. Upgrade of the software on the routers is straightforward. Once the routers are upgraded, PPM needs to be enabled, and that is the extent to which an ISP needs to get involved in the scheme; therefore, ISP involvement is low. Additional PPM-enabled routers can be added independently, which indicates good scalability. The number of packets required for path reconstruction is measured in thousands for the original proposal in [4], and decreases to just under 1000 packets for the improved scheme described in [5]. For partial deployment to be effective, the victim must be aware of the network topology and routing on the network. Processing overhead in network elements is incurred for every packet. For each packet the decision is made if it should be marked or not by generating a random number. Additionally, if the packet is marked, more processing overhead is incurred associated with composing the mark and updating the ID field and Reserved Flag in that packet. The overhead associated with packet marking is minimal, and should not require major upgrades to the router hardware. Major processing overhead will be incurred at the destination during reconstruction. Potentially, the victim could have to perform searches of data structures consisting of billions of entries. Reconstruction data structures will require a large amount of memory as well. However, as mentioned earlier, overhead and additional memory required at the potential victim is not a major setback. Bandwidth overhead for this scheme is zero since all traceback information is scrambled in the IP packet header and completely inband. Finally, considering the improved version of PPM described in [5], evasion of the scheme is difficult since marks are authenticated. If a router that marks the packets becomes subverted, it can be reconfigured to incorrectly mark the packets and still be authenticated by the victim. This may result in an incorrectly reconstructed path. Relying on the assumption

PPM can handle packet modification transformations of the packets directed to the victim. However, in the case of packet generation transformation by a reflector, the traceback will be limited only to that reflector. Fragmented traffic will be corrupted by the scheme, but the traceback will not be affected.

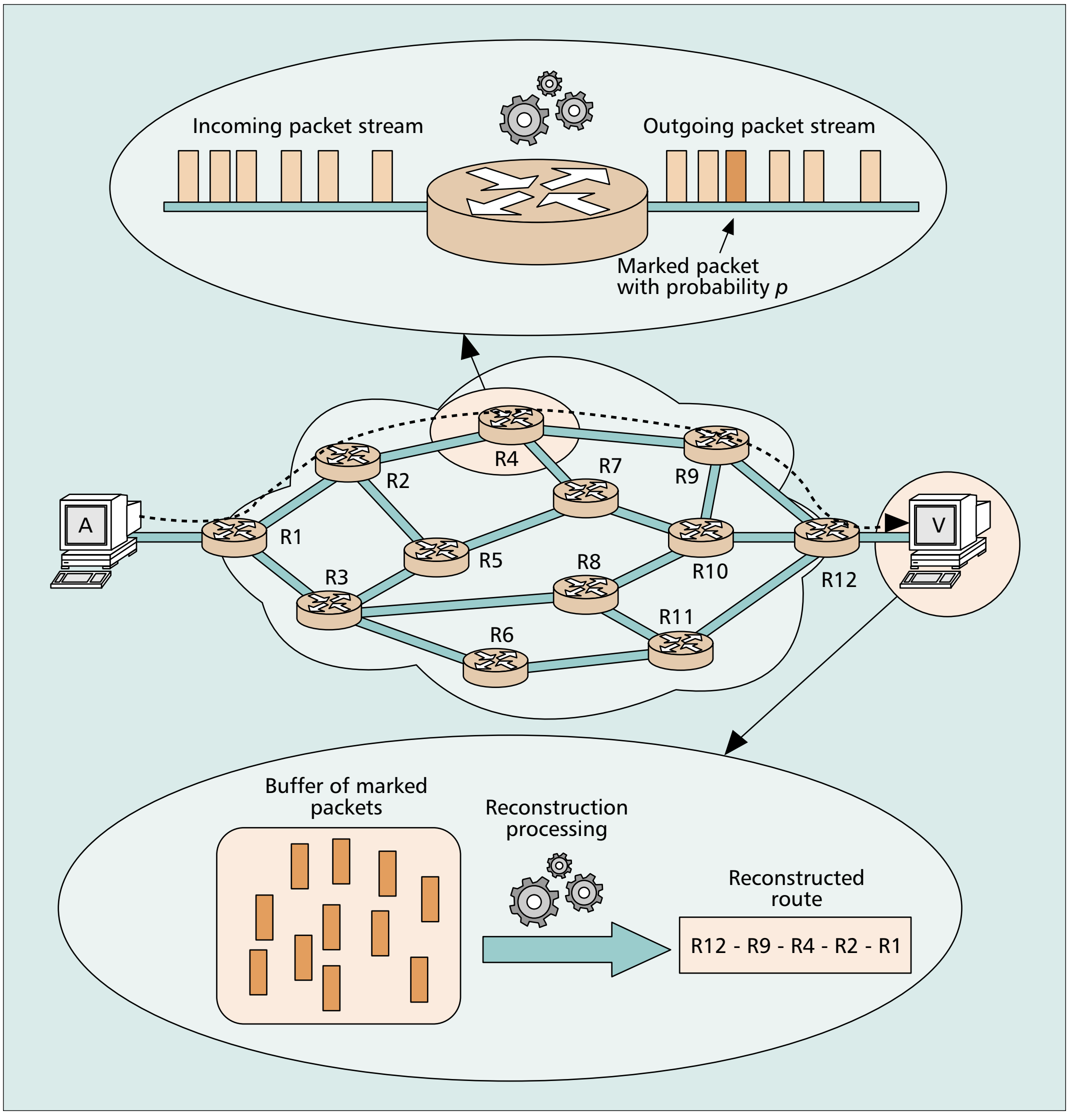

Figure 1. *Probabilistic packet marking.*

made in [5] that the victim is aware of the network topology, subversion of a small number of routers will not be a major problem. PPM can handle packet modification transformations of the packets directed to the victim. However, in the case of packet generation transformation by a reflector, traceback will be limited only to that reflector. Fragmented traffic will be corrupted by the scheme, but traceback will not be affected. The ID field normally used for fragmentation is used for the mark. If a single fragment of the original datagram is marked, the reassembly function would fail at the destination. Traceback would still be possible since the mark would be processed before reassembly. This problem is addressed by selecting a lower probability of marking for fragmented traffic, but this raises the number of packets needed for reconstruction. Also, tunneling may create a problem for reconstruction if marks are extracted before the outer header is removed. Carefully choosing the placement for the reconstruction function will remedy this potential problem. Both schemes described in [4, 5] are unable to perform traceback for a major DDoS attack with a large number of reflectors. Traceback with a PPM-like scheme is capable of tracing only a limited number of reflectors.

ICMP Traceback

ICMP traceback takes a different approach in determining the full path of the attack. This approach was originally introduced in [6].

Figure 2 illustrates the ICMP traceback scheme. Every router on the network is configured to pick a packet statistically (1 in every 20,000 packets recommended) and generate an ICMP traceback message or *iTrace* directed to the same destination as the selected packet. The iTrace message itself consists of the next and previous hop information, and a timestamp. As many bytes of the traced packet as possible are also copied in the payload of iTrace [6]. The time to live (TTL) field is set to 255, and is then used to identify the actual path of the attack. If routers in the path of the attack from A to V implement the scheme, the process illustrated in Fig. 2 occurs. The routers on the path generate a new packet with an iTrace message. This is unlike PPM, where the

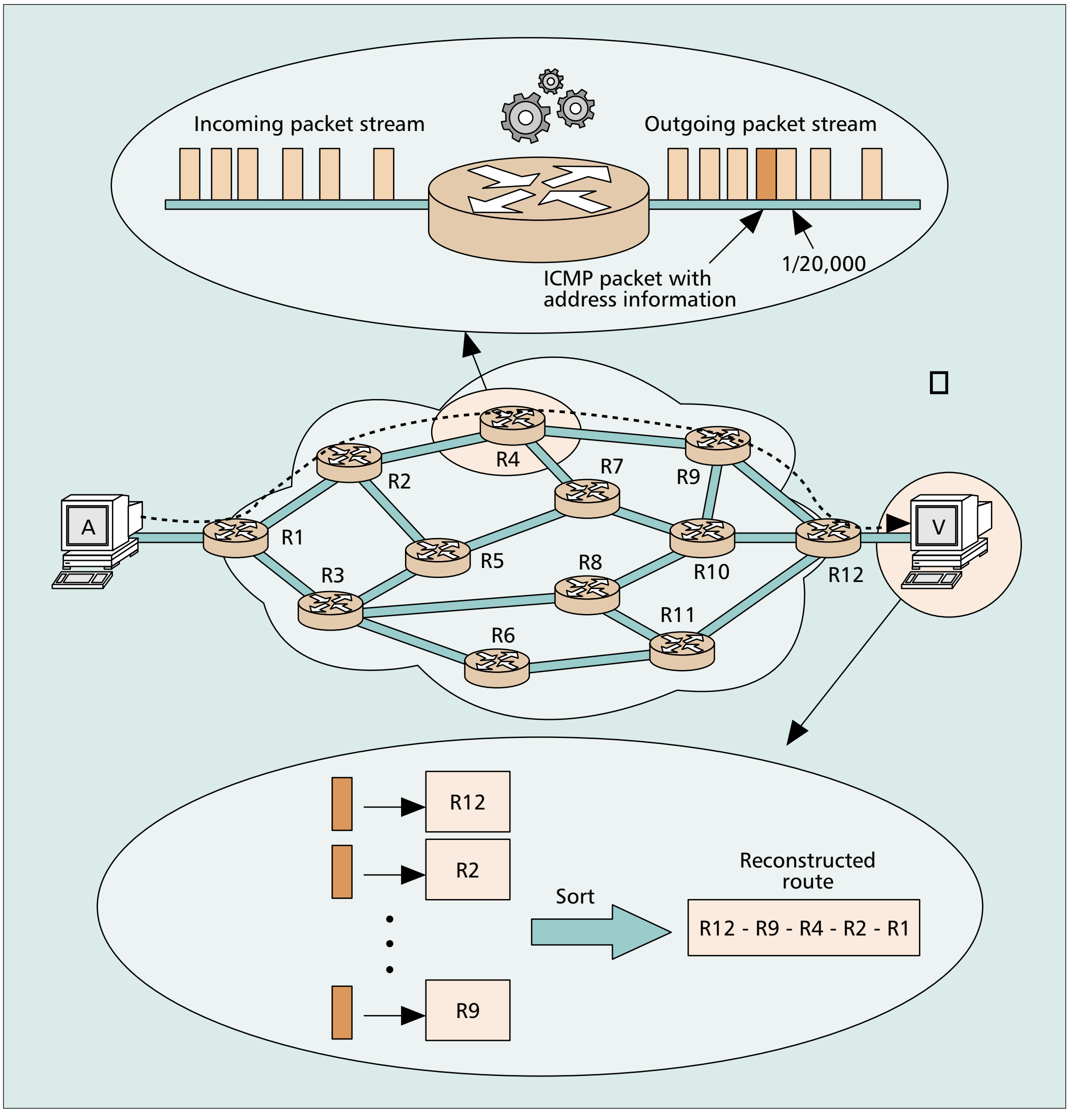

Figure 2. *ICMP traceback.*

By assuming the victim is under the (D)DoS attack, and therefore the volume of packets going to it is large, the victim will eventually get all the addresses of the routers on the attack path, which implement iTrace. By using TTL fields, these addresses can be sorted to reconstruct the attack path.

traceback information was completely in-band. By assuming the victim is under (D)DoS attack, and therefore the volume of packets going to it is large, the victim will eventually get all the addresses of the routers on the attack path that implement iTrace. By using TTL fields, these addresses can be sorted to reconstruct the attack path. It was shown in [7] that while this approach is efficient and reasonably protected, the chance of receiving a useful iTrace is small if the victim undergoes a major DDoS attack, especially if the attack was carefully orchestrated with the goal of reducing the probability of useful iTraces. The mechanism to resolve this statistical problem is to associate a weight or value with every iTrace generated. The value is affected by the distance from the victim, frequency of iTraces being sent to the victim, and time since the attack began. Having these three contributors to the value of iTrace, the original proposal [6] was augmented by an algorithm to make a more educated choice of packet for iTrace. While introducing definite benefits, these augmentations somewhat complicate the algorithm and require a change to the forwarding table on every router implementing this scheme.

The evaluation of this scheme is very similar to the evaluation of PPM. To deploy the scheme, vendors need to implement two functions: iTrace and reconstruction. It is worth mentioning that implementing value-based ICMP traceback will require a change to the structure of the routing tables on the routers. Once the iTrace function is available, the software on all routers needs to be upgraded. Upgrade of the software on the routers is straightforward. Once routers are upgraded, ICMP traceback must be enabled, and that is the extent of ISP involvement in this scheme. Additional ICMP traceback enabled routers can be added completely independently, which indicates good scalability. The number of attack packets required for path reconstruction is measured in thousands since the probability of generating an ICMP traceback message is 1/20,000. For partial deployment to be effective, the victim must be aware of the topology and routing on the network. Processing overhead in network elements is incurred for every packet. As in the case of PPM, for every packet the decision is made if it

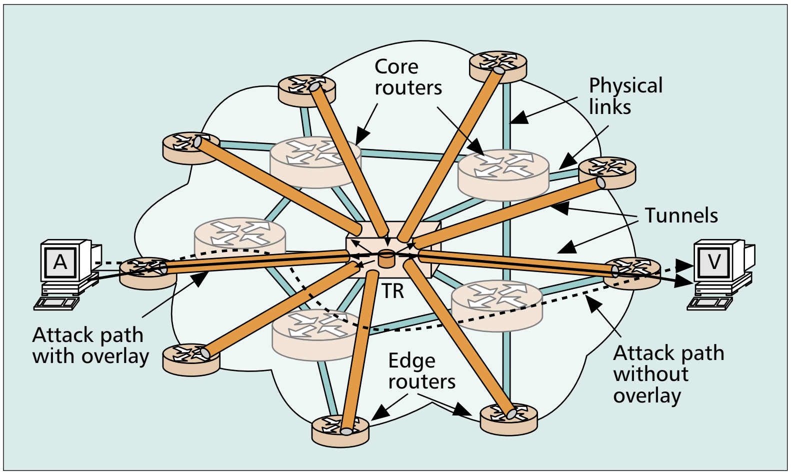

Figure 3. *Overlay network.*

should be marked or not by generating a random number. Additionally, if the packet is marked, more processing overhead is incurred associated with generating a new packet. The overhead associated with generating a new packet is minimal and should not require major upgrades to router hardware. A negligible amount of additional memory is necessary on all the routers if value-based iTraces are implemented. Major processing overhead is incurred at the destination during reconstruction. Potentially, the victim could have to perform searches of data structures consisting of thousands of entries. Reconstruction data structures will also require a large amount of memory. Bandwidth overhead for this scheme is minimal, and will be about 0.005 percent derived from the fact that about 1 in every 20,000 packets will be marked. If authentication mechanisms mentioned in [6] are implemented, evasion of this scheme would be difficult. However, the way the scheme is described, there is nothing to prevent an attacker from generating fake iTraces. DDoS attacks involve a massive amount of traffic from many different sources; plausible-looking fake chains could easily deceive a victim according to [6]. If a router that marks the packets becomes subverted, it can be reconfigured to generate incorrect iTraces, resulting in an incorrectly reconstructed path. The ability of handling major DDoS attacks with ICMP traceback was addressed in [7], where a few improvements, described earlier in this section, were suggested to enable the scheme to trace DDoS attacks. However, even with these improvements, ICMP traceback will not be able to perform the traceback for a DDoS attack with a large number of reflectors. Therefore, the ability to handle major DDoS attacks is poor. Ability to handle packet transformations is very similar to PPM. Transformation undergone by the stream of packets to the victim is not an issue, but transformation caused by a reflector will limit the traceback to the reflector only.

Overlay Network

This solution to the traceback problem is introduced in [8]. Logically, the solution introduces a tracking router (TR) in the network, as seen in Fig. 3. This TR monitors all traffic that passes through the network. In order to be able to monitor all of the traffic on the network, all packets have to be routed through this TR. This is accomplished by building a generic route encapsulation (GRE) tunnel from every edge router to this TR. Once the appropriate routing has been configured on the edge routers and TR, all traffic from an ingress edge router would travel over the GRE tunnel to the TR, and then from the TR over another GRE tunnel to the egress edge router. While core routers carry the traffic, logically it is only one hop from an edge router directly to the TR. Shaded network elements in Fig. 3 are simply transport for the overlay network. This architecture can be visualized as a star topology with the TR in the center and all of the edge routers on the network connecting to it with GRE tunnels. Since tunnels are built over the existing topology and utilize existing routing protocols, this star-like logical network is said to be an overlay network.

In reality, of course, a single TR will not be able to handle the load of packets from the whole network. Therefore, it is physically a fully connected mesh of several TRs, which can still be logically thought of as a single TR. The TR will utilize signature-based intrusion detection. This is different from all the other schemes, where intrusion detection was a function of the victim. When an attack is detected, meaning a single packet or sequence of packets that constitutes an intrusive action, the origin of the attack can be identified because it is only one hop away. In order to deploy this scheme, no additional functionality needs to be developed by vendors. The scheme takes advantage of the features available on most routers today. On the other hand, ISP involvement in this scheme is large. The ISP has to perform a traceback as well as identify the attack completely on its own. Also, a number of TRs and IDS servers would have to be purchased by the ISP. ISP involvement is therefore high. Adding another edge router to the network results in having to configure the TR in addition to a newly added device to enable traceback on them. The scheme has a severe limitation: it will only function well within a single administrative domain. In order for the overlay network to function well across ISPs, it would be necessary to somehow connect all of the TRs into a single system, which was not proposed in [8]. This presents a big scalability issue and constitutes a major limitation. A single packet is necessary to trace back any attack, given, of course, that the attack is identified. As soon as IDS on the TR identifies the attack, it can be traced to the endpoint of the tunnel. Termination of the tunnel will be associated with an interface that faces the network, not the customer. If the edge router has multiple interfaces facing customers, it is impossible to determine from which of these interfaces the attack was initiated. This scheme has a trade-off between overhead and protection. In the originally proposed configuration with GRE tunnels, the bandwidth is about 20 bytes for each packet. For an attack composed of really short packets, this can be a significant bandwidth overhead. The protection of the scheme is rather low since the tunnel packets can be forged by the attacker

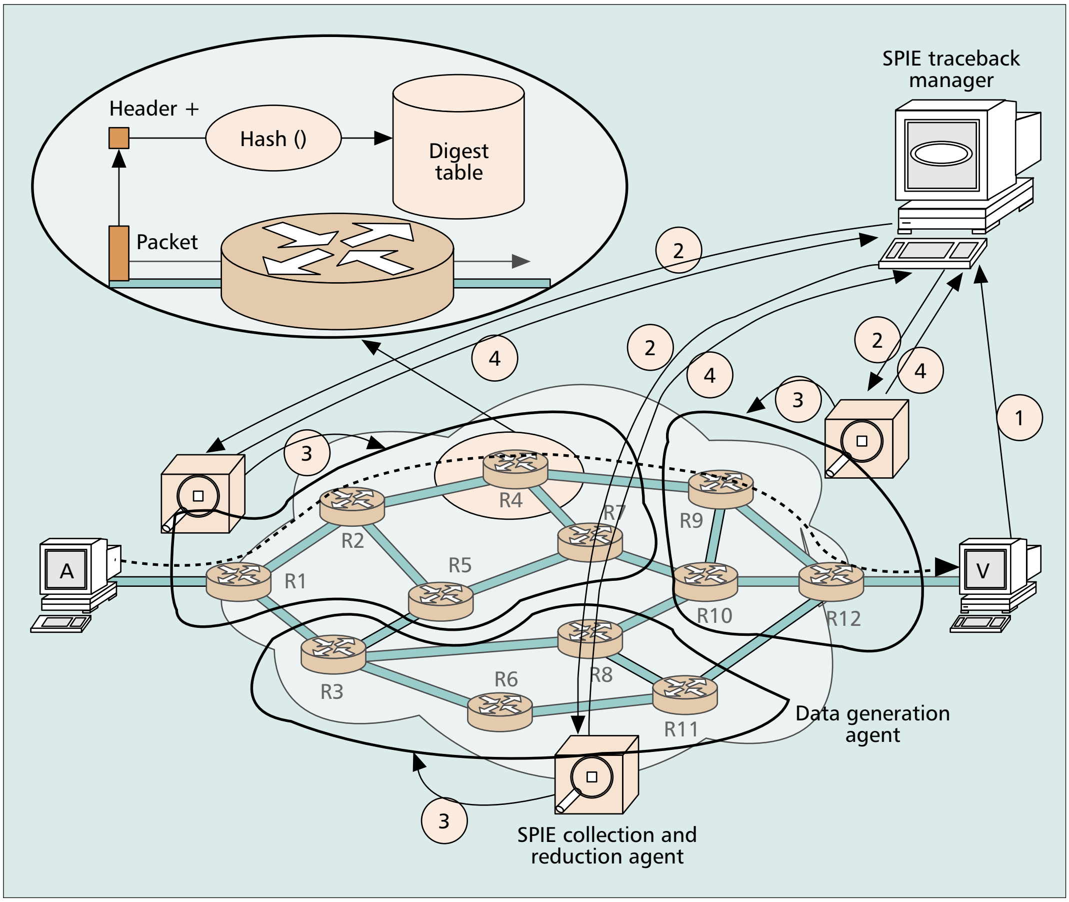

Figure 4. *Hash-based traceback.*

The scheme is able to handle major DDoS attacks in a sense that the source of any packet can be traced to the edge of the network. Handling packet transformations is not an issue for this scheme.

when a router is subverted. However, if the tunnels are built with IPSec, the bandwidth overhead is even larger, but the level of protection of the scheme becomes very high. Moreover, some processing overhead on both ends of the tunnel is incurred for every packet. The scheme is able to handle major DDoS attacks in a sense that the source of any packet can be traced to the edge of the network. Handling packet transformations is not an issue for this scheme.

HASH-BASED IP TRACEBACK

This approach is introduced in [9]. The scheme is officially called Source Path Isolation Engine (SPIE). In hash-based traceback, every router captures partial packet information of every packet that passes through the router, to be able in the future to determine if that packet passed through it. In this scheme such routers are called *data generation agents* (DGAs). DGA functionality is implemented on the routers. The network is logically divided into regions. In every region SPIE collection and reduction agents (SCARs) connect to all DGAs, and are able to query them for necessary information. The SPIE traceback manager (STM) is a central management unit that communicates to IDSs of the victims and SCARs, as seen in Fig. 4.

As packets traverse the network, digests of the packets get stored in the DGAs. In this scheme, constant fields from the IP header and the first 8 bytes of the payload of each packet are hashed by several hash functions to produce several *digests*. Digests are stored in a space-efficient data structure called a *bloom filter*, which reduces storage requirements by several orders of magnitude. When a given bloom filter is about 70 percent full, it is archived for later querying, and another one is used. The duration of using a single bloom filter is called a *time period*. Hash functions also change for different time periods. Also, a DGA is able to record any transformation (NAT, IPSec, etc.) that may affect those fields. The type of transformation and the data necessary to reconstruct it are stored in the transform lookup table (TLT). Each bloom filter for a given time period has its own TLT associated with it. When the STM receives notification of an attack from a victim's IDS (step 1), it sends the appropriate requests to SCARs (step 2). SCARs in turn obtain copies of the digests and transformation tables from DGAs for the appropriate time period (step 3). After analyzing and correlating the tables, SCARs are able to figure out which routers in the region, if any, forwarded the packet. The SCAR can then reconstruct the path along which the packet traversed through the region, and reports it to the STM (step 4). Based on this information, the STM is able to reconstruct the path through the network.

This scheme involves three functions that must be implemented: STM, SCAR, and DGA. ISP involvement in this scheme is high. The routers have to be upgraded to support the function of DGAs, and the ISP has to purchase

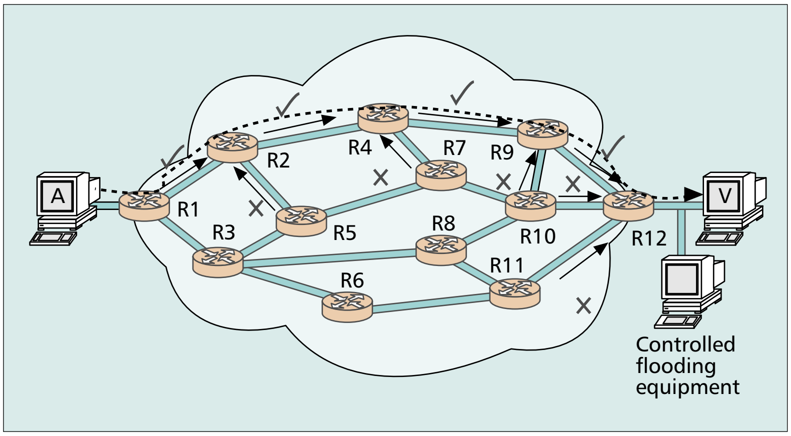

Figure 5. *Controlled flooding.*

equipment for SCARs and at least one STM. The scheme can perform a traceback with a single attack packet. Effects of partial deployment are similar to the case of PPM. If DGA functionality is implemented only on some routers in the ISP network, it is possible to reconstruct the path by checking for the digest at those nodes that implement DGA functionality and extrapolating paths between DGAs that report a hit, provided the topology of the network and routing are known to the STM. Inter-ISP tracing is possible provided there is a necessary degree of cooperation and trust. This issue is briefly mentioned in [9]. Processing overhead is incurred for every packet on every router to store its digests in the bloom filter. During traceback, all three functional components incur additional processing. There is no additional processing incurred at the client site. There is no bandwidth overhead associated with every packet; however, there is some minimal bandwidth overhead incurred during traceback. Additional memory required at DGAs is minimal, 0.5 percent of the link bandwidth per unit time, and can be incorporated in the router. A more substantial amount of memory is required by SCARs and the STM, but these devices are dedicated to the traceback function, and a large amount of memory required for those functions is not a concern. The scheme is extremely difficult to evade. While the scheme is equipped to handle practically any packet transformation, a combination of several packet transformations done in a particular order coupled with loss of particular packets may potentially make some transformations irreversible. These conditions are not a major concern because they are unlikely to occur. A subverted DGA can be potentially reconfigured to report that it has seen packets that never passed through it, and vice versa. This will produce paths with one hop in error. Subverting a DGA will not, however, allow the attacker to learn of any packet content since nothing can be learned by examining the content of the bloom filter. If a SCAR becomes subverted, the whole segment of the path can be incorrect. Finally, if STM becomes subverted, the traceback will not produce a correct traceback at all. Reiterating the fact that taking control of a device is extremely difficult, these considerations should not be the major factor. In order to add another DGA, a SCAR needs to be reconfigured, and in order to add another domain to the hierarchy, the STM needs to be reconfigured. Due to the fact that configuration of other devices is involved when adding another device, the scheme does not scale very well. The scheme is able to handle massive DDoS attacks. The limitation of the scheme is the timing issue. For high-rate interfaces, traceback must be performed within a very short period of time. The problem is magnified for the interdomain case when time synchronization cannot be expected. Also, such a strict timing constraint on traceback prohibits post-mortem traceback (i.e., long after the attack has finished). This becomes important when the victim does not realize it is being attacked, or cannot contact the STM during the attack for some reason.

Controlled Flooding

Controlled flooding is introduced in [10]. It is only valid for DoS attacks. It relies on the fact that during DoS attacks the links of the attack path should be heavily loaded. This assumption may not hold for modern backbone networks with abundant bandwidth available on the links. By carefully measuring incoming traffic to the attacked system and loading the links of the suspected path even more, a drop in the rates of the attack packets should be observed. The process can be repeated for the next hop and so on until the source of the attack is identified.

This concept is illustrated in Fig. 5. Once the DoS attack based on flooding is identified by V, equipment that measures the load on the link and equipment will be used to generate traffic on the network. Once this is accomplished, the traceback begins. Routers that connect to **R12**, the router closest to the victim, are determined. Then a short burst of traffic is generated from **R11**, hoping that the rate of attack packets to the victim will drop. In this particular case, it did not. Thus, this link and all the paths that may utilize it are excluded from the set of possible paths. Second, the link from **R10** to **R12** is loaded. Again, no drop in the rate of packets to the victim is observed; therefore, the link is also excluded. When the link between **R9** and **R12** is loaded, the desired drop in the rate is observed. It is thus concluded that the link between **R9** and **R12** belongs to the attack path. The process is recursively repeated until the source of the attack or the nearest router to it is identified.

The links are suggested to be loaded using the **chargen** service on the routers. The originator of the **chargen** service opens a connection to a device on TCP or UDP port 19. In response, this device generates a large amount of data back to the originator. This outcome is not desirable since the task here is to only load a single link. In order to avoid this, the source address of the equipment is spoofed as the next hop address to this router. In order to load the link between **R2** and **R4**, controlled flooding equipment spoofs its source address as the interface of **R4** connecting to **R2**, and starts the **chargen** service on **R2**.

The packets generated would be directed to **R4**, thus loading the link between these two routers.

There are several limitations to this approach. First of all, contrary to the claim in [10], most routers have **`chargen`** disabled. In fact, it comes disabled by default now on most equipment. Second, the approach assumes access to routers on the ISP network. This is also a big assumption. Even if the routers on the ISP network are publicly addressable, it is very unlikely that the customer will be allowed to access them in any way. Such readily available access would be constantly exploited by hackers. This method of denying service is easier. In addition, the authors suggested basically initiating DoS attacks on the network, although brief ones, in order to determine the source of a similar attack.

On the positive side, however, this is the only method introduced so far that does not rely on any ISP cooperation. This is an important and desirable quality of an ideal traceback scheme. Only a single function must be created to perform control flooding. The number of packets required for the scheme to successfully complete a traceback is large. Processing overhead is incurred only during the traceback and only at the equipment of the victim. The bandwidth overhead is extremely high. Additional memory requirements are very limited and required only at the victim's site. Partial deployment is not applicable here since equipment needs to be deployed only when and where an attack occurs. Ease of evasion and protection of this method are not an issue since there is no threat of compromising the traceback data. The scheme is not sensitive to packet transformations. Only DoS attacks can be traced with this scheme since traceback is limited to one attacking stream. Therefore, this scheme is not able to trace large-scale DDoS attacks. ISPs, as mentioned above, are not involved in this scheme. While controlled flooding can be automated, the authors made a point in [10] that it is preferable to stay manual because of potentially severe consequences of a programming error. For this reason alone, this approach is not feasible for wide deployment in addition to the limitations mentioned above. Needless to say, this scheme cannot trace the attack when it is over. It is possible that certain customers may engage in controlled flooding; however, it is absolutely infeasible for ISPs to encourage or support such efforts.

IP Traceback with IPSec

This approach is introduced in [11] as part of a network-based intrusion detection framework called *DECIDUOUS*. While the framework itself is beyond the scope of this article, the mechanism of identifying the source address of an attack is of interest.

The mechanism is based on an assumption that complete network topology is known to the system. What follows is the underlying principle: If there is an IPSec security association between an arbitrary router and the victim, and the attack packets detected are authenticated by the association, the attack is originated on some device further than this router; if the packets of the attack are not authenticated by this security association, the attack is originated on some device between this router and the victim. By establishing these security associations, it is possible to identify a single router or group of routers from which the attack was initiated.

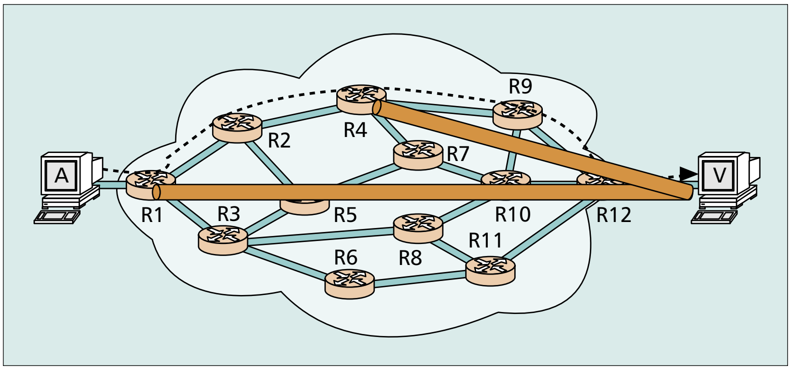

Figure 6. *Using IPSec for traceback.*

In Fig. 6, when the attack is detected, an IPSec security association is built between **R4** and V. If A was in fact an attacker, attack packets have to be authenticated since they will go through the tunnel. Next, the tunnel from **R1** to V is built. Note that from **R4** to V there will be two tunnels encapsulating traffic from A. In reality (this is not obvious from the figure) the second tunnel will be encapsulated in the first tunnel. Since the traffic is authenticated by two security associations, it is clear that the attack originated from somewhere behind **R1**. If, for example, the attack packets were only authenticated by the first tunnel and not the second, it would mean that the attack comes from somewhere between **R1** and **R4**; in the case of Fig. 6, it is **R2**.

How the system determines with which routers the victim should build IPSec associations if the source address is not known is a valid question. The answer is not simple. In short, the system goes through many iterations considering every possible path. Interested readers can familiarize themselves with the intricacies of these algorithms in [11].

No new functionality needs to be developed by vendors to enable this scheme, since it uses IPSec, which is available on most routers. An ISP must get involved in that it must disclose its topology to all of its clients so that they can build IPSec tunnels to all the routers. All the routers on the network must be configured to be able to build IPSec tunnels with all clients. The scalability of the scheme is therefore low. If "shared secret" authentication is used, all end systems need to be notified of any change on any router in the network, resulting in unacceptable scalability. We assume that digital certificates will be used for authentication of the security association. With the latter method of authentication, scalability is improved, but still not acceptable for wide inter-ISP deployment. The number of packets necessary for traceback is low. The victim has to build several IPSec tunnels and receive at least one packet after a given tunnel is built in order to trace back the attack. While not discussed explicitly, there is

		PPM	iTrace	Overlay	Hash-based IP traceback	Controlled flooding	Traceback with IPSec
ISP involvement		Low	Low	Large	Fair	None	High
Scalability		High	High	Poor	Fair	N/A	Poor
Vendor involvement (# of functions to implement)		2	2	None	3	1	None
Number of attack packets required for traceback		Thousands	Thousands	1	1	Huge	Fair
Is partial deployment within a single ISP possible?		Yes	Yes	No	Yes	N/A	Yes
Is prior knowledge of topology and routing required for traceback?		Yes, only if deployed partially	Yes, only if deployed partially	No	Yes, only if deployed partially	Yes	Yes
Is inter-ISP deployment possible?		Yes	Yes	No	Yes	Yes	Yes
Network processing overhead	Every packet	Low	Low	Low	Low	None	None
	During traceback	None	None	Low	Low	None	High
Victim processing overhead	Every packet	None	None	None	None	None	None
	During traceback	High	High	None	None	Fair	High
Bandwidth overhead	Every packet	None	Low	High	None	None	None
	During traceback	None	None	None	Low	Huge	High
Memory requirements	Network	None	Low	Low	Fair	None	None
	Victim	High	High	None	None	Low	None
Ease of evasion		Low	High	Low	Low	N/A	Low
Protection		High	High	Fair	Fair	N/A	High
Ability to handle packet transformations		Good	Good	Good	Good	Good	Good
Ability to handle major DDoS attacks		Poor	Poor	Good	Good	Unable	Poor
Limitations		DoS and DDoS attacks only.	DoS and DDoS attacks only.	Single ISP. Single point of failure.	Strict timing constrains on traceback process. Single point of failure.	DoS only. Manual. Unsafe. Inconsistent. Traceback is possible only while attack is in progress.	Single ISP.

Table 1. *Comparison of traceback schemes.*

no reason the scheme cannot be deployed partially, only on some routers. Deploying it on some routers would require, however, knowledge of those routers by all potential victims in advance. The processing overhead is high due to the processing associated with setting up tunnels with digital certificates, in real time, both at the victim's site and on the routers. This overhead will only be incurred during traceback. Bandwidth overhead is potentially high, as it is for all schemes that involve IPSec. This scheme is very difficult to evade because IPSec is very secure. Protection of this scheme is high since the worst thing that can happen if a router becomes subverted is that the IPSec tunnel between this router and the client would be impossible to build. This is equivalent to the situation when this router is not involved in the scheme to begin with. This scheme is not sensitive to most practical packet transformations. Also, the scheme is capable of tracing major DDoS attacks by tracing paths one by one; however, there is an issue with DDoS attacks: the routers can become the target of the attack themselves. Recall that the routers in the ISP have to be open for clients to set up IPSec tunnels. This can easily be exploited by attackers. By attempting to create IPSec tunnels to the router, an attacker can exhaust resources on the router. The tunnels will never be created because authentication will fail, but resources will be allocated before authentication failure occurs. One of two things can happen. The router will be so busy with opening new tunnels and authentication that the forwarding of the packets will be degraded, which will constitute a denial of service; or, if there is a set limit on the number of IPSec tunnels the router can handle, this limit can be reached and traceback will be impossible. For this reason, this scheme is deemed unable to handle complex DDoS attacks.

IP Traceback Implications and Challenges

In addition to the technical aspects of IP traceback, there are also legal and societal aspects [12]. Several U.S. federal laws relevant to traceback were not written with computer networking in mind. Currently, an insufficient number of court cases and precedents make it difficult to understand all implications of traceback.

Collecting packet headers is generally not considered invasion of privacy and is legal. Collecting payloads, on the other hand, is illegal. Collecting digests is currently a gray area. Schemes like PPM, overlay network, traceback with IPSec, and controlled flooding do not collect any data from the payload, and the results of traceback may be admissible in court. On the other hand, ICMP traceback and hash-based IP traceback collect either digests or actual content of packets, and may be inadmissible. The developers of IP traceback schemes must be aware of legal implications, and that these methods can potentially intrude on the privacy of individuals and corporations. According to [12], privacy of information is a higher priority than attack traceback even for organizations that may become targets of attack, and the incentive for implementing traceback schemes is minimal.

Implication policies are also very important. Legislation that requires IP traceback may be needed for ISPs to start implementing and deploying the schemes. This is a big problem in itself. Resolving this problem may not be enough since other countries do not have to comply with U.S. laws. With noncompliance, any traceback solution would be able to conduct traceback as far as the edge of the compliant network. Noncompliant countries may become a safe haven for attackers. The attacks may initiate from or go through them and be untraceable.

The issues discussed here are only the tip of the iceberg of many legal, political, and societal issues traceback may involve. Developers of the schemes should keep in mind that even the best traceback solutions from a technical standpoint may be unsuitable for implementation and deployment for nontechnical reasons.

Conclusions

In this article we present the state of the art in IP traceback, along with proposed solutions to this problem. Table 1 provides a summary of the evaluation and offers a comparison of IP traceback techniques.

As seen from Table 1, none of the methods possesses all the qualities of an ideal scheme. Solutions to a problem are rarely ideal. Very often several solutions produce a useful taxonomy. For the problem of IP traceback, several solutions have been proposed. Each has its own advantages and disadvantages. So far, none of the methods described in this article has been used on the Internet. When economic or political incentives become strong enough to justify deployment of IP traceback, some new requirements and metrics for evaluation might emerge.

Acknowledgments

This work has been supported in part by the New Jersey Commission on Higher Education via the NJI-TOWER project, and the New Jersey Commission on Science and Technology via the NJ Wireless Telecommunications Center. The authors would like to acknowledge Joseph Frank for proofreading the final manuscript, and the 11 anonymous referees and experts for their constructive comments that helped improve the quality of the article.

References

[1] R. K. C. Chang, "Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial," *IEEE Commun. Mag.*, Oct. 2002, pp. 42–51.

[2] J. Li *et al.*, "SAVE: Source Address Validity Enforcement Protocol," *Proc. INFOCOM*, 2002, vol. 3, pp. 1557–66.

[3] C. Morrow and B. Gemberling, "How to Track a DoS Attack (nanog post)," http://www.secsup.org/Tracking/.

[4] S. Savage *et al.*, "Network Support for IP Traceback," *IEEE/ACM Trans. Net.*, vol. 9, no. 3, June 2001, pp. 226–37.

[5] D. X. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," *Proc. INFOCOM*, 2001, vol. 2, pp. 878–86.

[6] S. M. Bellovin, "ICMP Traceback Messages," IETF draft, 2000; http://www.research.att.com/smb/papers/draft-bellovin-itrace-00.txt.

[7] S. F. Wu *et al.*, "On Design and Evaluation of 'Intention-Driven' ICMP Traceback," *Proc. 10th Int'l. Conf. Comp. Commun. and Nets.*, 2001, pp. 159–65.

[8] R. Stone, "Centertrack: An IP Overlay Network for Tracking DoS Floods," *Proc. 9th USENIX Sec. Symp.*, 2000, pp. 199–212.

[9] A. C. Snoeren *et al.*, "Single-Packet IP Traceback," *IEEE/ACM Trans. Net.*, vol. 10, no. 6, Dec. 2002, pp. 721–34.

[10] H. Burch and B. Cheswick, "Tracing Anonymous Packets to Their Approximate Source," *Proc. USENIX LISA*, 2000, pp. 319–27.

[11] H.Y. Chang *et al.*, "Deciduous: Decentralized Source Identification for Network-Based Intrusions," *Proc. 6th IFIP/IEEE Int'l. Symp. Integrated Net. Mgmt.*, 1999.

[12] S. C. Lee and C. Shields, "Challenges to Automated Attack Traceback," *IT Professional*, vol. 4, no. 3, May-June 2002, pp. 12–18.

Biographies

Andrey Belenky (avb0168@njit.edu) received his M.S. in telecommunication networks and B.S. in computer engineering (summa cum laude) in 1998 from Polytechnic University. From 1998 to 2002 he was with Telcordia Technologies Inc. working on various projects dealing with the design and deployment of IP networks. He was primarily involved in IP network routing and network security. He is pursuing his doctorate in computer engineering at the New Jersey Institute of Technology (NJIT), focusing on network security, in particular, IP traceback.

Nirwan Ansari (Nirwan.Ansari@njit.edu) received his B.S.E.E. (summa cum laude), M.S.E.E., and Ph.D. from NJIT, University of Michigan, and Purdue University in 1982, 1983, and 1988, respectively. He joined the Department of Electrical and Computer Engineering, NJIT, in 1988, and has been a professor since 1997. He is a technical editor of *IEEE Communications Magazine* as well as the *Journal of Computing and Information Technology*, and is General Chair of ITRE 2003. He was instrumental, while serving as its Chapter Chair, in rejuvenating the North Jersey Chapter of the IEEE Communications Society, which received the 1996 Chapter of the Year Award. He served as Chair of the IEEE North Jersey Section and on the IEEE Region 1 Board of Governors, 2001–2002, and currently serves on various IEEE committees. He was the recipient of the 1998 NJIT Excellence Teaching Award in Graduate Instruction and a 1999 IEEE Region 1 Award. His current research focuses on various aspects of high-speed networks and multimedia communications. He authored with E. S. H. Hou *Computational Intelligence for Optimization* (1997; translated into Chinese, 2000), and edited with B. Yuhas *Neural Networks in Telecommunications* (1994), both published by Kluwer.

Several US federal laws that are relevant to traceback were not written with computer networking in mind. Currently, an insufficient number of court cases and precedents make it difficult to understand all implications of the traceback.