2007

An approach in identifying and tracing back spoofed IP packets to their sources

Krishnun Sansurooah
Edith Cowan University

DOI: 10.4225/75/57ad3b667ff26
Originally published in the Proceedings of the 5th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, December 3rd 2007.

An approach in identifying and tracing back spoofed IP packets to their sources.

Krishnun Sansurooah
School of Computer and Information Science (SCIS)
Edith Cowan University Perth, Western Australia.
Email: ksansuro@student.ecu.edu.au

Abstract

With internet expanding in every aspect of businesses infrastructure, it becomes more and more important to make these businesses infrastructures safe and secure to the numerous attacks perpetrated on them conspicuously when it comes to denial of service (DoS) attacks. A Dos attack can be summarized as an effort carried out by either a person or a group of individual to suppress a particular outline service.

This can hence be achieved by using and manipulating packets which are sent out using the IP protocol included into the IP address of the sending party. However, one of the major drawbacks is that the IP protocol is not able to verify the accuracy of the address and has got no method to validate the authenticity of the sender's packet. Knowing how this works, an attacker can hence fabricate any source address to gain unauthorized access to critical information. In the event that attackers can manipulate this lacking for numerous targeted attacks, it would be wise and safe to determine whether the network traffic has got spoofed packets and how to traceback. IP traceback has been quite active specially with the DOS attacks therefore this paper will be focusing on the different types of attacks involving spoofed packets and also numerous methods that can help in identifying whether packet have spoofed source addresses based on both active and passive host based methods and on the router-based methods.

INTRODUCTION:

Referring to RFC 1791, (1981) packets that are sent out using the IP protocol include the IP address of the sender. After receiving this packet, the recipient directs replies to the sender using the original source address. Nonetheless, the correctness of this address is not verified by the IP protocol that unfortunately has no way of validating the packet's source if not only been based on the sender's IP address. Therefore this involves that an attacker can at any pointing time fake the source address and act as a legitimate party to gain access to unauthorized details or information. Most of the time sending spoofed packet is carried out to legitimately access unauthorized information.

Spoofing of network traffic can certainly occur at many different layers. One of the layers that could be affected by is the network layer which is responsible in dealing with MAC spoofing or at a non IP transport layer such as IPX, NetBEUI or even at an application layer in the instance of Email spoofing.

Even through tough access control technologies such as firewalls which are mainly used in protecting the network, they are helpless when it comes to specific attacks ranging from SYN-flood, TCP connection spoofing, Smurf and many more. Subsequently with these attacks increasing, more and more companies are now turning towards deploying Intrusion Detection System (IDS) onto their network. However, it does detect the network attacks and hence display the alerts but unfortunately does not identify the attacker's source. This is quite enigmatic especially when it comes to DOS attacks because the attacker normally can remain masked due to the fact that these attacks, the attacker does not have to interact with the targeted host as s/he does not need to receive any packet as s/he is initialing the attack. The focus of this report is to

illustrate an overview of the numerous ways that can be used to determine whether the received IP packets on the network has been spoofed and if so, how to trace them back to their originators or their source addresses. This process is most common known as IP traceback. The main concept behind the IP traceback is to determine the exact IP address of the source from which the attacks are being launched. Despite that this can normally be gathered by locating the IP address field from the IP packet, the attacker can unfortunately easily manipulate and changed these details, thus masking its original and true identity.

However, the concept of IP traceback is not well defined as to what it normally should be performing. Its purpose is mainly to identify the true IP address and the source of its attacker, in other words, the ability of identifying the source of a particular IP packet, its destination and an approximate time of reception of the packet. IP traceback can hence be summarized as belonging to two different methods: proactive and reactive.

Proactive Tracking Methodology

This method would involve detecting and tracing attacks when packets are in transit. If packet tracing is needed, the victim can therefore refer to this information to identify the attacking source. However the proactive methods can be further split into two different proactive methods namely marking and packet messaging respectfully and described below

Packet Marking:

This would involve packets that contain information about each and every router that they go through as they (IP packets) has through the network. Therefore, this means that the receiver of the designated packet can make use of the information held by the router to trace back packet's route to its originator. It is imperative that routers can imprint and sign packets without interrupting the normal packet processing.

Message Marking:

In this particular approach, the different routers, through which the packets travel across, generate and broadcast messages with call the information about the forwarding nodes that a particular packet transit across.

Reactive Tracking Methodology

The reactive tracing method operates differently to the proactive one. In this approach, the tracing will only commence when an attack has been perpetrated and following its detection. However, the numerous trials in developing a practical traceback algorithm and packet matching techniques have tried to resolve these dilemmas. Among those analyzed approaches are hop-by-hop tracing, IPSec authentication and monitoring traffic patterns matching.

The focus of this paper will be to identify the various types of attacks involved with spoofed packets and also how to analyze the tracking back of suspicious packets to their originator(s). Whilst, accessing the routers logs about all the data packets that have been passed through and even to other nodes. However, the methodology used in this particular approach will be reactive and using the hop-by-hop tracing with a probabilistic packet marking scheme.

IP TRACEBACK:

During the past decade, a lot of attention has been focused on the security of Internet infrastructure in place as being part of transmission, reception and storage of paramount importance within the increasing e-commerce applications. Yet, with high-profile Distributed Denial-of Service (DDOS) attacks, numerous ways have been elaborated to identify the source of these attacks and one methodological approach is using IP traceback.

Normally, IP traceback is not restricted to only DOS or DDOS attacks but with the ability to forge the IP address of those packets make the identification of the originator even harder and in such routine approaches of locating the system (attacker) with a given IP address (e.g. Traceroute) is no longer feasible due to the fact that the packet has already been spoofed. Belenky, A. & Ansari, N. (2003) implies that more advanced approaches of locating the source of attacking packets are needed. They also mentioned that identify the source of the offending packet would not necessarily means identifying the attack originator as these packets may be a host linked in a chain a reflector, a zombie or a device that by the attacker at an earlier stage. However, they did mention that IP traceback approaches are not meant to impede or cease those attacks but they are used to identify the source(s) of the initial incriminating packets during and after the attack.

IP TRACEBACK CLASSIFICATION:

End-host IP Traceback

Probalistic Packet marking (PPM)

This approach is structured around the routers that imprint the packets that flow through them through either their address or part of their address. This is normally carried out in a randomly process. This PPM technique also known as hop-by-hop tracing introduced by Savage, S. et al. (2001) originally and hence been improved by Song, D.X. & Derrig, A. (2001) in its coding and security was primarily targeted at both DOS and DDOS.

Figure 1 below gives an overview of how the PPM works where attacker would launch an attack towards victim A and as shown in Figure 1, assuming that the packet travel path is R1, R2, R4 and R12, each router enforcing PPM recognize the packet flow and prior to routing them to their destinator, it probalistically imprint them with its partial address – i.e. the router's partial address into to IP address header. Therefore when the victim acknowledges reception of enough packets, it can then remodel the addresses of all PPM-enabled packets and hence reconstruct the attack path. Obviously to reconstruct the whole attack path, a large number of packets would be required as well as the reconstruction of the data frames.

We do note that PPM can deal with the modification of packet which are directed to the victim. But, when it comes to packet generation transformation by a reflector, traceback will only be obviously, the traffic will become fragmented and will be corrupted without having whatsoever impact on the traceback. This means that when fragmentation takes place, usually the ID field is the field that is being marked and if only a single fraction of the source is marked, the reassembly process will not be possible at the destination. Even that this might pose a problem, traceback would still be in a position of retracing the path due to the fact that the marking would have taken place before the reassembly process. This is resolved by opting for a much reduced option in the marking of the packets but that need to be understood is that on doing so, this will definitely increases the number of packets needed for reconstructing the path.

Another issue with PPM is when using tunneling technology, it must be ensured that the markings are not removed prior to the headers are removed.

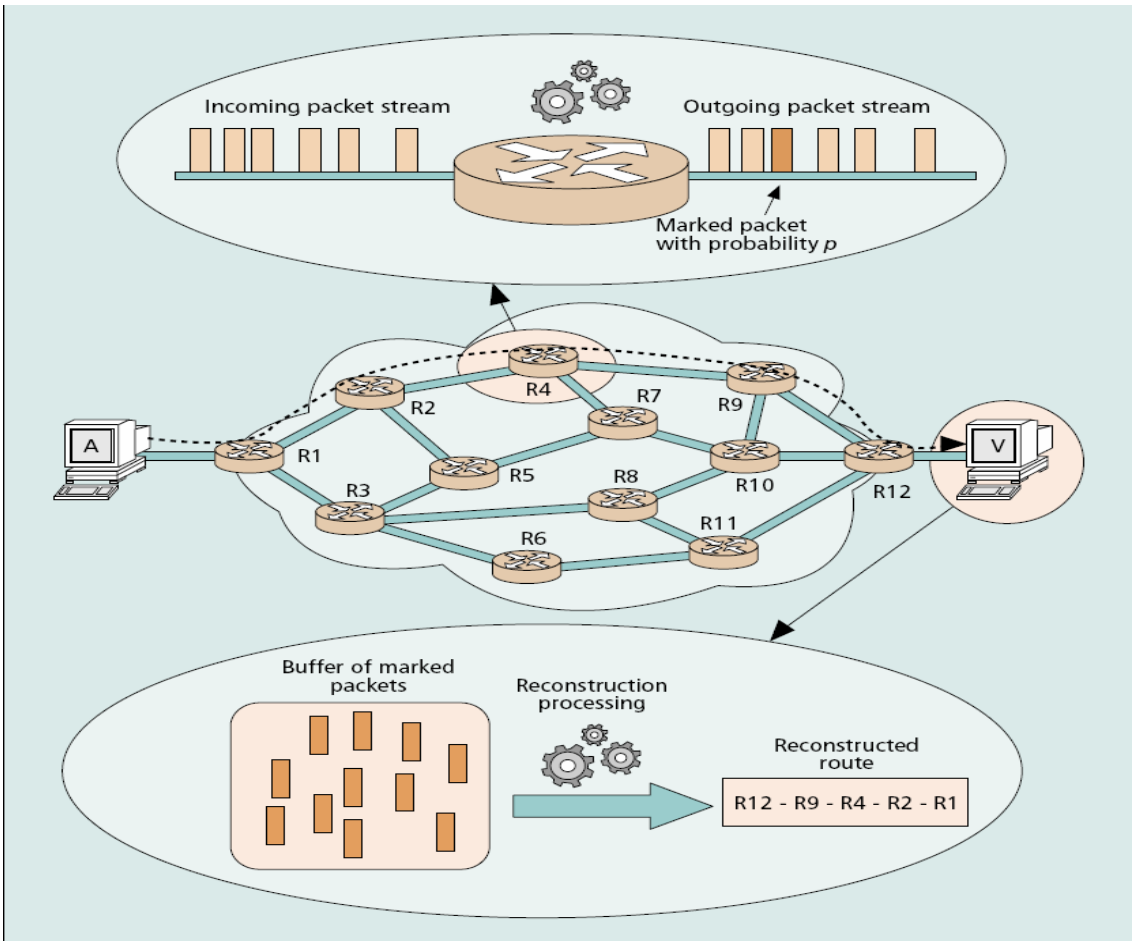

Figure 1 Probalistic packet marking (Belenky, A. & Ansari, N. (2003))

ICMP Traceback

With the ICMP traceback, the concept of tracking down the full pathway of the intrusion is completely different from PPM. In figure 2 there is an illustration of how an ICMP traceback schema operates. According to Belenky, A. & Ansari, N. (2003) every router on the network is set up in such a way that they have the ability to pick any packet at random (one in every 20,000 packets recommended) and hence produce an ICMP traceback message which would be targeted to the corresponding destination of the selected packet. The ICMP traceback message would normally consist of the following information:

i) The previous hop,
ii) The next hop,
iii) A timestamp of the packet

However, since there are numerous bytes of the traced packet which are possible, which are duplicated in the payload of the traceback message. Therefore the Time-To-Live (TTL) field is extended to 255 in order to be used in identifying the attack pathway with the ICMP traceback; the routers sitting on the network pathway will produce a completely new packet with an ICMP traceback message. The entire opposite of how PPM would handle this situation. The traceback information was entirely in-band. If we go by the assumption that the victim is under a DOS attack it would definitely means that the volume of packets going through would be huge hence afterward capturing all the addresses of the different routers on the attack pathway that generate the traceback message.

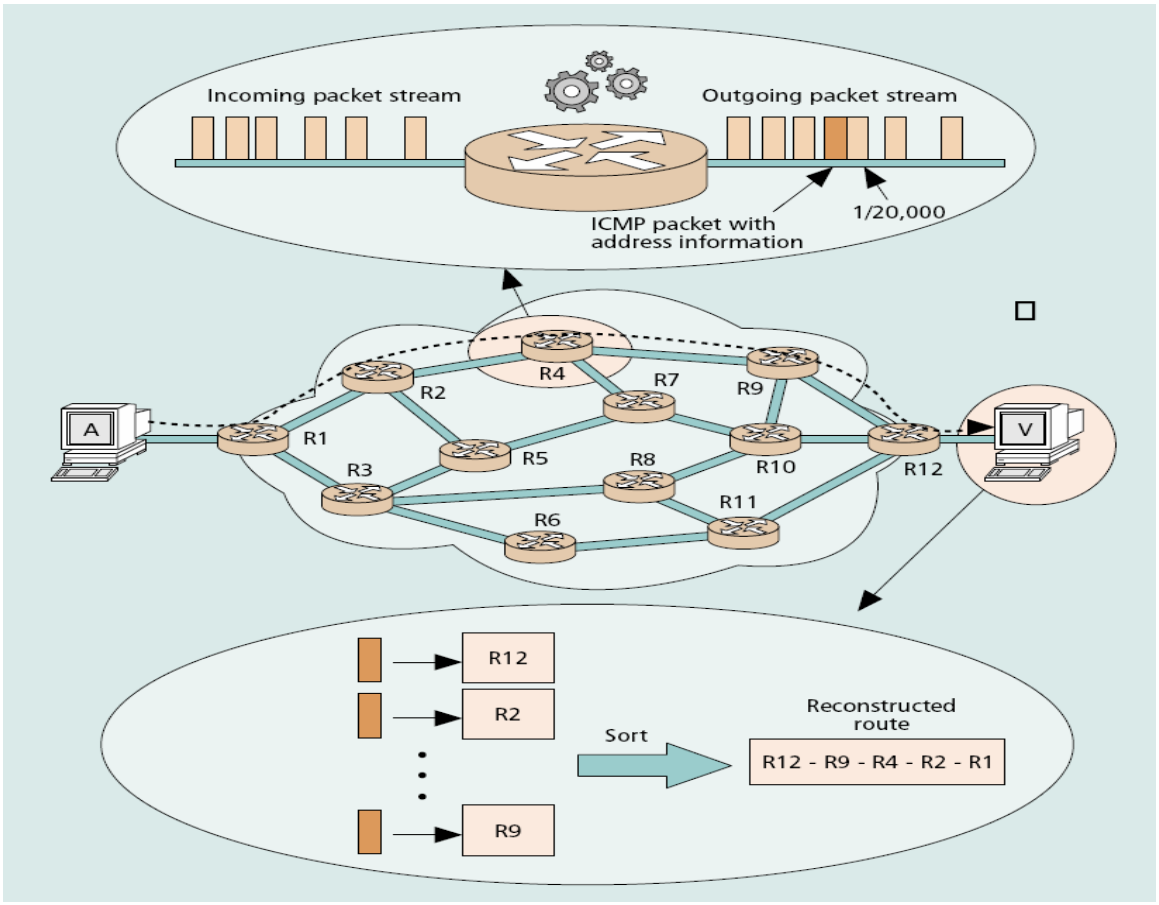

Figure 2 shows the process of the ICMP Traceback (Belenky, A. & Ansari, N. (2003)

Wu. F.S et al. (2001) did mention that while this methodological approach is quite conducive and probably secure, the probability of getting a meaningful traceback message is very minimal if a major DDOS attack is cascaded onto the victim mainly if proper attention been meticulously invested in minimizing the chances of detecting the traceback messages.

However, this can be resolved by associating a particular score to every traceback message created. If there is any allocation, the value will be affected therefore, to deploy and implement the traceback and its attack path reconstruction based on the ICMP traceback will involve a change in the organization's routing tables of the routers sitting onto the network. We need to keep in mind that prior to start tracing back the attack, all the software of the routers would need to undergo upgrades from their respective vendors and then only the ICMP traceback must be activated at the Internet Service Provider (ISP) interaction. Furthermore, with ICMP traceback, routers can be set up individually thus allowing good scalability. The number of packets required for reconstructing the attack pathway is planned and based on thousands since the chances of producing an ICMP traceback message is 1/2000 an for partial stratagem to be effective, the victim must be conscious about the network topology and the routing of the network. We also have to keep in mind that the reconstruction of data frames would consist of thousand entries thus leading to require enormous memory to process those entries. If in case that a router that is responsible for the marking of the packets happen to be corrupted, it can hence be reprogrammed to produce incorrect traceback messages thus giving out false reconstruction attacks pathways. Based upon Wu. F.S. et al. (2001) described in his report that handling major DDOS attacks with ICMP Traceback was possible but would not be true if a large number of reflectors were to be used. Therefore, the capability very similar to Probalistic Packet Marking thus leading to the conclusion that transformation involving reflector prove to be more difficult thus confining the limit of the traceback to the reflector.

Packet Logging IP Traceback:

According to Snoeren, A.C et al, (2002), this scheme is more commonly known as Source Path Isolation Engine (SPIE). With packet logging also referred as hash-based traceback, every single router on the network keep some data or information of every single that have passed through that particular router so that later, it can hence determine if that packet have already been through. The mechanism though which that hash-based IP traceback operate is that routers on the network are called data generation agent (DGAs) and since that network is symmetrically divided into zones. In each and every zone, SPIE collection and reduction agent (SCARs) are linked to all DGAs thus allow a communication link to be established as depicted in figure 3. Also the SPIE traceback Manager (STM) is a central unit that has a link to IDS of the victims.

So basically when packets flows across the network, partial information of packets get captured and stored in the DGAs. This partial information consist of the IP header and the initial 8 bytes of the payload of each packet is hashed and then recorded in bloom filter which reaching 70% of its capacity is archived for later analysis and a new one is used. However, these bloom filters are used during the capturing stage and to the time taken to use of these bloom filter is known as a time period. Having said so, DGAs can capture any transformations that occurs an influence on the field. Normally the type of transformation and the information required to reconstruct the attack pathway are recorded in a Transform Lookup Table (TLT) which each bloom filter has it's won TLT for it time period.

Therefore when the STM is notified of an attack from the victim's IDS, appropriate request are transmitted to SCAR which in turn look up for recorded partial information and transformation tables from DGAs from the allocated time period. After analyzing and comparing the tables, SCAR will be able to trace which router in the zone forwarded that packet. It is then the responsibility of the scar to retrace the router through which the packet has been going through and finally send a report to the STM. With this schematic hash-based IP Traceback approach, it can easily handle massive DDOS attacks. It is quite normal that a bigger amount of memory is required by the SCAR and the STM which is dedicated to the traceback processes. Given that this scheme is extremely difficult to bypass, is can therefore handle more or less any packet transformation.

Specialized Routing IP Traceback:

With this approach, the introduction of Tracking Router (TR) onto the network checks all the traffic that flows through the network and this is achieved by routing all the packets through the TR. This is then realized by creating a Generic Route Encapsulation (GRE) from every interface of the router to the TR. This architecture is illustrated in Figure 3 with the TR in the centre and with all of the edge routers on the network connected to with GRE tunnels using a star-like logical network is known as an overlay network.

However a single TR will not be capable of handling this load of packet from the entire network thus having several TRs which can still be logically implemented as a single TR that will be using signature-based intrusion detection. With this approach, if an attack is sensed, this means that the source of the attack can be traced back because it is only one hop away according to Belenky, A. & Ansari, N. (2003).

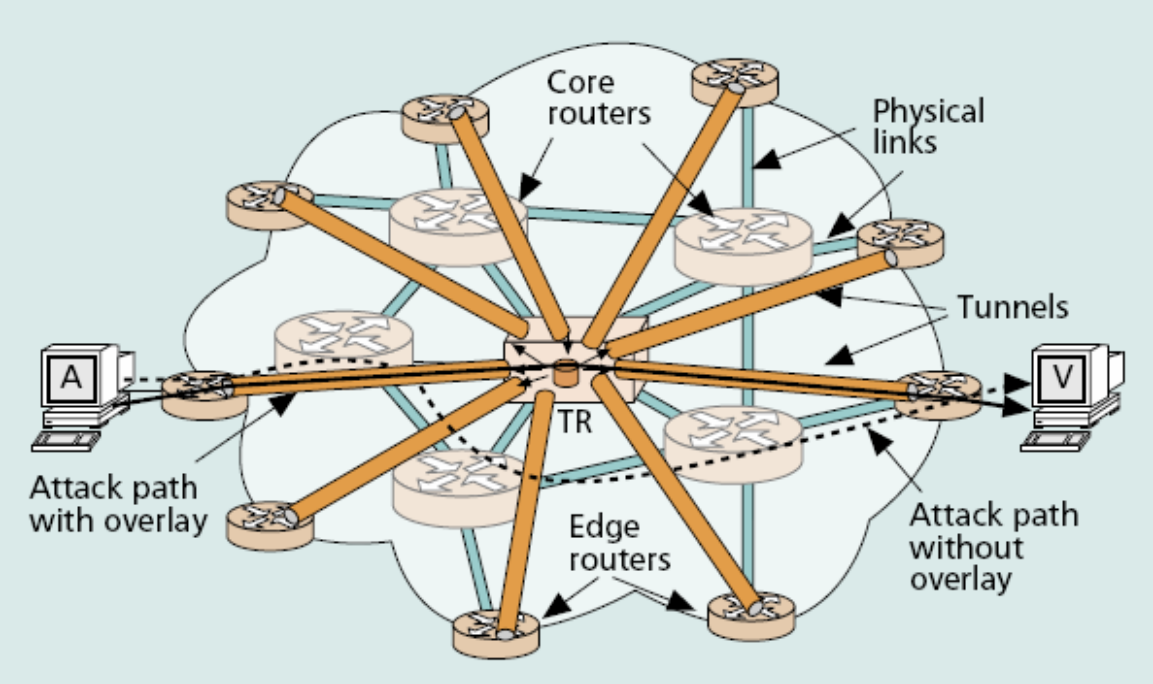

Figure 3. depicts the overlay network (Belenky, A. & Ansari, N. 2003)

This method make use of the most common features that are available on routers of today consequently the involvement of ISP is massive as it will have carry out a traceback and identify the source of the attack on its own.

Nevertheless this approach does have an extreme condition; it will only function within a single domain, therefore for the overlay network to be effective over ISPs, it would be wise to connect all the TRs into a single system. Of course with the method, a single packet is necessary to traceback an attack provided that the attack has been identified and reported. This then happen as soon as the IDS sitting on the TR identified the attack, it would trace it to its endpoint of the GRE tunnel. Now, if the edge router has numerous interfaces then it would be impossible to know exactly from which interfaces the attack was launched. However with this approach has a give-and-take collaboration between overhead and protection. But if the tunnels are mounted with IPSec, consequently the overhead bandwidth will increase and so will the level of protection. Moreover, this approach would be suitable to manage extensive DDOS attacks where tracing back the source of any particular packet ever to the edge of the network. Also handling packet transformation will not be a problem with this method.

IPSec Traceback with IPSec

This method is normally introduced as forming part of an intrusion detection structure specially designed for network based mask known as ***DECIDUOUS*** Chang H.Y et al. (1999). Given that this particular framework is far beyond the scope of this report, the method of operation in detecting the source address of any attack is of great significance. Therefore this approach relies on the fact that the complete network structure is of understanding and control to the system. This denotes that if at any pointing time there is an IPSec security involvement between an inconsistent router and the victim. However, if the identified packets are picked up by the security associations, therefore the initial attack is further than this router but if the opposite happens, if the identified packets are not detected, it will denote that the origin of the attack lied between this router and its victim. Thus allowing us to possibly identify a single or a group of routers from where the attack was launched.

Following the explanation, in figure 4 below when an attack is sensed, an IPSec security association between Router 4 (R4) and victim denoted by the letter ‘V’. In fact if ‘A’ being an attacker, his or her

attack packets will have to flow through the network and the tunnel thus requesting them to be authenticating before hopping through the tunnel. Then the tunnel from Router 1 (R1) to the victim is created. It is also noted that from Router 4 (R4) to the victim point 'V' there will be 2 tunnels that will encapsulate data traffic from the attacker. (This is not actually represented onto the figure 4)

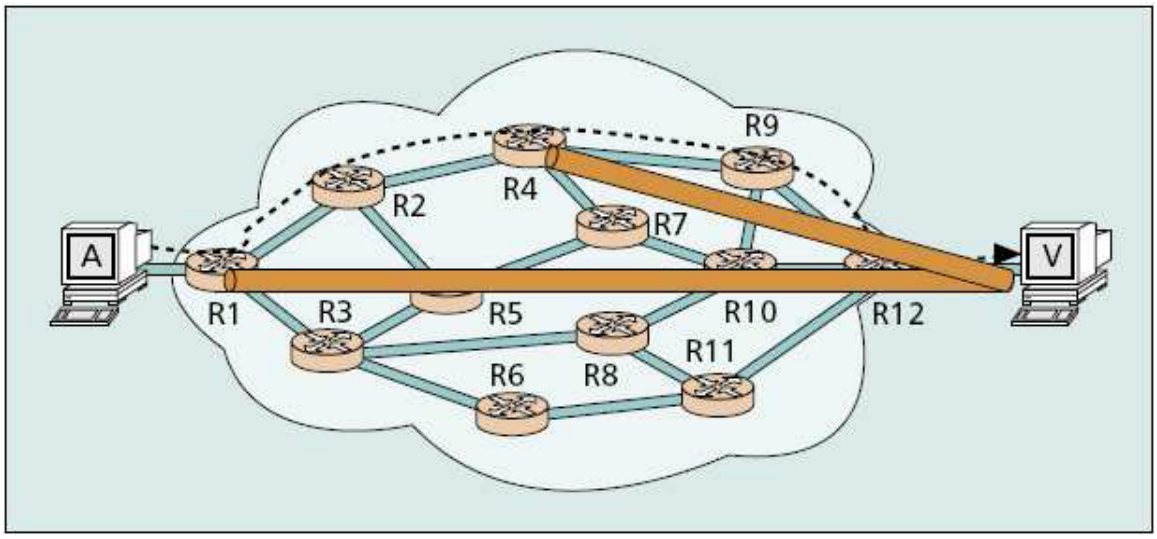

Figure 4 illustrating the IPSec Traceback approach (Belenky, A. & Ansari, N. 2003)

The second tunnel will perform its encapsulation over the first tunnel. Moreover, using two security associations to authenticate the traffic flow, it is obvious that if the attack was initiated behind RA but if the attack were to be authenticated by the only first tunnel then it would clearly identify the attacker would lies between Router 1 (R1) and Router 4 (R4) it would be on Router 2 (R2).

The system, however, will undergo numerous possible iterations in order to consider all the viable pathways before determining with which routers the victim should lean on for the IPSec security associations given that the source address is unknown. With this approach, the only interaction would come from the ISP which will have to communicate its network topology to its entire client in order to create the IPSec tunnels to the routers. Yet we to understand that in case 'shared key' authentication is used, every systems need to be aware of any changes on any routers on the network. We also assume that the authentication process will include digital certificate to be used at security associations. It is also to mention that with this approach the number of packets to perform traceback is low compared to the previously discussed methods.

This approach is very secure and even packet transformation is not an issue, being even capable of tracing back major DDOS attacks by tracing the path individually but we have to understand that when DDOS attacks are being performed, DDOS can themselves be targeted since ISPs have to remain open for clients to create IPSec tunnels thus making it unsuitable to manage complex DDOS attacks.

SPOOFED PACKET ATTACKS:

Packet spoofing can actually be part of different and various attacks type. So having the knowledge of how they operate would definitely reveal itself to be fruitful as then we know how they behave. One of the major aspect in whichever packet spoofing attack types, it does not have to receive packet replies from the targeted source. Therefore this part of this report will elaborate on the different types of attacks and discept their security associations.

SMURF Attack

According to Computer Emergency Response Team (1998), SMURF attack can be defined as an invasion on the network that floods that network with a huge amount of requests that unfortunately disrupt the

normal traffic. This is hence carried out by sending spoofed ICMP echo requests (ping) packet to a subnet broadcast. Therefore when a broadcast address is picked up by all the nodes on the subnet, this will then drive each active host to send an echo reply to the source where here in this attack. The source address is directed to the address of the target. This then amplifies causing a huge amount of packet to be generated and directed towards the target. This would definitely congest the network thus causing a major degeneration of the service on its network. Again Smurf attack is much more concerned with the multiplication of the packets and address spoofing to overflow the targeted network. Moreover, packet return is of no importance to the attacker as it is not desired. For successfully accomplish this type of attack, the attacker should be able to grab the broadcast address and then create to ICMP echo requests which unfortunately are broadly available.

SYN-Flood Attack

SYN-Flood attacks the most classic denial-of-service (DOS) attack where as mentioned earlier return packets are irrelevant. With SYN-Flood attacks, the attacker has to continually send a huge number of TCP SYN packets to the target which the host in return will acknowledge by sending an acknowledgement packet to the pretended sender. The sending host will wait for an acknowledgement reply which will never be sent out by the attacker forcing the host to engage in an undetermined wait thus causing the buffer on the target host to be tied up. Eventually when the entire buffer is used, no further network connection would be possible. Obviously the connection will finish by timing-out where the kernel will release the buffer and allow for new connections. Because of the huge amount of packet sent earlier in the SYN-Flood by the attacker is more likely to use the buffer again rather than a normal packet from a genuine connection. This SYN-Flood attack is not interested with the return packets but for the attack to be successful, the attacker will have to spoof source addresses from host that are non-existent or inactive.

Bounce Scanning

Using scanning in general resides a difficulty that the attacker must be able to view the replies and hence make it quite complex to use spoofed addresses. One of the easiest ways of achieving this type of attack is to spoof the address of a different computer on your network subdivision and hence listen to the network traffic for echo to the spoofed address. Following a report published by security focus (2001), a brilliant option would be to make use of spoofed packets and then to unwillingly listen to the targeted replies.

This particular type of attack normally exploit the IP header known as the “identification number” field. This number is usually incremented by one each time that a packet is sent over. Therefore the bounce attack makes use of that to send spoofed SYN packets to the targeted host through a part. However, Wu, S. et al (1998) mentioned that of the port is closed, a reply is generated back with a reset. Where on action is undertaken when it is received by the spoofed host. After all in the case that the port is opened, the target still replies back to the spoofed source through an acknowledgement. However given that the spoofed host is not responsible for launching the SYN-flood attack, it therefore transmit a reset to the target whilst still incrementing the IP id number.

For these particular attacks to be successful there are 3 main factors that need to be considered:

1) Scrutinize the spoofed host requesting to locate it actual id number.
2) Direct the spoofed scan packet to the targeted source.
3) Reconfirm the id number of the spoofed host

Once these 3 main areas have been achieved by the attacker, s/he can easily determine whether the targeted host’s port was either open or closed denoted by the id member incrementing by one would return port was closed and if it were to incremented by 2 then this would mean that the port was opened.

However, the attacker has to make sure that other packets are not directed to the spoofed host during the scanning, this is therefore achieved by either selecting a host to spoof with some or no network activity at all and such example would be a printer onto the network. On the other hand, if the spoofed host does not show any increase in the id numbers by at least one, the attacker can therefore make use of a multitude of

queries to each port and therefore deduce its mode by monitoring the changes occurring with the id numbers.

TCP Spoofing

What make TCP connection spoofing attack quite unique is that it is a combination and coordination of more than one attack. This would mean creating a DOS attack on one hand and on the other hand spoofing packets of the attack target. SO, basically to conduct a DOS attack on a trusted host could be anything that would prevent the trusted host to send out reset packet to the host in the instance of a SYN-Flood attack. The other form of attack would require the device to be transmitting spoofed packet to the target while impersonating the source host. Moreover, due to the DOS being launch prior to sending the spoofed packet to the target, the trusted host unfortunately cannot reply to the packets being received from the target. In addition the attacker can forcefully lead the target to believe that the send packets are from a trusted source thus later allowing the attacker to use the target to act as a trusted host at a pointing time.

However, this type of attack is quite complex and hence does require some knowledge as TCP does require reply packets to carry the sequence number of the former packet if the attacker cannot examine the packet, therefore she/he would have to guess the sequence number which could be made very hard to guess but is still feasible to achieve.

Zombie Control attack

McNevin, T. (2005), a DDOS has two primary goals, firstly to flood the victim's server and secondly to consume the most of the bandwidth of the victim. Nonetheless DDOS attacks normally consist of attacker(s), several intermediary computers, and finally many "zombies". Zombies are when an attacker has been able to infiltrate other computers through their weaknesses. Once the attacker has gain control of the machine, she/he can install tools, or programs that will allow that attacker the ability to communicate back and with other zombies. McNevin, T. (2005) also pointed out that it is very probable that an attacker might go thought the process of recruiting numerous zombies over an indefinite period of time and when the attacker have build up a huge network of zombies begin flooding packets towards its victim(s). In other words when the attacker decide to launch his or her attack, the initiator(s) will hence convey the message to the intermediary computers which will trigger the zombies to start flooding insignificant data in the diversion of the victim. However, this data traffic flow is not always ludicrous as the attacker may mask their traffic to appear like genuine and appropriate traffic to overcome any filtering defenses in detecting packet attacks.

Above are some of the ways through which spoofed packets could be used for different DOS or DDOS attacks. It is somehow useful to know if the packets have been spoofed or not as it will definitely help in mitigating attacks, or even help to traceback the true attack source.

SPOOFED PACKETS DETECTION METHODS

Spoofed packet detection methods can be categorized as of those depending upon router sustenance, passive host based, active host-based methods and finally upon administrative methods which is one of most frequently used methodology. This implies that when an attack occurs, the responsible personnel at the attacked location will liaise contact with the authorized personnel at the supposedly attack site and ask for an acknowledgement which is totally delicate. Therefore, the need of having computerized ways and means of detecting whether IP packets have been spoofed. This section of this report will have a closer look into the different methods and approaches in detecting spoofed packets.

ROUTING APPROACHES AND METHODS

Routers are devices that can identify IP addresses and through which network interface to they originally come from and can also point out packets that should not have been received by an interface. This therefore

means that a router will definitely be able to identify if addresses are either for the internal or external network. Nonetheless, if the router is addressed with IP packets with external IP address on an internal interface and vice versa, therefore it may come to the conclusion that the packet source has probably been faked.

Recently with DOS attacks including spoofed attack packets, methods to counter measure these treats have been developed and are put in place filtering methods are being implemented most commonly known as Ingress Filtering, which filter inbound packets thus protecting from outside attacks. Similarly for filtering outbound traffic known as Egress Filtering which prevent internal machine to be compromised from spoofed attacks. Yet, if all the routers would have that sort of filtering in place, both Ingress and/or Egress then it would push an attacker to corrupt that router. However, internal routers with an adequate understanding of the inside and outside can spot fake packets but with the implementation of certain network topologies, unnecessary pathways make its confusing, thus making use of host based methods to detect the spoofed packet at the router's level.

Templeton J.S & Levitt K.E. (2000) mentioned that IANA does control a certain number of IP addresses for special purposes. Table 1 does illustrate those special IP addresses and what they used for. Most firewalls will then compares that table with the packets they're handling. Nevertheless this method dose poses a constraint and can only be used only when IP packets are being thrown through. Yet an attacker cold still fake packets if on the same Ethernet subnet as both the IP and the MAC address would be spoofed. Therefore the need for other approaches is needed.

NON – ROUTING APPROACHES AND METHODS

ACTIVE DETECTION METHODS

With the active detection methods inquiries are performed to identify the originating source of the packet (reactive) or influence protocol specific guideline for the sender to take action upon (proactive). These different approaches do have an enormous advantage over the routing approaches and methods discussed earlier as there is no involvement of ISPs and prove to be impressive and operative even through the attacker may be sitting on the same subnet as the targeted host.

The active methods normally will depend on an acknowledgement from the claimed source but only if the spoofed host is operational that in can be influenced meaning that only when linked to the network gathering and handling packets. It is to mention that if a host is fully firewalled thus not replying to quests is then deduced to be inactive. However, these hosts are frequently manipulated as source addresses in spoofed packets. Therefore when hosts do not reply to any requests, passive methods is then used for acknowledgement and validation.

Time-To-Live (TTL) Approach

Since IP packets are normally routed across the network, their time-to-live (TTL) is subject to decrease. Therefore, the IP packet header can be controlled to make sure that IP packets are not being routed interminably when their host cannot be located in a fix number of hops according to Stevens W.R (1994). This technique is also used by some networked devices to halt IP packets from being transmitted outside a host's subnet. Moreover, Templeton J.S & Levitt K.E. (2000) mentioned that TTL is a useful approach and method in detecting spoofed packet but are based upon the following assumptions.

1. The number of hops will be the same then a packet is sent between two hosts provided the same routing path is taken
2. Packet transmitted near in time to each other will use the same trace route to its destination
3. Routes rarely changes
4. Whenever routes are altered, there is no compelling modification in the number of hops.

Having described the above assumptions, these mentioned approaches might outcome in false alerts if these assumptions are not respected. Hence repeatedly checking the packets should not breach those assumptions.

Direct TTL Requests:

With direct TTL, the mechanism of operation is quite simply by sending out a packet to a host that will have induce an acknowledgement, we can verify whether the TTL from the acknowledgement is identical to the one that was transmitted as it they are from the same protocol, they usually will have the same TTL thus considering the fact that different protocols would definitely use different basic TTL. However when different protocols are used then it is imperative that we deduce the true hop count. In the instance of TCP/UDP, 64 and 128 are the most commonly used whereas with ICMP, the initial value used are 128 and 255 respectively the number of hops can be calculated by subtracting the monitored TTL from the assumed value, we can then deduce the number of hops.

After all if an attacker does happen to have the same number of hops, this approach will return a false negative, but if the attacker was to be aware fo the exact number of hops between the faked host and target, it fake the TTL field

OS Fingerprinting

OS fingerprinting can be classified into 2 different categories, active fingerprinting which is associated with the direct probing of computer, and passive fingerprinting which refers to initially monitor the traffic and hence forth analyzing it to the various standardized benchmark for various operating system. On the other hand, we can create a controlled passive fingerprint as we monitor the traffic from a host and afterwards examine this against an active OS fingerprinting. Thus ascertaining if both OS are likely to be the same but if that is not the case, we then deduce that the packets are faked.

TCP Flow Control

In general a TCP header does have a send window size (SWS) which is the upper bound on the number outstanding frames that the sender can transmit (not the ACKed) or the maximum number or amount of data that the sender can receive. So, if even those SWS are to be set to zero, then the sender should stop transmitting data. But it is very important that the sender respond to the flow control by acknowledging the ACK-packets. Other wise the sender should terminate once the very first window size is over flowed else we could infer that the packets are spoofed. To make sure that this is not to occur, the sender can transmit a primary window size that is quite small thus if this margin is exceeded, the conclusion would be that the packets have been faked.

However, TCP packet spoofing is quite hard to achieve as the correct sequence number to a lot of TCP packets is required and most of the TCP connection don’t get pass the initial acknowledgement packet (ACK-PK) Therefore the best way for this to be effective is in the handshake process. The TCP handshake normally requests that the host transmitting initiate that first SYN wait for the SYN-ACK before transmitting the first ACK packet. Modifying the wonder size of the ACK-SYN to zero, would indicate whether the sender is accepting and reacting to the packets. Moreover, if the sender just return an ACK-PK certain that true originator is not addressing to our packets and hence a spoofed packet

TCP Approach and Methods

When it comes to detect spoofed TCP packets, a number of approaches and methods on top of the IP packet methods described later. The role of the TCP is to maintain reliable packet transmission which implies that both sender and receiver should be communicating. Thus allowing us to uncover the faked packet by masquerading the fact that the sender spoofed data packet will not be responsive to any packet from the receiver. Two different approaches combining ACK-PK will be used

1. Request the sender to delay sending packets
2. Request the sender to recovery a packet

IP Identification Number

As mentioned earlier in the Bounce Scanning, the sending host normally increase the identification number (IP) in the IP header through each packet send. Given that IP identification number can easily be altered,

thus making all the alterations calculable. Compared with TTL, IP ID can be used to detect spoofed IP packets despite the attacker sitting onto the same subnet as the target.

In a very simple way when we sent inquiring packets to the so said “claimed” source we expect to receive a reply, therefore the IDs should be very close to the previously received one from the host. Yet the packets should be slightly higher than the IDs’ in the controversial case. Therefore, we can infer that the packets sent out where not from the claimed source. It is also totally true that, if the host is bounded with the so called “source” is considerably active; the ID’s will change speedily.

With this approach, it is not unfamiliar to come across certain system that unfortunately changes the initiating IDs using a more complex methodology rather than incrementing by a constant number. To be in accordance with the RFC 791, Postel, J., (1981) mentioned that for fragmented packet assembly to be possible, only the ID numbers have to be in a successive order. Therefore this will favor more complex ID numbers. These can hence be overcome in 2 different ways. Firstly we could use a separate counter for every packet and secondly use a pseudo-random value which will have for aim to limit the actual IP data stream from interfering with each other. In those causes where more complex ID number is being used, using this particular approach might become ambiguous.

Retransmission of packets

In TCP, the use of sequence number is vital as it helps in determining which packets have already been acknowledged. Therefore, an ACK-PK normally informs the receiver of all the packets that has been send out together with the sequence number of the successfully received packet. When the packet is received the ACK-PK number is compared to the minimum and the maximum values and if less or greater than the required value, the packet is dropped thus allowing the connection to be resynchronized by sending out a packet with the minimum ACK-PK number. Moreover these replies can still be exploited to examine faked packets by probing a packet which has been spoofed from the internal host having an ACK-PK number higher than the required minimum value. We hence force a resynchronization ACK from the host being forged where if an RST reply is received, we can therefore deduced that the connection has been tampered with.

However, a major concern that arouse with this approach according to Joncheray, L (2001), it will conduce to an ACK-Storm since both ends-ie sender and receiver will struggle for resynchronization. Yet, their approach is better carried out on a firewall where the fake reply could be seized thus prevent an ACK-Storm as the interval host will not see the reply.

PASSIVE APPROACH

With the passive approach, we can hence say that monitored data will have a predicted value as we can learn what values are to be expected and hence separate the packets that don’t fit the expected norms. Since TTL values are dependable upon the hosting OS, the network topology, the packets protocols are fairly, static; therefore TTL can be opted as a core support for passive detection. Unfortunately this cannot be applied to IP identification Numbers which have a predominantly distinct association with packet and consequently eliminate the choice of using a passive approach.

Passive TTL approach

As previously discussed with TTL values are a very good way of identifying the different hops that lie between both the sender and the receiver’s and i.e. the source and the destination. Therefore, on performing a monitoring observation over a lap of time, we can learn the TTL values of specific IP addresses source and who deduce what would be their expected values at the host’s side. Most of times, if not nine out of ten times, we will come across packets that the host has never come across. Therefore, by comparing the IP addresses which are generally the same for the number of hops away.

Nonetheless, to be able to construct a better model for the detection of spoofed IP packets, both passive and reactive approaches should be used in conjunction where the reactive methods would be used for when certain packets seems dubious.

Since passive TTL approaches are very firm and reliable especially when it come network routing attacks which normally happened when packets destined for a host are routed to a different host impersonating the first host. Unfortunately, this is not entirely classified as packet spoofing due to the fact that these packets are still emanating from a valid IP address of the sender. In addition using passive TTL approach will definitely act as a routing detector

RELATED WORKINGS:

So far, there have not been a lot of work relations about detecting spoofed IP packets. Most of the works that have been published were addressing different spoofing attacks. The most common one ARP spoofing and according to Whalen, S (2001) associated with sending packet through Ethernet MAC of a different host than the IP address thus confusing local hosts of the local network to channel the packets to the wrong interface onto the network. In accordance to Aura, T. and Nikander, D. (1997), some firewalls make use of SYN-Cookies as an approach to reduce the effect of SYN-Flood type Dos. SO basically a SYN-cookie is a crypto-graphical ICP sequence number which is related to time, port number, and source IP address. The process is quite simple, if a SYN packet is received, instead of going around opening a buffer for the connection, the server will send a SYN-ACK-PK with the SYN_Cookie thus creating a stateless handshake. However, when an ACK-DK is acquired from an inactive socket, the returned sequence value is verified and compared if it is a valid SYN packet which was sent out from the host. Provided that the sequence number is valid, then only a buffer is allowed and hence begins the connection else the packet is dropped. This method is somehow used to mitigate SYN-flood attacks but not detect spoofed packets.

CONCLUSION:

Tracing IP packets have got a few limitations, such as tracing beyond corporate firewalls. To be able to effectively traceback the IP addresses, we need to be able to reach the host from where the attack where initiated and hence the difficulty of tracing back these IP packets through corporate firewalls and when trying to get the last IP address would show the firewall address.

However, detecting spoofed IP packets goes well beyond simple detection. Since faked packets are among the most common attacks, therefore identifying them at an earlier stage and hence preventing them in occurring will be of a major help to improve the network security. Though this paper we have shown a large range of techniques available in detecting spoofed packets. These various techniques and approaches can either be used on their own or could be combined to enhance detection effectiveness due to their ease of implementation. Yet we do understand that all the discussed methods above are not entirely complete as an attacker can still transmit spoofed packets which remain undetected. We also need to keep in mind that there is no such system which is fully –ie 100% reliable. There approaches are not the entire solution but they greatly contribute to increase the detection of spoofed IP packets.

REFERENCES:

Belenky, A & Ansari, N. (2003). *On IP traceback*. Retrieved September 7, 2007, from http://ieeexplore.ieee.org/iel5/35/27341/01215651.pdf

Bellovin, M.S. (2000). *ICMP Traceback Messages*. Retrieved September 3, 2007, from http://www.research.att.com/smb/papers/draftbellovin-itrace-00.txt.

CERT (1998). *Smurf IP denial-of-service attacks*. Retrieved September 10, 2007, from http://www.cert.org/advisories/CA-1998-01.html

CERT (1999). *Spoofed/Forged Email*. Retrieved September 7, 2007, from http://www.cert.org/tech_tips/email_spoofing.html

Chang, H., Narayan, R., Wu, S., et al.(1999). *DECIDUOUS: decentralized source identification for network-based intrusions*. Proc. of the Sixth IFIP/IEEE International Symposium on Integrated Network Management.

Chang, R.K. (2002). *Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial*. IEEE Communication Magazine.

Chang, H., Wu, S., and Jou, Y. (2001). *Real-Time Protocol Analysis for Detecting Link-State Routing Protocol Attacks*. ACM Transaction on Information and System Security (TISSEC).

Droms, R. (1997). *RFC 2131: Dynamic Host Configuration Protocol*. Retrieved August 28, 2007, from http://www.ietf.org/rfc/rfc2131

Joncheray, L. (1995). *A Simple Active Attack against TCP*. Retrieved August 15, 2007, from www.insecure.org/stf/iphijack.txt

Lau, F., Rubin, S.H., Smith, M.H., and Trajkovic, L. (2000). *Distributed denial of service attacks.* Proc. 2000 IEEE Int. Conf. on Systems, Man, and Cybernetics, Nashville, TN, pp. 2275-2280, October 2000.

Postel, J. (1981). *RFC 791: DARPA Internet Program Protocol Specification*. Retrieved August 23, 2007, from http://www.ietf.org/rfc/rfc791

Postel, J. (1981). *RFC793: Transmission Control Protocol.* Retrieved August 27, 2007 from http://www.ietf.org/rfc/rfc793.txt

Savage, S., Wetherall, D., Karlin, A., & Anderson, T. (2000). Practical Network Support for IP Traceback. Retrieved July 26, 2007, from http://www.cs.washington.edu/homes/savage/traceback.html

Stevens, R. (1994). *TCP/IP Illustrated*. Volume I – The Protocols. Addison-Wesley. 1st edition.

Templeton, S., and Levitt, K. (2000). *A Requires/Provides Model for Computer Attacks*. Retrieved September 17, 2007, from http://seclab.cs.ucdavis.edu/papers/DetectingSpoofed-DISCEX.pdf

Staniford, S., Hoagland, J., and McAlerney, J. (n.d.*). Practical Automated Detection of Stealthy Portscans*. Retrieved August 26, 2007, from http://www.silicondefense.com/pptntext/Spice-JCS

Snoeren, C.A. et al. (2002). *Single-Packet IP Traceback*. Retrieved August 31, 2007, from http://delivery.acm.org/10.1145/620000/611410/01134298.pdf?key1=611410&key2=2006031911&coll=GUIDE&dl=GUIDE&CFID=37231185&CFTOKEN=12343255

Song, D.X.,and Perrig, A. (2001). *Advanced and Authenticated Marking Schemes for IP Traceback.* Proceedings of INFOCOM.

Stone,R. (2000). *An IP Overlay Network for Tracking DoS Floods*," Proceedings of the 9th USENIX Sec. Symposium.

Whalen, S. (2001*). An Introduction to ARP Spoofing*. Retrieved August 11, 2007, from http://packetstorm.securify.com/papers/protocols/intro_to_arp_spoofing.pdf

Wu, S., Chang, H., et al.(1999). *Design and Implementation of a Scalable Intrusion Detection System for the OSPF Routing Protocol.* Journal of Computer Networks and ISDN Systems.

Zalewski, M. (2001). *Strange Attractors and TCP/IP Sequence Number Analysis*. Retrieved September 21,

2007, from http://razor.bindview.com/publish/papers/tcpseq.html

COPYRIGHT