IJSEAT International Journal of Science Engineering and Advance Technology

Spoofing Attacks on Packets And Methods For Detection And Prevention Of Spoofed Packets

K. Phalguna Rao[1] , Prof. Ashish B.Sasankar, *Dr. Vinay Chavan*[3]

[1] PhD Scholar, Dept of Dept. of Computer Science, Nagpur University, M.S, India.

, Head and Associate Professor of the Dept. of Computer Science, G.H. Raisoni Institute of information Technology, Nagpur,M.S, India.

[3] Head and Associate Professor of the Dept. of Computer Science, S.K.Porwal College, Kampat, Nagpur MS, India

Abstract—**In a spoofing attack, the attacker creates misleading context in order to trick the victim into making an inappropriate security-relevant decision. A spoofing attack is like a con game: the attacker sets up a false but convincing world around the victim. The victim does something that would be appropriate if the false world were real. Unfortunately, activities that seem reasonable in the false world may have disastrous effects in the real world. Spoofing attacks are possible in the physical world as well as the electronic one. For example, there have been several incidents in which criminals set up bogus automated-teller machines (ATM), typically in the public areas of shopping malls. The machines would accept ATM cards and ask the person to enter their PIN code. Once the machine had the victim's PIN, it could either eat the card or "malfunction" and return the card. In either case, the criminals had enough information to copy the victim's card and use the duplicate. In these attacks, people were fooled by the context they saw: the location of the machines, their size and weight, the way they were decorated, and the appearance of their electronic displays. In this Research we discus spoofing attacks and detection methods of spoofing attacks.**

Keywords: spoofing, security, decision, malfunctions.

I. INTRODUCTION

Packets sent using the IP protocol [15] include the IP address of the sending host. The recipient directs replies to the sender using this source address. However, the correctness of this address is not verified by the protocol. The IP protocol specifies no method for validating the authenticity of the packet's source. This implies that an attacker could forge the source address to be any he desires. This is a well-known problem and has been well described [5][10][12]. In all but a few rare cases, sending spoofed packets is done for illegitimate purposes. Sending IP packets with forged source addresses is known as packet spoofing and is used by attackers for several purposes. These include obscuring the true source of the attack, implicating another site as the attack origin, pretending to be a trusted host, hijacking or intercepting network traffic, or causing replies to target another system.

Because none of these are desirable, it is useful to determine if a packet has a spoofed source address. In cases where an ongoing attack is occurring it is beneficial to determine if the attack is from a particular location. In many cases we are able to determine when packets are spoofed, and generally from where they originate. Spoofing of network traffic can occur at many layers. Examples include network layer spoofing (e.g. Ethernet MAC spoofing), non-IP transport layer spoofing (e.g. IPX, NetBEUI), as well as session and application layer spoofing (e.g. email spoofing). All of these have significant security concerns. However, for the purposes of this paper we will focus only IP packet spoofing. A related issue is attacks that cause packets to be routed to a different host than the sender intends. These are attacks on routing [9] and the DNS system [4]. Packet spoofing is restricted to false source addresses in the IP packet header. This paper discusses a variety of methods that can help determine if received packets have spoofed source addresses. Routing-based methods rely on routers and other network devices to identify traffic with unexpected source addresses or can aid spoofed packet detection. Non routing methods include both active and passive techniques a host can use to determine if a received packet is spoofed. Active methods involve either probe, such that the response will corroborate the authenticity of a received packet, or methods that cause changes in network behavior such that the observed change (or lack of change) can corroborate the authenticity of the packets. Passive methods involve observing packet data that would be anomalous in legitimate packets. These methods are not intended to function in isolation, rather to provide supplemental information to other IDS components or to help assess the significance of far too common nuisance alerts generated by commercial IDSs. Spoofed packet detection is an example of techniques to provide supplemental information to corroborate other IDS reports when needed.

II. . Packet Spoofing Attacks

Because packet spoofing can be part of many different types of attacks, it is important to have an understanding of how they are used. A key factor in all packet-spoofing attacks is that it is not necessary for the attacker to directly receive packet replies from the target. Replies are either unimportant, their contents can be inferred, or the packets can be observed in transit. This section describes several such attacks and discusses their security implications.

Man–in-the-middle: packet sniffs on link between the two endpoints, and can pretend to be one end of the connection

Routing re-direct: redirects routing information from the original host to the hacker's host (a variation on the man-in the-middle attack)

Source routing: redirects individual packets by the hacker's host Blind spoofing: predicts responses from a host, allowing commands to be sent, but does not get immediate feedback

Flooding; SYN flood fills up the receive queue from random source addresses; smurf/fraggle spoofs victims address, causing everyone to respond to the victim.

A. **SYN-flood**

A SYN flood works by establishing half-open connections to a node. When the target receives a SYN packet to an open port, the target will respond with a SYN-ACK and try to establish a connection. However, during a SYN flood, the three-way handshake never completes because the client never responds to the server's SYN-ACK. As a result, these "connections" remain in the half-open state until they time out. Imagine this process occurring several thousand times per second. Soon, the target server will run out of memory/resources, or cause a system crash. Additionally, any stateful devices in the path between the attacker and target will also be overwhelmed with connection requests, possibly filling up the session table on those devices if the SYN flood is not dealt with effectively. Because SYN packets are normal and necessary for TCP communication, a system cannot simply drop all SYN packets as in the case of a "Ping of Death" DoS attack, for example. SYN floods can be mitigated effectively up to a certain point using a SYN proxy feature in a stateful firewall. Above this rate, a stateless screening router can be used to further limit TCP-SYNs.

SYN Flood Protection: TCP SYN cookies SYN cookies as a reaction to an attack

SYN cookies are a particular choice of the initial sequence number.

The server generates the initial sequence number α such as:

α= h (SSYN, DSYN, K)

SSYN: src address of the SYN packet

DSYN: address of the server

K: a secret key is a cryptographic hash function.

At arrival of the ACK message, the server calculates again. Then, it verifies if the *ack number* is correct. If yes, it assumes that the client has sent a SYN message recently (considered as normal behavior), and allocates memory.

B. SMURF

In a SMURF attack you can be affected in one of several ways:

- As a victim or target of the attack
- As a network which is abused to amplify the attack
- As a party harboring the instigator of the attack

SMURF and similar Denial-of-service (DoS) attacks can do serious damage to your network services, be it either as an individual end-user or as an entire institution in that your network or host can be inundated with unwanted and maliciously sent traffic.

A SMURF attack (named after the program used to perform the attack) is a method by which an attacker can send a moderate amount of traffic and cause a virtual explosion of traffic at the intended target. The method used is as follows:

- The attacker sends ICMP Echo Request packets where the source IP address has been forged to be that of the target of the attack.
- The attacker sends these ICMP datagram's to addresses of remote LANs broadcast addresses, using so-called directed broadcast addresses. These datagram's are thus broadcast out on the LANs by the connected router.
- All the hosts which are «alive» on the LAN each pick up a copy of the ICMP Echo Request datagram (as they should), and sends an ICMP Echo Reply datagram back to what they *think* is the source. If many hosts are «alive» on the LAN, the amplification factor can be considerably (100+ is not uncommon).

- The attacker can use largish packets (typically up to Ethernet maximum) to increase the «effectiveness» of the attack, and the faster network connection the attacker has, the more damage he can inflict on the target and the target's network.

C . Preventing SMURF attacks

The availability of the directed broadcast function is an important element in these attacks. The current Proposed Standard for "Requirements for IP Version 4 Routers" states that a router must default to forwarding directed broadcasts that a knob must exist to turn it off, but it must default to the «on» position. However, the current sentiment is that this should no longer be a requirement. Thus, to prevent your network from being abused as an amplifier network in a SMURF attack, you should turn off the forwarding of directed broadcast on all router ports or take other measures to assure your network cannot be abused in this manner.

Another component which is important in this type of attack is that the attacker has to be able to inject packets into the network with forged IP source addresses. It is possible to enable functions in routers which will prevent the trivial forgery of IP source addresses, and doing so for a local network will prevent SMURF attacks from being launched locally. (Do however note that access lists *can* have a performance impact, so judicious use of such tools is advised.) This sort of ingress filtering has been documented in RFC2267, and is effective not only for preventing local origination of SMURF attacks, and also makes tracking attacks (or denying origination of attacks) much easier.

Since SMURF attacks use forged source addresses, tracking SMURF attacks back to their source can be a challenge. It has to be done while the attack is ongoing, and requires the swift cooperation of all the network service providers along the path. In practice this has proven to be quite difficult.

D. TCP Connection Spoofing

This attack requires coordination of several attacks; primarily denial-of-service of a trusted host, and packet spoofing of the attack target. The DoS component can be anything that prevents the trusted host from sending reset packets to the target. One such means would be a SYNflood. The other component requires sending packets spoofed to be from the trusted host to the target. Because of the DoS attack, the trusted host cannot reply to packets received from the target, and the attacker can cause the target to believe the packets are from the trusted host. This will allow the attacker to use the target as if it were the trusted host. This attack is made difficult because TCP requires reply packets to include the sequence number of the preceding packet. If the attacker cannot directly observe the packets, it must guess the sequence numbers. RFC 1948[6] provides recommendations for increasing the difficulty of predicting sequence numbers. Theoretically sequence numbers could be made un guessable. However, while more difficult than in the past, it is still possible and not as difficult as is widely believed [12].

A common misconception is that "IP Spoofing" can be used to hide your IP address while surfing the Internet, chatting on-line, sending e-mail, and so forth. This is generally not true. Forging the source IP address causes the responses to be misdirected, meaning you cannot create a normal network connection. However, IP spoofing is an integral part of many networks that do not need to see responses.

III. 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. If you monitor packets using network-monitoring software such as netlog, look for a packet on your external interface that has both its source and destination IP addresses in your local domain. If you find one, you are currently under attack. Another way to detect IP spoofing is to compare the process accounting logs between systems on your internal network. If the IP spoofing attack has succeeded on one of your systems, you may get a log entry on the victim machine showing a remote access; on the apparent source machine, there will be no corresponding entry for initiating that remote access.
Source Address Validation:

- Check the source IP address of IP packets
  - filter invalid source address
  - filter close to the packets origin as possible
  - filter precisely as possible
- If no networks allow IP spoofing, we can eliminate these kinds of attacks.
- We can check and drop the packets which have unused address everywhere, but used space can be checked before aggregation.

A. ***Prevention*** *IP spoofing*

The best method of preventing the IP spoofing problem is to install a filtering router that restricts the input to your external interface (known as an input filter) by not allowing a packet through if it has a source address from your internal network. In addition, you should filter outgoing packets that have a source address different from your internal network in order to prevent a source IP spoofing attack originating from your site. If your vendor's router does not support filtering on the inbound side of the interface or if there will be a delay in incorporating the feature into your system, you may filter the spoofed IP packets by using a second router between your external interface and your outside connection. Configure this router to block, on the outgoing interface connected to your original router, all packets that have a source address in your internal network. To prevent IP spoofing happen in your network, the following are some common practices:

1- Avoid using the source address authentication. Implement cryptographic authentication system-wide.

2- Configuring your network to reject packets from the Net that claim to originate from a local address.

3- Implementing ingress and egress filtering on the border routers and implement an ACL (access control list) that blocks private IP addresses on your downstream interface. If you allow outside connections from trusted hosts, enable encryption sessions at the router.

B . 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 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 method[17].

There are several IP addresses that are special in one way or another. These addresses are for special purposes or are to be put to special use.

- Addresses significant to every IP subnet
  - Network Address
  - Broadcast Address
- Addresses significant to individual hosts
  - Loop back Address
- Special Addresses of Global Significance
  - Private Addresses
  - Reserved Addresses

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.

Spoofed packet detection can be implemented as either an IDS sensor or as a firewall process. As a sensor, packets believed to have spoofed source addresses will generate alerts for use by the IDS. Used in a firewall, the packets can be dropped or passed but flagged as possibly spoofed. Security monitoring systems could use this in detecting attacks. A robust and efficient spoofed packet detection process should use a combination of methods to make its determinations. The system we are constructing first determines if the packet is suspicious using passive techniques then active probes to corroborate the passive detectors prediction. A number of different probes are used to determine if the packet was spoofed. If the probes indicate the packet is not spoofed, the system will update the static classifier to include the new values. Although updating the detector could incorporate router attacks as valid packet sources, it also allows us to learn more of the existing network relations. While it is possible to check all packets, for efficiency we see these methods being most useful as an on-demand adjunct to primary IDS.

IV. FIGURES AND TABLES

TABLE I. Table of Words

Address	Used for	Reference
10.0.0.0	Private-Use Networks	RFC 1918
14.0.0.0	Public-Data Network	RFC1700
127.0.0.0	Loopback address	RFC1700
169.254.0.0	Link Local	RFC 3927
224.0.0.0	Multicast	RFC3171

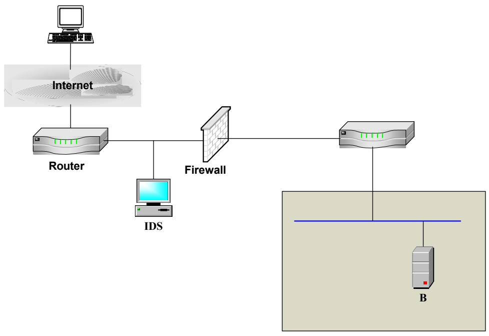

Fig. 1. Packet moving in the internet.

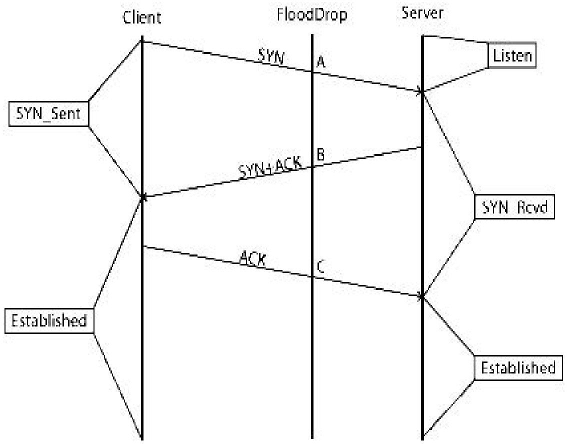

Fig2. Establishing half-open connections to a node.

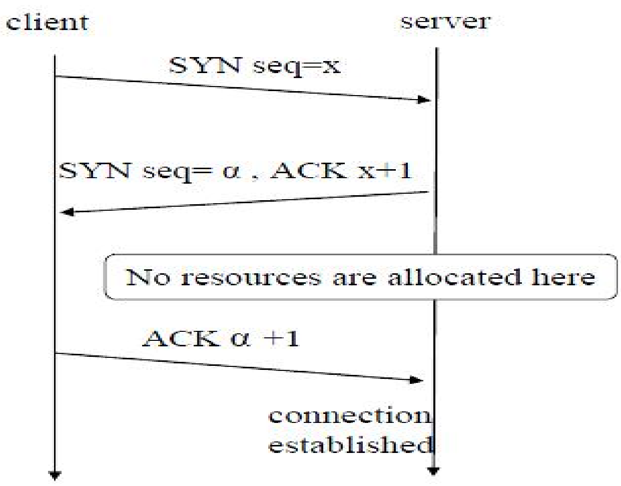

Fig3. Generates the initial sequence number

Conclusion

The original motivation for this research was our work in model based intrusion detection [29]. At issue was a lack of sensors to provide needed information to support correlation of events. Generally, these sensors required inferring something not directly in an observed packet. Some examples include, are attack packets are from the same attacker, was an attack successful, is a sniffer present, and if a packet was spoofed. We quickly found that such sensors are possible and could be used to support IDS. Furthermore, while investigating commercial IDSs, we observed that many of the alerts generated were false positives and could be eliminated if corroborating information were available. The ability to know if the packets that generated the alerts were spoofed is just one example of supplemental information that would help in filtering out those alerts of low significance. The utility of detecting spoofed packets extends beyond simple detection and assessment. When used at a firewall to detect and block spoofed packets, the discussed techniques can be used to prevent spoofed packet attacks.

ACKNOWLEDGMENT

We thank all the anonymous reviewers for their helpful comments which helped us to improve the quality and presentation of this paper. We are also grateful to the Associate Editor whose pertinent remarks also contributed toward further improvements of the paper.

REFERENCES

[1] T. Aura and P. Nikander. Stateless connections. Proc.International Conference on Information

and Communications Security (ICICS'97), Beijing, China, 1997.

[2] S. Bellovin. Using the Domain Name System for System Break-ins. Proc. of the 5th UNIX Security Symposium, pp.199-208, June 1995.

[3] S. Bellovin. Security Problems in the TCP/IP Protocol Suite. Computer Communications Review, vol. 19, no. 2, pp. 32-48, April 1989.

[4] H. Chang, R. Narayan, S. Wu, B. Vetter, X. Wang, M. Brown, J. Yuill, C. Sargor, F. Jou, and F. Gong. DECIDUOUS: decentralized source identification for network-based intrusions. Proc. of the Sixth IFIP/IEEE International Symposium on Integrated Network Management May 1999.

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

[6] L. T. Heberlein and M. Bishop. Attack Class: Address Spoofing. Proc. of the 19th National Information Systems Security Conference, pages 371-377, October 1996.

[7] L. Joncheray. A Simple Active Attack Against TCP. Proc. Fifth Usenix UNIX Security Symposium, 1995.

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

[9] V. Paxson. End-to-end Routing Behavior in the Internet. to appear in Proc. SIGCOMM '96, August 1996.

[10] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for IP traceback. Proc. of the 2000 ACM SIGCOMM Conference, August 2000.

[11] C. Schuba and E. Spafford. Countering abuse of name-based authentication. Proc. 22nd Annual Telecommunications Policy Research Conference, 1996.

[12] D. Schnackenberg, K. Djahandari., and D. Sterne. Infrastructure for Intrusion Detection and Response. Proc. of the DARPA Information Survivability Conference and Exposition (DISCEX '00), 2000.

[13] S. Staniford-Chen and L. T. Heberlein. Holding Intruders Accountable on the Internet. Proc. of the 1995 IEEE Symposium on Security and Privacy, Oakland, CA, pages 39-49, May 1995.

[14] S. Templeton and K. Levitt. A Requires/Provides Model for Computer Attacks. Proc. of the New Security Paradigms Workshop 2000, Cork Ireland, September 2000.

[15] J. Postel. RFC 791: DARPA Internet Program Protocol Specification. http://www.ietf.org/rfc/rfc791, September 1981.

[16] M. Zalewski. Strange Attractors and TCP/IP Sequence Number Analysis. http://razor.bindview.com/publish/papers/tcpseq.html,

[17] S. Templeton and K. Levitt. A Requires/Provides Model for Computer Attacks. Proc. of the New Security Paradigms Workshop 2000, Cork Ireland, September 2000.

[18]. G. Kanwal, & Rshma, C. , "Detection of DDoS Attacks Using Data Mining," *International Journal of Computing and Business Research (IJCBR),* vol. 2, pp. 1-10., 2011.