Tracking and Tracing Spoofed IP Packets to Their Sources

Alaaeldin A. Aly, *College of IT , aly@uaeu.ac.ae*
Ezedin Barka, *College of IT, ebarka@uaeu.ac.ae*
U.A.E. University, Al-Ain, P.O. Box: 17555, U.A.E.

Abstract

As the Internet becomes increasingly important as a business infrastructure, the number of attacks on it, especially denial of service (DoS) attacks grows. A DoS attack is an attempt by a person or a group of persons to cripple an online service. Consequently, there are currently a lot of efforts being made to come up with mechanisms to detect and mitigate such attacks.

Research on IP traceback has been rather active since the late 1999 DOS attacks. Several approaches have been proposed to trace IP packets to their origins. This paper examines the current best practices and the most promising research approaches in a search for near-term and long-term solutions to the traceback problem. However, it is clear that technical approaches alone can never offer a complete solution to the problem. Along with the proposed technical solutions, the policy implications and issues brought by the technology are discussed.
This paper discusses a variety of methods that can help determine if received packets have spoofed source addresses. Our approach that depends on analyzing routers' log files is also discussed.

1. INTRODUCTION

Although access control technologies such as firewalls, are commonly used to prevent network attacks, they cannot prevent some specific attacks, including TCP SYN flooding. Consequently, more companies are deploying intrusion detection systems (IDS). The IDSs detect network attacks; however, they don't let us identify the attack source. This is especially problematic with Denial of Service (DoS) attacks, for example, because the attacker doesn't need to receive packets from the target host and thus can remain hidden. Several efforts are in progress in many different research and business places around the world to develop source-identification technologies to trace packets even when an attacker fakes its IP address.

The purpose of IP traceback is to identify the true IP address of a host originating attack packets. Normally, we can do this by checking the source IP address field of an IP packet. Because of a sender can easily fake this information, however, it can hide its identity. If we can identify the true IP address of the attack host, we can also get information about the organization, such as its name, and the network's administrator email address, from which the attack originated. Existing IP traceback methods can be categorized as proactive or reactive tracing. The proactive tracing detects attacks when packets are in transit while the reactive tracing starts after an attack is detected.

Existing IP traceback methods can be categorized as proactive or reactive tracing. The proactive tracing prepares information for tracing when packets are in transit. If packets tracing is required, the attack victim (target) can refer to this information to identify the attack source. Two proactive methods – packet marking [1] and messaging [2] – have been studied and reviewed. In packet matching [1], packets store information about each router they pass as they travel through the network. The recipient of the marked packet can use this router information to follow the packet's path to its source. Routers must be able to mark packets, however, without disturbing normal packet processing. In messaging approaches [2], routers create and send messages containing information about the forwarding nodes a packet travels through. The approach relies on the Internet control message protocol (ICMP).

The reactive tracing starts tracing after an attack is detected. Most of the methods trace the attack path from the target to its source (origin). The challenges are to develop effective traceback algorithms and packet-matching techniques. Various proposals attempt to solve these problems. Among those studied techniques are hop-by-hop tracing, hop-by-hop tracing with an overlay network [3], IPsec authentication [4], and traffic pattern matching [5]. In hop-by-hop tracing, a tracing tool logs into the router closest to the attached host and monitoring the incoming packets. If the tool detects the spoofed packet, it logs into upstream routers and monitors packets. If the spoofed flooding attack is still occurring, the tool can detect the spoofed

packet again on one of the upstream routers. This procedure is repeated recursively on the upstream routers until the tool reaches the attack's actual source IP address.

In hop-by-hop tracing, the more hops there are, the more tracing processes will likely be required. To decrease the number of hops required for tracing, hop-by-hop tracing with an overlay network is being used [3]. With the IPsec authentication [4], when the IDS detects an attack, the Internet key exchange (IKE) protocol establishes IPsec security associations (SAS) between the target host and some routers in the administrative domain. The last technique being surveyed is the traffic pattern matching in which the trace is done by comparing traffic patterns observed at the entry and exit points of the network with the Internet map [5]. A survey has been done to investigate the DDoS vulnerabilities and IP spoofing as mentioned in [6, 7, 8, 9, 10].

In this paper, we will develop our own approach to trace suspected packets to their sources. In our approach, routers log data about traversing packets as well as information about other nodes in the packet's path. A distributed management approach will be developed to enable tracing across networks with different access polices. Our approach is a reactive and it relies on hop-by-hop tracing. In our reactive approach, forwarding nodes such as routers log information about traversing packets on the Internet and then use the log data to trace each packet from its final destination to its source, hop-by-hop. Information about the packets remains in forwarding nodes as packets traverse allowing us to trace even a single attack packet to its source.

2. METHODS OF IP TRACEBACK

The purpose of IP traceback is to identify the true IP address of a host originating attack packets. Normally, we can do this by checking the source IP address field of an IP packet. Because a sender can easily forge this information, however, it can hide its identity. If we can identify the true IP address of the attack host, we can also get information about the organization, such as its name and the network administrator's email address, from which the attack originated. With IP traceback technology, which traces an IP packet's path through the network, we can find the true IP address of the host originating the packet. To implement IP traceback in a system, a network administrator updates the firmware on the existing routers to the traceback support version, or deploys special tracing equipment at some point in the network.

Existing IP traceback methods can be categorized as proactive or reactive tracing.

2.1 Hop-by-Hop IP Traceback

The most common and basic method in use today for tracking and tracing attacks is hop-by-hop traceback. This method is only suitable for tracing large, continuous packet flows that are currently in progress, such as those generated by ongoing denial-of-service (DoS) packet flood attacks. In a DoS flood attack, the source IP addresses are typically spoofed (i.e., they are forged addresses inserted into the source address field of a packet to disguise the true IP address of the machine that originated the packets), so tracing is required to find the true origin of the attack.

For example, assume that the victim of a flood attack has just reported the attack to their ISP. First, an ISP administrator identifies the ISP's router that is closest to the victim's machine. Using the diagnostic, debugging, or logging features available on many routers, the administrator can characterize the nature of the traffic and determine the input (ingress) link on which the attack is arriving. The administrator then moves on to the upstream router (i.e., the router one previous hop away that is carrying attack packets toward the victim). The administrator repeats the diagnostic procedure on this upstream router, and continues to trace backwards, hop-by-hop, until the source of the attack is found inside the ISP's administrative domain of control (such as the IP address of a customer of the ISP) or, more likely, until the entry point of the attack into the ISP's network is identified. The entry point is typically an input link on a router that borders another provider's network. Once the entry point into the ISP's network is identified, the bordering provider carrying the attack traffic must be notified and asked to continue the hop-by-hop traceback. Often there is little or no economic incentive for such cooperation.

2.2 Ingress Filtering

Much of the attacks on the Internet by attackers is accomplished using attack packets with spoofed source addresses. The occurrence of packets with spoofed source addresses, and their ability to transit the Internet,

can be greatly limited through cooperative efforts by ISPs, using a basic packet filtering approach called network ingress filtering.

For example, assume that an ISP provides Internet connectivity to a customer network and assigns the customer a fixed set of IP addresses. Assume that the connectivity is provided via the ISP's router R. To limit IP source address spoofing, the ISP places an ingress (input) filter on the input link of router R, which carries packets from the customer network into the ISP's network and onto the Internet. The ingress filter is set to forward along all packets with source addresses that belong to the known set of IP addresses assigned to the customer network by the ISP, but the filter discards (and optionally logs as suspicious) all packets that contain source IP addresses that do not match the valid range of the customer's known IP addresses. Hence, packets with source addresses that could not have legitimately originated from within the customer network will be dropped at the entry point to the ISP's network.

The widespread use of ingress filtering by all service providers would greatly limit the ability of an attacker to generate attack packets utilizing a broad range of spoofed source addresses, making tracking, and tracing the attacker a much easier task. Any attacker located within the customer network, in our example above, would either have to generate packets that carry the attacker's legitimate source address or (at worst) spoof a source address that lies within the set of IP addresses assigned to the customer network. So, even in the worst case, an attack originating within the customer network in our example can be traced to some machine in that customer network, simply by reading the source address on the attack packet. With the help of the administrator of the customer network, the search for the attacker can then proceed in a greatly narrowed search space.

3. SPOOFED PACKETS DETECTION METHODS

Detection methods can be classified as those requiring router support, active host-based methods, passive host-based methods, and administrative methods. Administrative methods are the most commonly used methods today. When an attack is observed, security personnel at the attacked site contact the security personnel at the supposed attack site and ask for corroboration. This is extremely inefficient and generally fruitless. An automated method of determining the whether packets are likely to have been spoofed is clearly needed. This section describes a number of such methods.

3.1 Routing methods

Because routers (or IP level switches) can know which IP addresses originate with which network interface, it is possible for them to identify packets that should not have been received by a particular interface. For example, a border router or gateway will know whether addresses are internal to the network or external. If the router receives IP packets with external IP addresses on an internal interface, or it receives IP packets with an internal IP address on an external interface, the packet source is most likely spoofed. In the wake of recent denial-of-service attacks involving spoofed attack packets, ISPs and other network operators have been urged to filter packets using the above-described method. Filtering inbound packets, known as ingress filtering, protects the organization from outside attacks. Similarly, filtering outbound packets prevents internal computers from being involved in spoofing attacks. Such filtering is known as egress filtering. It is interesting to note that if all routers were configured to use ingress and/or egress filtering, attacks would be limited to those staged within an organization or require an attacker to subvert a router. Internal routers with a strong notion of inside/outside can also detect spoofed packets. However, certain network topologies may contain redundant routes making this distinction unclear. In these cases, host based methods (discussed in section 4.2) can be used at the router. A number of IP addresses are reserved by the IANA for special purposes. These are listed in table 1. The addresses in the first group are private addresses and should not be routed beyond a local network. Seeing these on an outside interface may indicate spoofed packets. Depending on the particular site, seeing these on an internal address would also be suspicious. The other addresses in table 1 are special purpose, local only addresses and should never be seen on an outer interface.

Many firewalls look for the packets described in this section. Typically they are dropped when received.
Because firewalls have been a popular security product, research into routing methods has been active. Most all research has been in this area. Routers can also take a more active role in detecting spoofed packets. A number of advanced router projects have dealt with this and spoofed packet traceback.
These are discussed in section 6. We have proposed a number of proactive methods that can be used to detect and prevent spoofed packets.

One limitation of routing methods is that they are effective only when packets pass through them. An attacker on the same subnet as the target could still spoof packets. When the attacker is on the same

Ethernet subnet as the target, both the source IP address and the Ethernet MAC would be spoofed. If the spoofed source address was an external address, the MAC would be that of the router. This implies that other techniques are required.

3.2 Non-routing methods

Computers receiving a packet can determine if the packet is spoofed by a number of active and passive ways. We use the term active to mean the host must perform some network action to verify that the packet was sent from the claimed source. Passive methods require no such action, however an active method may be used to validate cases where the passive method indicates the packet was spoofed.

3.3 Active Methods

Active methods either make queries to determine the true source of the packet (reactive), or affect protocol specific commands for the sender to act upon (proactive). These methods have an advantage over routing methods in that they do not require cooperation between ISPs and can be effective even when the attacker is on the same subnet as the target. Active methods require a response from the claimed source. Only if the spoofed host is active (i.e. connected to the network and receiving and processing packets) can it be probed. A host that is heavy firewalled and cannot respond to probes is effectively inactive. Because inactive hosts are commonly used as source addresses in spoofed packets, if these packets are seen in an attack, it is likely they are spoofed. When hosts will not respond to any probes, passive methods will be required for corroboration.

TTL methods

As IP packets are routed across the Internet, the time-to-live (TTL) field is decremented. This field in the IP packet header is used to prevent packets from being routed endlessly when the destination host can not be located in a fixed number of hops. It is also used by some networked devices to prevent packets from being sent beyond a host's network subnet. The TTL is a useful value for detecting spoofed packets. Its use is based on several assumptions, which, from our network observations, appear to be true.?

IP Identification Number

As discussed in the section on Bounce Scanning, the sending host increments the Identification Number (ID) in the IP header with each packet sent. Because this is a value that is easily probed and changes in its value are predictable, we can use it to determine if a packet is spoofed. Unlike TTL values, IP ID numbers can be used to detect spoofed packets even when the attacker and the target are on the same subnet.

If we send probe packets to the claimed source and we receive a reply, the ID values should be near the value of questionable packets recently received from the host. Also, the ID values observed in the probe should be greater than the ID values in the questionable packets. If not the packets were likely not sent by the claimed source. If the host associated with the claimed source is very active, the ID values may change rapidly. To be effective, the probes must be done very close in time to receipt of the questionable packets.

.

OS Fingerprinting

The above techniques illustrate aspects of the more general task of OS fingerprinting where a series of various probes are used to identify the operating system of a particular host. Active fingerprinting refers to direct probing of a computer, while passive fingerprinting refers to monitoring traffic and comparing it to expected norms for different OSs. We can perform a limited passive fingerprint as we observe network traffic from a particular host, then by comparing this to an active OS fingerprint, we can determine if the two are likely to be the same OS. If not we can infer the packets are spoofed.

TCP Specific Methods

Flow Control

The TCP header includes a window size field. This is used to communicate the maximum amount of data the recipient can currently receive. This can also be interpreted as the maximum amount of data the sender can transmit without an acknowledgement from the recipient. This is the TCP flow control method. If the window size is set to zero, the sender should not send more data. If the packets we are receiving are spoofed, then the sender will never see the recipient's ACK-packets. This implies that the sender will not respond to flow control. If the recipient does not send any ACK-packets, the sender should stop after the initial window size is exhausted. If it does not, it is likely the packets are spoofed. One way of implementing this check is to always send an initial window size that is extremely small. If packets received exceed this threshold, we can infer the packets are spoofed. Because spoofing replies with the correct sequence number to multiple TCP packets may be challenging, most spoofed TCP connections do not progress past the first ACK-packet. This implies that the best chance to detect spoofed packets requires it be

done in the handshake. Fortunately the TCP handshake requires the host sending the initial SYN wait for the returned SYN-ACK prior to sending its first ACK packet. By setting the window size in the SYN-ACK to zero, we can we can determine if the sender is receiving (and responding to) our packets. If the sender sends an ACK-packet with any data, we know the true source is not responding to our packets, and were likely a spoofed packet.

Packet Retransmission

TCP uses sequence numbers to determine which packets have been acknowledged. An ACK-packet communicates to the recipient that all packets it has sent, up to and including the packet with the sequence number in the packet have been successfully received. When a packet is received with an ACK-number that is less than the minimum expected, or greater than the max expected, the packet is dropped and as a way to resynchronize the connection, a reply with the minimum expected ACK-number is sent. We can exploit these replies to probe for spoofed packets. By sending a probe packet, spoofed to be from the internal host, with an ACK number greater than the minimum expected, we can induce a resynchronization ACK from the host being probed. If the probe receives a RST in reply, we can infer the connection was spoofed. A concern with this method is that it may lead to an ACK-storm as both sides attempt to resynchronize. This method is best performed on a firewall where the probe reply could be captured. This will prevent the internal host from seeing the reply, and will prevent an ACK-storm.

Traceroute

Traceroute is a widely used network tool to discover the route from the site traceroute is executed on to another. When used to detect spoofed packets, it may tell you the number of hops to the true source. Unfortunately it is very slow and generally fails when the site being checked is behind a firewall. If the firewall blocks the probing UDP packets (or the ICMP replies), the traceroute program will know only the number of hops to the firewall. However, when the firewall is more hops away from the monitored site than the true site, traceroute will return a hop count greater than expected of the questionable packet. In this case, traceroute can be useful as a detector. Because of its performance, traceroute is a poor general technique for spoofed packet detection. However, in cases where the attacker is nearer the target than the true source site's firewalls, and the firewall will not allow probes to succeed, traceroute or similar techniques should be considered.

The issues with traceroute introduce a different method of spoofed packet detection base only on previously observed packets. Because the TTL and ID fields are set by the true source, we can learn the expected values for a particular host. Such passive methods are discussed in the next section.

3.4 Passive Methods

Passive methods are a logical extension of the reactive methods discussed earlier. Where observed data will have a predictable value, not relative to some prior packet, we can learn what values are to be expected and consider packets with unexpected values suspicious. Because TTL values are a function of a host's OS, the packet's protocol, and the network topology, all which are reasonably static, TTLs can be used as a basis for passive detection. Conversely, IP ID numbers, which generally have a strong relation to prior packets, do not make good candidates for the basis of a passive system. The next section describes several different passive methods and how they could be used to detect spoofed packets.

Passive TTL Methods

By recording, over a period of time, the TTL values of distinct source IP address/protocols we can learn which values are expected from particular hosts. We believe that these are reliable, predictable values of a given IP address/protocol. (See section 7 for experimental validation of this.) This will give us a reasonable basis for identifying suspicious packets from previously observed hosts. Our implementation of this compares observed packets to the expected TTL values for that packet. If the values were anomalous, the packet would be flagged as suspicious. In many cases, we will receive packets from hosts not previously encountered. These will have no entry in the table. Without further information we will not be able to know if the packet's TTL values are suspicious. How to flag such packets should be left up to the particular application.

However, by taking advantage of the fact that similar IP addresses are commonly the same number of hops away from a monitoring point, we can expand the above method to predict values for previously unseen packets. In addition to leaming IP address/protocol to TTL relations we can also learn IP subnet to TTL relations. The predictability based on subnets is not expected to be as high as specific IP address/protocols, but will provide additional information. Rather than use passive methods alone, by using them in

combination with reactive methods we can construct an efficient spoofed packet detection system. The reactive method can be initiated only when the packet seems suspicious. This minimizes the amount of probing required, and allows us to test packets using a number of methods. The specifics or our implementation are described in sections 5 and 7. One of the strengths of passive TTL methods is that they are resistant to network routing attacks. These occur when packets intended for a particular host are routed to another host posing as the first. Such an attack is not strictly packet spoofing because the packets are coming from the effective IP address of the sender. However, if the network distance between the two hosts has changed, we will identify these packets as spoofed. This allows passive spoofed packet detection to also act as a routing change detector.

OS Idiosyncrasies

We have identified a number of other features that can be used to find suspicious (possibly spoofed) packets. These include the expected source port for a TCP or UDP communication, expected ID values for certain packets, and type of service (ToS) or differential service code point (DSCP) values. The TCP window size has also been observed to be highly predictable given the source. Other useful features are likely. Basically, any that is specific to a particular host, OS, NIC, etc. is a potential identifier for that host. How useful a particular feature is depends on how predictable a particular feature is and how likely another computer will generate the same value as the claimed source. Features with values common to many computers will tend to generate false negatives while those that vary significantly will tend to generate false positives.

4. THE PROPOSED APPROACH

Denial-of-service (DOS) attacks are a pressing problem in today's Internet. Their impact is often more serious than network congestion due to their targeted and concentrated nature. In a distributed DOS (DDOS) attack, the attacker uses a number of compromised slaves to increase the transmission power and orchestrate a coordinated flooding attack. Particularly, DDOS attacks with hundreds or thousands of compromised hosts, often residing on different networks, may lead to the target system overload and crash.

Because the current Internet routing infrastructure has few capabilities to defend against IP spoofing and DDoS attacks, we need to design a new defense mechanism against these attacks. In particular, our proposed approach is to defend against these attacks and should satisfy the following properties:

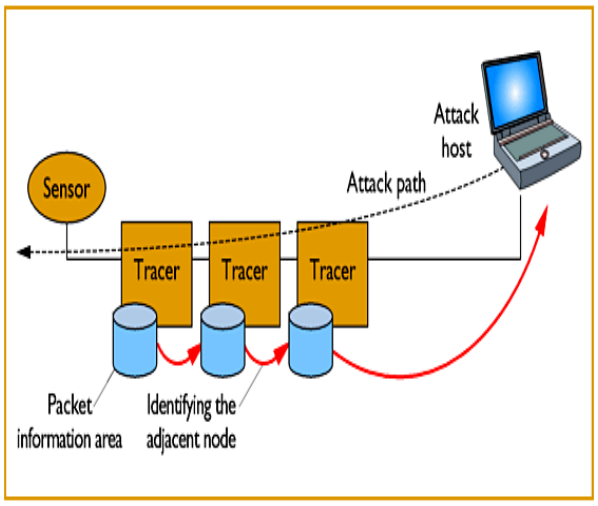
- Fast response: The proposed approach should be able to rapidly respond and defend against attacks. Every second of Internet service disruption causes economic damage. We would like to immediately block the attack.
- Scalable: Some attacks, such as TCP hijacking, involve only a small amount of packets. However, many DDoS attacks are large scale and involve thousands of distributed attackers and an even larger number of attack packets. A good defense mechanism must be effective against low packet count attacks but scalable to handle much larger ones.
- Victim filtering: Almost all DDoS defense schemes assume that once the attack path is revealed, upstream routers will install filters in the network to drop attack traffic. This is a weak assumption because such a procedure may be slow, since the upstream ISPs have no motivation to offer this service to non-customer hosts and networks.
- Efficient: The proposed approach should have very low processing and state overhead for both the routers in the Internet and, to a lesser degree, the victims of the attacks.
- Support incremental deployment: The proposed approach is only useful and practical if it provides a benefit when only a subset of routers implement it. As an increasing number of routers deploy the scheme, there should be a corresponding increase in performance.

Also, the deployment of the solution should not leak proprietary information about an ISP's internal network, as some ISPs keep their network topology secret to retain a competitive advantage.

CONCLUSION

IP traceback has several limitations, such as the problem with tracing beyond corporate firewalls. To accomplish IP traceback, we need to reach the host where the attack originated. It is difficult, however, to trace packets through firewalls into corporate intranets the last-traced IP address might be the firewall's address. Knowing the IP address of the organization's network entry point, however, allows us to obtain information about the organization where the attacker's host is located, such as the organization's name and

the network administrator's e-mail address. If we can identify the organization from which the attack originated, the organization can often identify the user who launched the attack.



Basic method of the traceback approach. Forwarding nodes, or tracers, store data from an incoming packet as well as its datalink-level identifier in the packet information area, and they identify the adjacent forwarding node.

ACKNOWLEDGMENT

This work has financially supported by the Research Affairs at the UAE University under a contract no. 03-05-9-11/04.

REFERENCES

[1] S. Savage et. al. "**Practical Network Support for IP Traceback**," *Proc. 2001 ACM SIGCOMM*, vol. 30, no. 4, ACM Press, New York, Aug. 2001, pp. 295-306; available on line at http://www.cs.washington.edu/homes/savage/traceback.html.

[2] S. Bellovin, M. Leech, and T. Tylor, "**ICMP Traceback Messages**," *Internet draft, work in progress*, Oct 2001; available online at http://www.ietf.org/internet-drafts/draft-ietf-itrace-01.txt

[3] R. Stone, "**CenterTrack: An IP Overlay Network for Tracking DoS Floods**," *Proc. 9th Usenix Security Symposium, Usenix Association*, Berkeley, California, Aug 2000; available online at http://www.usenix.org/publications/library/proceedings/sec2000/stone.html

[4] H.Y. Chang et. al., "**DecIdUous: Decentralized Source Identification for Network-Based Intrusions**," *Proc. 6th IFIP/IEEE International Symposium. Integrated Network Management, IEEE Comm. Soc.*, New York, May 1999, pp. 701-714.

[5] K. Ohta et. al., "**Detection, Defense, and Tracking of Internet Wide-Illegal Access in a distributed Manner**," *Proc., INET 2000, Internet Society,* Reston, VA, July 2000; available online at http://www.isoc.org/inet2000/cdproceddings/1f/1f_2.htm.

[6] CERT, "**TCP SYN flooding and IP spoofing attacks**," *Advisory CA-96.21*, September 1996.

[7] Vern Paxson, "**An analysis of using reflectors for distributed denial-of-service attacks**," *Computer Communication Review*, 31(3), 2001.

[8] Mike Kristovich, "**Multi-vendor game server DDoS vulnerability**," http://www.pivx.com/kristovich/adv/mk001/, November 2002.

[9] CERT, "**IP spoofing attacks and hijacked terminal connections**," *Advisory CA-1995-01* www.cert.org/advisories/CA-1995-01.html, February 2001.

[10] L. Joncheray, "**Simple active attack against TCP**," www.insecure.org/stf/iphijack.txt, February 2001.