International Journal of Networks and Communications 2012, 2(3): 33-37
DOI: 10.5923/j.ijnc.20120203.03

Block Spoofed Packets at Source (BSPS): A method for Detecting and Preventing All Types of Spoofed Source IP Packets and SYN Flooding Packets at Source: A Theoretical Framework

Noureldien A. Noureldien, Mashair O. Hussein*

Department of Computer Science, University of Science and Technology, Omdurman, Sudan

Abstract In this paper, we present a theoretical framework for a simple and efficient method that detects and blocks source IP spoofed packets and TCP/SYN flooding packets at source. The method is based on a network authentication server (AS), which performs an authentication process on SYN packets. The authentication process verifies the legitimacy of SYN packet's source IP address that initiate a connection request from a network subnet host to an external host. During the authentication process of SYN packets, AS identifies and blocks SYN packets with legal source IP address that chip in a TCP/SYN flooding attack. AS preserves network performance by exchanging authentication messages in plain text, and acts as a stateful inspection firewall and only SYN packets are subject for inspection. Our method which is capable to detect and prevent all types of spoofing packets including subnet spoofing contributes to standard ingress/egress methods in eliminating bogus traffic on the Internet.

Keywords IP Spoofing, SYN Flooding, Authentication

1. Introduction

On TCP/IP networks, packets sent from one host to another include an IP header that contains two fields: Source IP address and Destination IP address. The source IP address identifies the sending host and destination IP address identifies the receiving host. The recipient host directs replies to the sender using this Source IP address. However, the IP at the recipient has no means to validate the authenticity of the packet's source address. This vulnerability can be exploited by attackers to send packets with forged or spoofed source IP address. Sending IP packets with forged source addresses is known as packet spoofing or source IP spoofing. Source IP spoofing is well-known TCP/IP vulnerability and has been well described in[11][4][5].

Attackers can spoof different types of IP addresses to use as source IP in their spoofed-packets. These spoofed source IP addresses can either be a random IP addresses; subnet IP address, En Route IP addresses, or Fixed Spoofed IP addresses[7]. In Random Spoofed Source Address attackers spoofed random Source IP address in attack packets. This is simply achieved by generating random 32-bit numbers and stamping packets with them. An En Route Spoofed Source Address attack would spoof the address of a machine or subnet that lies along the path from the agent machine to the victim. There have not been any known instances of attacks that use en route spoofing technique could affect route-based filtering.

In Fixed Spoofed Source Address, the attacker performs a reflector attack to place a blame for the attack on several specific machines. His spoofed packets carry a source address chosen from a fixed given list of IP addresses. Fixed spoofing is counted by the same random spoofing techniques. In Subnet Spoofed Source Address, the attacker spoofs a random address from the address space assigned to the attacker machine's subnet. For example, a machine which is part of 130.170.192.0/24/ network could spoof any address in the range 130.170.192.0 – 130.170.192.255.

Many different types of attacks can be classified as packet-spoofing attacks. Attackers relay on IP spoofing either to make the attack possible, such as DoS/DDoS flooding attacks, and/or to obscure the attack source to complicate tracing back methods. A key factor in most of these packet-spoofing attacks is that, they overwhelm the target with floods of spoofed packets and render it to an unresponsiveness state. Such attacks do not receive packet replies from the target. Replies are either unimportant, their contents can be inferred, or the packet can be observed in transit[12].

* Corresponding author:
mashair_omer@yahoo.com (Mashair O. Hussein)
Published online at http://journal.sapub.org/ijnc

In this paper we present a simple and efficient method for early detection of spoofed packets, which we call, the Block Spoofed Packets at Source (BSPS) method. The method is based on an authentication server (AS) that performs a non-cryptographic authentication process to authenticate the source IP address in IP headers of SYN packets. The AS was intended to be placed ahead of the edge router in the network architecture so as to accept subnet connection requests before arriving to the network router. AS extracts the SYN's source IP, destination IP and MAC address and performs checks on the legitimacy of this information using two tables, a static Host Information Table (HIT), which stores information about subnet hosts and a dynamic Connection State Table (CST) that stores information about the currently connected hosts. On successful primary checks AS requests the host initiating SYN to send its MAC address, which will be used in a final verification check. The rest of the paper is organized as follows. In section 2, we discuss the current proposed spoofing prevention methods. In section 3, we give a detailed description of our proposed BSPS method. In section 4, we present our ideas for future work. Finally, conclusions are drawn in section 5.

2. Spoofing Prevention Methods

A variety of methods that help in determining whether a received packet have spoofed source IP address have been proposed. These include routing-based methods that rely on routers and other network devices to identify spoofed packets, and non-routing methods which apply both active and passive techniques a host can use to determine if a received packet is spoofed[12]. Routing based methods are based on the fact that, routers and firewalls can easily distinguish between incoming (inbound) packets and outgoing (outbound) packets, since they deploy different network interface, which make it possible for them to identify packets that should not have been received by a particular interface.

Ingress filtering[10] is based on the internal capability of an edge router or a gateway to identify internal IP addresses from external IP addresses. So if a router receives IP packets with external IP addresses on an internal filtering is to block such packets. Egress filtering is archetypal to ingress filtering. If a router or a gateway receives IP packets with an internal IP addresses on an external IP interface, then this is a spoofed packet and should be blocked. The major drawbacks for ingress and Egress filtering are that; their capabilities in combating IP spoofing is limited and they entail a global deployment.

In the last few years, new methods that based on determining validity of a packet's source IP address while it is in the path routers in the Internet are proposed. StackPI[1] is a router based proposed solution in which routers on the packet's path towards its destination are assumed to deterministically mark bits in the packet's IP Identification field. The deterministic marking are assumed to guarantee that packets traveling along the same path will have the same marking. Another yet new marking approach is MASK[9].

Source Address Validity Enforcement Protocol (SAVE) [8], employ a mechanism to construct router table that map ranges of IP address to the proper incoming interface for packets with those source address. It proposed that routers would learn this information on the fly, and maintain an incoming routing table, where the valid incoming interface, for a given source address would be stored.

TTL field in the IP header is used to play a role in filtering spoofed packets as in[6][3]. Filtering in[6],[3] is done differently, but they share the same concept that packets originating at the same source traverse the same route. The Spoofing Prevention Method[2], proposed the use of a filtering key that is a function of the Source IP and IP destination pair and it is filtering technique is independent to the route the packets traverse. With exception to ingress/egress filtering which prevent spoofing at source, all of the above mentioned proposed solutions are destination-based solutions that detect spoofing at destination to prevent victims from spoofed attacks. Our contribution in this paper is to improve filtering capabilities at source by detecting and blocking all types of source IP spoofing packets including subnet spoofing and TCP/SYN flooding attack packets.

3. Block Spoofed Packets at Source

The BSPS method was designed to detect and block all types of spoofed packets including subnet spoofing packets and TCP/SYN flooding attack packets at source before reaching the network exit router.

The basic design characteristics that have been considered in designing BSPS are simplicity and efficiency. To realize these characteristics, BSPS was designed to constitute a few uncomplicated databases and routines, and to use techniques that lessen network overhead.

The BSPS design was centered on an Authentication Server (AS), which has to authenticate connections between subnet hosts and external hosts to ensure the authenticity of the source IP addresses, and as part of this authentication process AS has to detect subnet hosts that participate in a TCP/SYN flooding attack. To perform these functionalities, AS acts as a connection service provider and receives all SYN packets that initiated by subnet hosts prior reaching the network gateway.

In receiving a SYN packet, AS employ a three-way authentication process with the initiating host. This authentication process requires specific design considerations on both, server side (AS) and client side (subnet host).

On the server side, the design of AS comprises a simple database that was made-up of two tables, and three routines that act on these tables.

The first table is a static Host Information Table (HIT), which stores the subnet hosts assigned IP addresses and their corresponding MAC addresses, and the second table is a dynamic Connection State Table (CST), which stores the IP source, IP destination and destination port for currently ac-

tive connections.

The first routine, Check_SYN(), is used to capture SYN packets and to verify, using CST, that a given packet is not part of a SYN flooding attack. Also Check_SYN() is responsible for updating the CST. The Primary_Auth_Check() routine is invoked by Check_SYN() and checks for external spoofing, and IP source subnet spoofing by validating the SYN IP Source, and MAC address respectively against HIT entries. On successful matching the routine invokes the third routine, Final_Auth_Check() for a final verification.

The Final_Auth_Check() verifies the authenticity of captured SYN's MAC address by requesting the source host to send its MAC address within a constrained time limit. On timely response and successful matching with the captured MAC address, the captured SYN packet is considered genuine, and CST is updated via Check_SYN() routine . Otherwise a comprehensive attack is detected and blocked where both IP source and MAC address are spoofed.

On the client side, only a simple Send_MAC() routine is required. Send_MAC monitor the client connection requests to AS and responds to the expected MAC address request from AS.

To lessen the network overhead during the authentication process, AS captures only SYN packets, thus it acts as a stateful inspection firewall, in addition, messages are exchanged as plain texts between the AS and clients, thus it eradicates processing overhead caused by encryption.

In the following subsections we will discuss in details each of the above mentioned BSPS design components.

3.1. Host Information Table (HIT)

The Host Information Table (HIT) was used to store information about hosts in the network. Each host has a record in HIT consisting of two fields, namely, the host source IP, and the host MAC address.

For each host, the MAC address field is filled up manually. The host source IP address field is filled either manually, in case of static IP address assignment, or automatically in case of dynamic IP address assignment via a DHCP. The HIT is used by the Primary_Auth_Check() to perform an initial authentication check on the received SYN packet.

3.2. Connection State Table (CST)

The Connection State Table (CST) is a dynamic table that stores information about the current active connections. Each connection is represented by an entry containing three fields: the source IP address, destination IP address and destination port number. Hereafter, we will refer to this triple as connection signature.

Entries are added and removed to CST by Check_SYN() routine, which uses CST in monitoring ongoing connections between subnet hosts and external ones in order to detect TCP/SYN flooding packets launched by subnet hosts.

3.3 The Check_SYN() Routine

Check_SYN() routine is used to capture clients connection requests, that is, the SYN packets, and to check these packets to determine whether they belongs to a TCP/SYN flooding attack. Here is how Check_SYN() works.

1- It captures the SYN packet, and extracts its connection signature.

2- It looks up the CST for the SYN connection signature.

3- If it finds the signature, then it recognizes the packet as a SYN flooding packet and it block and log the packet.

4- Otherwise, it passes the SYN's source IP address and MAC address to the Primary_Auth_Check() routine and start a timer

5- On receiving an acknowledgment from Final_ Auth_Check(), it adds the SYN connection signature to the CST table.

6- On seeing a FIN packet, it removes the corresponding entry from CST. Figure (1) shows a simplified description for Check_SYN().

```
/* Assume AS receives a SYN packet */
Begin
{
Extracts SYN connection signature;
if (SYN(Source IP, Destination IP, Destination Port Num-
ber) ! in CST)
{
   /* call the primary-check function */
    Primary_Auth_Check (SYN(source IP, MAC
    address));
    Start a timer;
}
else
{
     /* SYN flooding attack s detected */
         Block SYN;
         Log SYN;
}
If (an acknowledgment received in time)
{
   Adds SYN's connection signature to CST;
}
If FIN packet received
{
  Removes SYN's connection signature from CST;
}
End;
```

Figure 1. A Simplified Description for Check_SYN()

3.4. The Primary_Auth_Check() Routine

The Primary_Auth_Check() routine performs the following tasks:

1- It searches HIS records for the SYN(source IP).

2- If search fails, then this indicates and external spoofing, so it blocks and logs the packet and terminates the authentication session.

3- If the search is succeeding, then it compares the SYN(MAC address) with the MAC value stored in the HIS record that match SYN(source IP).

4- If SYN(MAC address) is not identical to the stored MAC value, then this indicates a subnet spoofing where the

attacker spoofed only the source IP. So it blocks and logs the SYN packet and terminates the authentication session.

5- If SYN(MAC address) is identical to the stored MAC value, it passes the SYN(source IP), and SYN(MAC address) to the Final_Auth_Check() routine. Figure (2) shows a simplified description for how Primary_Auth_Check() routine works.

```
/* Assume Primary_Auth_Check() received SYN(source IP,
MAC address) */
/* Assume the maximum size of HIT is n */
Begin
/* Search for SYN(source IP) in HIT */
Found = false;
for   i = 1 to n do
if (SYN(Source IP) =   HIT[record_i (source IP)])
      {
            Found = true;
            Break;
      }
/* check exit condition */
if   ! Found
{
        /* External Spoofing is Detected */
       Block SYN;
       Log SYN ;
}
else
         if (SYN(MAC) !=   HIT[record_i (MAC)])
          {
      /*Subnet Spoofing is detected*/
      /* Attacker Spoofed Source IP */
      Block SYN ;
      Log SYN ;
        }
    else
       /* call the Final_Auth_Check */
       Final_Auth_Check(SYN(sourceIP,MAC
       address))
End.
```

Figure 2. A simplified description for how the Primary_Auth_Check() Routine works

3.5. The Final_Auth_Check() Routine

On receiving the SYN's source IP and MAC address, the Final_Auth_Check() routine works as follows:

1- It uses the SYN's source IP to requests the host to send its MAC address, and starts a timer.

2- If the timer expires before a response is received (this happens when the host initially does not request a connection service from the AS), then a subnet spoofing is detected, where the attacker spoofed both the source IP and the MAC address of another subnet host, so it blocks and logs the SYN packet and terminates the authentication session.

3- Otherwise, if a response is received in time with the correct MAC, then this indicates the authenticity of the connection, so it forward the SYN packet, and acknowledges Check_SYN() to add the SYN connection signature to the CST. Figure (3) shows a simplified description for how Final_Auth_Check routine works.

```
/*   Assume   Final_Auth_Check()   received
SYN(Source IP and MAC address) */
Begin
Send a request to the SYN's source IP host to send its MAC
address;
Start a timer;
If (a response is received before timer expires)
{
     If (received MAC = SYN(MAC)
     {
/*SYN is authentic*/;
          Forward SYN;
          /* Acknowledge Check_SYN() */
     Check_SYN(OK);
}
else
/*Subnet spoofing  is detected;*/
/* Attacker spoofed both source IP and MAC address */
Block SYN;
Log SYN;
End.
```

Figure 3. A Simplified Description for How Final_Auth_Check() Routine Works

3.6. The Send_MAC() Routine

The Send_MAC() routine runs on the client side. The basic task of this routine is to monitor the client requests to AS for connection, and to respond to the expected request from AS that requesting host MAC address.

On sending a SYN packet from the host to AS the routine performs the following:

1- It starts a timer once a SYN packet is sent, and waits for a response from AS. The role of the timer here is to evade the host from responding at any time to AS requests. So Send_MAC() will respond to AS requests only when the host was previously requesting AS a connection service and the AS request arrives within the time limit.

2-If a response is received within the timer limit, it sends the host MAC to AS and terminates. Figure (4) shows a simplified description for how Send_MAC() works.

```
/* Assume the host sends a SYN */
Begin
    Start a timer;
    If (AS response is received before timer expires)
        Send host MAC;
End.
```

Figure 4. A simplified Description for How Send_MAC() Routine Works

4. Future Work

Our future work is to implement and to test BSPS in an experimental network environment. We plan to use Linux as a network development and testing environment.

Also we have to decide on the essential software needed to perform the testing experiments, such as packets generating tools on client side to generate SYN packets and packet

capturing tools on the server side to capture SYN packets.

5. Conclusions

This paper proposes a new approach to prevent IP spoofing at source using a simple non-cryptographic authentication method. The method overcomes ingress limitation in preventing subnet spoofing, and provides an efficient means for detecting and blocking TCP/SYN flooding packets.

Similar to Ingress/Egress filtering, our developed method necessitates an international deployment by network administrators and ISP's to alleviate bogus traffic in the Internet.

REFERENCES

[1] A. Yaar, A. Perrig, and D. Song, "Pi: A path identification mechanism to defend against ddos attacks," in Proc .IEEE Symposium on Security and Privacy, 2003.

[2] Bremler-Barr, A. Levy, H, "Spoofing Prevention Method" , in proceedings of 24th Annual Joint Conference of IEEE computer and communications societies, INFOCOM 2005, V1, Pages 536-547.

[3] C. Jin, H. Wang, and K. Shin, "Hop-count filtering: An effective defense against spoofed ddos traffic," in Proc. ACM Conference on Computer and Communications Security, 2003.

[4] Computer Incident Advisory Committee (CIAC). Advisory Notice F-08 "Internet Spoofing and Hijacked Session Attacks". USES 1995.

[5] Daemon9"IP Spoofing Demystified". Phrack Magazine Review, Vol 7, No. 48, 48-14, June 1996.

[6] G. Pazi, A. Bremler-Barr, R. Rivlin, and D. Touitou, "Protecting against distributed denial of service attacks," 2002, Patent Application 20030110274.

[7] Jelena MirKovic, Peter Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms"ACM SIGCOMM Computer Communications review, Volume 34, Number 2, April 2004,pp.39-54

[8] J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang, "Save: Source address validity enforcement protocol," in Proc. INFOCOM 2002, 2002.

[9] LU XiCheng, LÜ GaoFeng, ZHU PeiDong & CHEN YiJiao, "MASK: An efficient mechanism to extend inter-domain IP spoofing prevention". Springer, Science in China series, 2008.

[10] P. Ferguson and D. Senie."Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing", RFC2827.May 2000.

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

[12] Steven J. Templeton and Karl E. Levitt, "Detecting Spoofed Packets". In proceedings of DISCEX03, 2003.