International Journal of Multidisciplinary Approach and Studies

ISSN NO:: 2348 – 537X

Defence against SYN – Flooding Attack

R.Saranya[*] & N.Prathap[**]

[*]*Assistant Professor, Department of CSE , Pannai College of Engineering and Technology Sivagangai, Tamilnadu, India*

[**]*Assistant Professor, Department of ECE, J.J. College of Engineering and Technology Trichy, Tamilnadu, India*

ABSTRACT

This paper describes the new dimension of Distributed Denial-of-Service (DDoS) attack called TCP SYN flood attack, which exist in the society for several years. The attack exploits the implementation characteristics of Transmission Control Protocol (TCP), which amplify the severity of damage and can be used to make the server processes incapable of responding to legitimate clients requests for a new TCP connection. In this attack, illegitimate user sends a succession of SYN requests to a target's system in attempt to consume enough server resources to make the system unresponsive to the legitimate traffic. Various mitigations against these attacks and trade off each are described. This article archives the explanations of the attack, why it works, and follows with an overview and assessment of the current tactics that are used in both end hosts and network devices to combat SYN flooding attacks for the benefit of TCP implementers and administrators of TCP servers or networks.

Keywords

Three way handshake, flooding,snoofing, caches, TCP SYN, FIN,cookies

I. INTRODUCTION

Internet has transformed and greatly improved the way to business, this vast network and its associated technologies have opened the door to an increasing number of security threats from which corporations must protect them. The recent attacks on popular web sites like Yahoo, eBay and E*Trade, and their consequent disruption of services have exposed the vulnerability of the Internet to Distributed Denial of Service (DDoS) attacks [2]. It has been shown that more than 90% of the DoS attacks use TCP [9]. The TCP SYN flooding is the most commonly-used attack. It consists of a stream of spoofed TCP SYN packets directed to a listening TCP port of the victim. Not only the Web servers but also any system connected to the Internet providing TCP-based network services, such as FTP servers or Mail servers, are susceptible to the TCP SYN flooding attacks. The Attacker creates a random source address for each packet .SYN flag set in each packet is a request to open a new connection to the server from the spoofed IP address victim responds to spoofed IP address, then waits for confirmation that never arrives (waits about 3 minutes) victim's connection table fills up waiting for replies after table fills up, all new connections are ignored legitimate users are ignored as well, and cannot access the server once attacker stops flooding server, it usually goes back to normal state (SYN floods rarely crash servers) newer operating systems manage resources better, making it more difficult to overflow tables, but still are vulnerable SYN flood can be used as part of other attacks, such as disabling one side of a connection in TCP hijacking, or by preventing authentication or logging between servers. To counter SYN

flooding attacks, several defense mechanisms have been proposed, such as Syn cache, Syn cookies, SynDefender , Syn proxying , and Synkill. All of these defense mechanisms are installed at the firewall of the victim server or inside the victim server, thereby providing no hints about the sources of the SYN flooding. They have to rely on the expensive IP traceback to locate the flooding sources. Because the defense line is at, or close to, the victim, the network resources are also wasted by transmitting the flooding packets. Moreover, these defense mechanisms are stateful, i.e., states are maintained for each TCP connection or state computation is required. Such a solution makes the defense mechanism itself vulnerable to SYN flooding attacks. Recent experiments have shown that a specialized firewall, which is designed to resist SYN floods, became futile under a flood of 14,000 packets per second [8]. The stateful defense mechanisms also degrade the end-to-end TCP performance, e.g., incurring longer delays in setting up connections. In the absence of SYN flooding attacks, all the overheads introduced by the defense mechanism become superfluous. We, therefore, need a simple stateless mechanism to detect SYN flooding attacks, which is immune to the SYN flooding attacks. Also, it is preferred to detect an attack early near its source, so that one can easily trace the flooding source without resorting to expensive IP traceback. Depleting the backlog is the goal of the TCP SYN flooding attack, which attempts to send enough SYN segments to fill the entire backlog

The basis of the SYN flooding attack lies in the design of the 3-way handshake that begins a TCP connection. In this handshake, the third packet verifies the initiator's ability to receive packets at the IP address it used as the source in its initial request, or its return reachability. Fig.1 shows the sequence of packets exchanged at the beginning of a normal TCP connection.

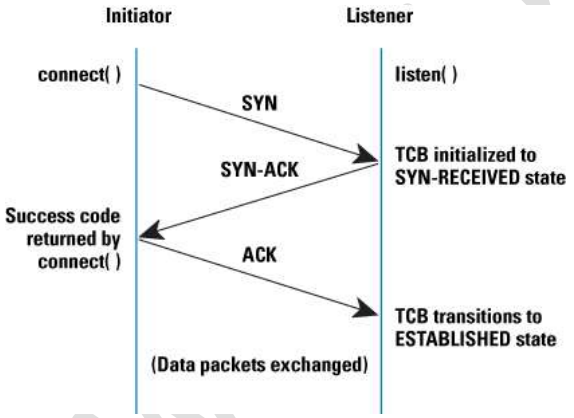

Fig.1 Three way handshake

The Transmission Control Block (TCB) is a transport protocol data structure (actually a set of structures in many operations systems) that holds all the information about a connection. The memory footprint of a single TCB depends on what TCP options and other features an implementation provides and has enabled for a connection. Usually, each TCB exceeds at least 280 bytes, and in some operating systems currently takes more than 1300 bytes. The TCP SYN-RECEIVED state is used to indicate that the connection is only half open, and that the legitimacy of the request is still in question. The important aspect to note is that the TCB is allocated based on the SYN packet— before the connection is fully established or the initiator's return reachability has been verified.

II. TYPES OF ATTACK

An unauthorized user gaining access to a computer (or part thereof) can perform many functions, install different types of devices to compromise security, includingoperating system modifications, software worms, key loggers, and covert listening devices. The attacker can also easily download large quantities of data onto backup media, for instance CD-R/DVD-R, tape; or portable devices such as keydrives,digital cameras or digital audio players. Another common technique is to boot an operating system contained on a CD-ROM or other bootable media and read the data from the harddrive(s) this way. The only way to defeat this is to encrypt the storage media and store the key separate from the system. Direct-access attacks are the only type of threat to Standalone computers (never connect to internet).

This method of attack is very easy to perform because it does not involve directly injecting or spoofing packets below the user level of the attacker's operating system. It can be performed by simply using many TCP *connect()* calls, for instance. To be effective, however, attackers must prevent their operating system from responding to the SYN-ACKs in any way, because any ACKs, RSTs, or *Internet Control Message Protocol* (ICMP) messages will allow the listener to move the TCB out of SYN-RECEIVED. This scenario can be accomplished through firewall rules that either filter outgoing packets to the listener (allowing only SYNs out), or filter incoming packets so that any SYN-ACKs are discarded before reaching the local TCP processing code.When detected, this type of attack is very easy to defend against, because a simple firewall rule to block packets with the attacker's source IP address is all that is needed. This defense behavior can be automated, and such functions are available in off-the-shelf reactive firewalls.

Another form of SYN flooding attacks uses IP address spoofing, which might be considered more complex than the method used in a direct attack, in that instead of merely manipulating local firewall rules, the attacker also needs to be able to form and inject raw IP packets with valid IP and TCP headers. Today, popular libraries exist to aid with raw packet formation and injection, so attacks based on spoofing are actually fairly easy.Many of the protocols in the TCP/IP suite do not provide mechanisms for authenticating the source or destination of a message. They are thus vulnerable to spoofing attacks when extra precautions are not taken by applications to verify the identity of the sending or receiving host. IP spoofing and ARP spoofing in particular may be used to leverage man-in-the-middle attacks against hosts on a computer network. Spoofing attacks which take advantage of TCP/IP suite protocols may be mitigated with the use of firewalls capable of deep packet inspection or by taking measures to verify the identity of the sender or recipient of a message.

For spoofing attacks, a primary consideration is address selection. If the attack is to succeed, the machines at the spoofed source addresses must not respond to the SYN-ACKs that are sent to them in any way. A very simple attacker might spoof only a single source address that it knows will not respond to the SYN-ACKs, either because no machine physically exists at the address presently, or because of some other property of the address or network configuration. Another option is to spoof many different source addresses, under the assumption that some percentage of the spoofed addresses will be unrespondent to the SYN-ACKs. This option is accomplished either by cycling through a list of source addresses that

are known to be desirable for the purpose, or by generating addresses inside a subnet with similar properties.

If only a single source address is repetitively spoofed, this address is easy for the listener to detect and filter. In most cases a larger list of source addresses is used to make defense more difficult. In this case, the best defense is to block the spoofed packets as close to their source as possible.Assuming the attacker is based in a "stub" location in the network (rather than within a transit *Autonomous System* (AS), for instance), restrictive network ingress filtering [7] by stub ISPs and egress filtering within the attacker's network will shut down spoofing attacks—if these mechanisms can be deployed in the right places. Because these ingress/egress filtering defenses may interfere with some legitimate traffic, such as the Mobile IP triangle routing mode of operation, they might be seen as undesirable, and are not universally deployed. *IP Security* (IPsec) also provides an excellent defense against spoofed packets, but this protocol generally cannot be required because its deployment is currently limited. Because it is usually impossible for the listener to ask the initiator's ISPs to perform address filtering or to ask the initiator to use IPsec, defending against spoofing attacks that use multiple addresses requires more complex solutions.

The real limitation of single-attacker spoofing-based attacks is that if the packets can somehow be traced back to their true source, the attacker can be easily shut down. Although the tracing process typically involves some amount of time and coordination between ISPs, it is not impossible. A distributed version of the SYN flooding attack, in which the attacker takes advantage of numerous drone machines throughout the Internet, is much more difficult to stop. The drones use direct attacks, but to increase the effectiveness even further, each drone could use a spoofing attack and multiple spoofed addresses.

Currently, distributed attacks are feasible because there are several "botnets" or "drone armies" of thousands of compromised machines that are used by criminals for DoS attacks. Because drone machines are constantly added or removed from the armies and can change their IP addresses or connectivity, it is quite challenging to block these attacks.

III. RELATED WORK

The burstiness of TCP connection request arrivals makes the detection of attack signatures much harder, since the critical characteristic of self-similar traffic is that there is no natural length of a “burst”. It is also site- and time-dependent.However, the strong positive correlation between SYN and FIN (RST) offers a clear indication for SYN flooding. According to the specification of TCP/IP protocol in normal operation, a FIN (RST) is paired with a SYN at the end of data transmission; but under SYN flooding attacks, this SYN–FIN (RST) pair’s behaviour will be violated, deviating from the normal operation.

In the TCP case, the SYN flooding attack is the most efficient and common-used one which exploits the standard TCP three-way handshake. In the TCP three-way handshake, when the server receives a client’s SYNrequest, it replies with a SYN/ACK packet and then waits for the client to send the ACK to complete the three-way handshake. While waiting for the final ACK, the server maintains a half-open connection. Since the SYN flooding attacker always chooses unreachable addresses as the spoofed source addresses of the attacking packets, the server will not receive the anticipated final ACK from the client. Given that the server has

limited resource for new connections, it is unable to provide service to the forthcoming connection request.

It is important to detect SYN flooding attacks at an early stage before there are a large number of half-open connections maintained by the protected server. Early detection also allows sufficient time for defense responses such as filtering, pushback and traceback.To improve detection efficiency, an active approach is preferred to a passive one. Thought traditional passive methods can give accurate detection result at the later stage when attack signatures become evident, they are inaccurate.

In addition to SYN flooding, several other attacks on TCP connections are possible by spoofing the IP source address and connection parameters for in-progress TCP connections [10]. If an attacker can guess the two IP addresses, TCP port numbers, and a valid sequence number within the window, then a connection can be disrupted either through resetting it or injecting corrupt data. In addition to spoofed TCP segments, spoofed ICMP datagrams have the capability to terminate victim TCP connections.Both these other attacks and SYN floods target a victim's TCP application and can potentially deny service to the victim using an attack rate less than that of brute-force packet flooding. However, SYN flooding and other TCP spoofing attacks have significant differences. SYN flooding denies service to new connections, without affecting in-progress connections, whereas other spoofing attacks disrupt in progress connections, but do not prevent new connections from starting. SYN flooding attacks can be defended against by altering only the initial handshaking procedure, whereas other spoofing attacks require additional per-segment checks throughout the lifetime of a connection. The commonality between SYN flooding and other TCP spoofing attacks is that they are predicated on an attacker's ability to send IP packets with spoofed source addresses, and a similar defense against these attacks would be to remove this capability through more universal deployment of address filtering or IPsec.

IV. IMPLEMENTATION

During the initial Panix attack, random spoofed source addresses were being used, but it was noted that the attack TCP SYNs all used the same source port number. A filter that denied incoming packets from this port was temporarily effective, but easy for the attacker to adapt to, and the attack segments began using random ports. Panix was able to isolate which of its ingress routers the attack was coming from and null-route packets destined for its servers coming through that router, but this solution was obviously a heavy-handed one, and seems to have also been overcome when the attacker started sending packets that were routed through a different upstream provider. Panix had mixed success in getting its providers to assist in tracing and blocking the attack, and the networking community was spurred into devising other solutions.

Two broad classes of solutions to SYN flooding attacks have evolved, corresponding to where the defenses are implemented. The first class of solutions involves hardening the end-host TCP implementation itself, including altering the algorithms and data structures used for connection lookup and establishment, as well as some solutions that diverge from the TCP state machine behavior during connection establishment.

The second class involves hardening the network, either to lessen the likelihood of the attack preconditions (an army of controlled hosts or the propagation of IP packets with spoofed source addresses), or to insert middleboxes that can isolate servers on the networks behind them from illegitimate SYNs.

A. End-host countermeasures

Increasing TCP Backlog: Because the basic attack mechanism relies on overflowing a host's backlog of connecting sockets, an obvious end host-based solution is to simply increase the backlog, as is already done for very popular server applications. In at least some popular TCP implementations, this solution is known to be a poor one because of the use of linear list traversal in the functions that attempt to free state associated with stale connection attempts. Increasing the backlog is typically possible through altering the *listen()* call of an application and setting an operating system kernel parameter named SOMAXCONN, which sets an upper bound on the size of the backlog that an application can request. This step by itself should not be seriously considered as a means to defend against SYN flooding attacks—even in operating systems that can efficiently support large backlogs—because an attacker who can generate attack segments will most likely be able to scale to larger orders than the backlog supportable by a host.In the world of information technology there are different types of cyber attack–like code injection to a website or utilising malware (malicious software) such as virus, trojans, or similar. Attacks of these kinds are counteracted managing or improving the damaged product. But there is one last type, social engineering, which does not directly affect the computers but instead their users, which are also known as "the weakest link". This type of attack is capable of achieving similar results to other class of cyber attacks, by going around the infrastructure established to resist malicious software; since being more difficult to calculate or prevent, it is many times a more efficient attack vector.

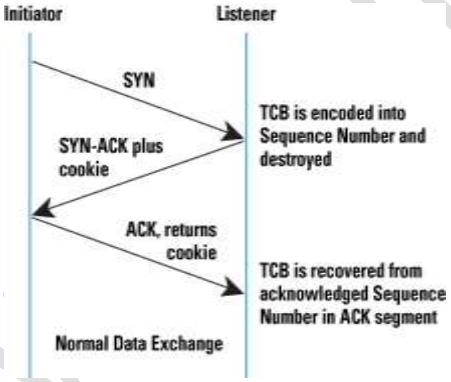

Fig.2: Connection Establishment with SYN Cookies

B. SYN caches

Two end-host defenses, called SYN caches and SYN cookies operate by reducing the amount of state allocated initially for a TCB generated by a received SYN, and putting off instantiating the full state [8]. In a host that uses a SYN cache, a hash table with a limited amount of space in each hash bucket is used to store a subset of the data that would normally go into an allocated TCB. If and when a handshake completing ACK is received, this data can be moved into a full TCB; otherwise the oldest bucket at a particular hash value can be reaped when needed. The SYN cache data structure is robust to attackers attempting to

overflow its buckets because it uses the initiator's local port number and some secret bits in the hash value. Because stacks are a more effective data structure to search than a simple linked list, stacks that use a SYN cache can have improved speed, even when not under attack. Under Lemon's tests, during an active attack a host using a SYN cache was able to establish legitimate connections with only about a 15-percent increase in latency.

C. SYN Cookies

In contrast to the SYN cache approach, the SYN cookies technique causes absolutely zero state to be generated by a received SYN. Instead, the most basic data comprising the connection state is compressed into the bits of the sequence number used in the SYN-ACK. Since for a legitimate connection, an ACK segment will be received that echoes this sequence number (actually the sequence number plus one), the basic TCB data can be regenerated and a full TCB can safely be instantiated by decompressing the Acknowledgement field. This decompression can be effective even under heavy attack because there is no storage load whatsoever on the listener, only a computational load to encode data into the SYN-ACK sequence numbers.

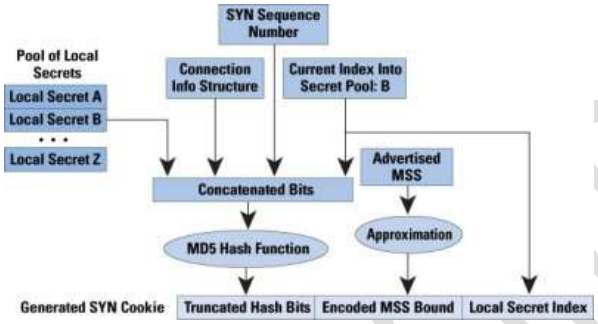

Fig.3: Process for Generation and Validation of TCP SYN Cookies.

D. Network-based countermeasures

Filtering: Networks receive packets from other networks. Normally a packet will contain the IP address of the computer that originally sent it. This allows devices in the receiving network to know where it came from, allowing a reply to be routed back (amongst other things).

However, a sender IP address can be faked ('spoofed'), characterizing a spoofing attack. This disguises the origin of packets sent, for example in a denial-of-service attack.

The most basic network-level defense is application of the filtering techniques described in RFC 2827. Using ingress filtering, an ISP refuses to further route packets coming from an end site with IP source addresses that do not belong to that end site. Ingress filtering would be highly effective at preventing SYN flooding attacks that rely on spoofed IP packets. However, it is not currently reliable because ingress filtering policies are not universally deployed. Ingress filtering is also wholly ineffective against SYN flooding attacks that use a distributed army of controlled hosts that each directly attack. Ingress filtering is also a

mechanism that an end site wishing to defend itself most often has no control over, because it has no influence upon the policies employed by ISPs around the world.

E. Firewalls and proxies:

A firewall or proxy machine inside the network can buffer end hosts from SYN flooding attacks through two methods, by either spoofing SYN-ACKs to the initiators or spoofing ACKs to the listener [9].Fig. 6 shows the basic operation of a firewall/proxy that spoofs SYN-ACKs to the initiator. If the initiator is legitimate, the firewall/ proxy sees an ACK and then sets up a connection between itself and the listener, spoofing the initiator's address. The firewall/proxy splits the end-to-end connection into two connections to and from itself. This splitting works as a defense against SYN flooding attacks, because the listener never sees SYNs from an attacker. As long as the firewall/proxy implements some TCP-based defense mechanism such as SYN cookies or a SYN cache, it can protect all the servers on the network behind it from SYN flooding attacks.

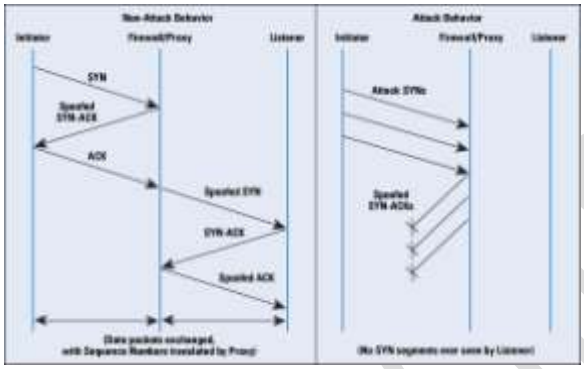

Fig 4: Packet Exchanges through a SYN-ACK spoofing Firewall/Proxy

V. RESULT

In this paper we successfully provided a simple experiment to produce a TCP SYN flooding DDOSattack, we estimate the packet rate on a victim server per second. In this method server sends SYN+ACK+craft message to client back , the result from the learning recording packet analyzer ,checks whether the TCP replyacknowledgement packet satisfies the specification given by the server using TCP probing to change the TCP window size.

To mitigate the SYN flooding attack SYN cookies is used SYN cookies work to alleviate SYN floods by calculating cookies that are functions of the source address, source port, destination address, destination port and random secret seed. On receiving SYN packet the server calculates a SYN cookie and sends it back to client as part of the SYN+ACK and do not allocate resources for the request send by client. when ACK packet is received the connection is established if a valid cookie is present in the ACK packet.SYN cache also used to mitigate the flooding attacks, It use the concept of backlog queue, a minimum amountof state is stored for each SYN request.

VI. CONCLUSION AND FUTURE WORK

There are several factors impacting the accuracy of detection and bringing potential false alerts. When the spoofed IP address of an attacking packet happens to meet an actually congested router, our method cannot find this malicious packet. If there is congestion during attacks, the accuracy will be influenced because it is difficult for our method to distinguish the attack from the network congestion. Though these cases do not happen frequently, these will bring some false negatives. Besides congestion and SYN flooding attack, some other reasons to cause the failure of the three-way handshake, such as errors in routers, will also bring potential false positive. More precise mechanism will be studied to give more accurate distinguish between network congestion and SYN flooding attack.

REFERENCES:

i. Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," BCP 38,RFC 2827, May 2000.

ii. Lemon, J., "Resisting SYN Flood DoS Attacks with a SYN Cache," BSDCON 2002, February 2002.

iii. Schuba, C., Krsul, I., Kuhn, M., Spafford, E., Sundaram, A., and D. Zamboni, "Analysis of a Denial of Service Attack on TCP," Proceedings of the 1997 IEEE Symposium on Security and Privacy.

iv. Touch, J., "Defending TCP Against Spoofing Attacks," Internet- Draft (work in progress), draft-ietf-tcpm-tcp-antispoof-05, October 2006.

v. J.Postel,"TransmissionControlProtocol",RFC793.

vi. Ma, M, "Mitigating denial of service attacks with password puzzles" in Information Technology: Coding and Computing,Vol. 2, May 2005. pp.62 1 - 626.

vii. Bharathi KrishnaKumar, P.Krishna Kumar "Hop Count Based Packet Processing Approach to Counter DDoS Attacks" International Conference on Recent Trends in Information, Telecommunication and Computing, 2010

viii. L .Kavisankar , C. Chellapan ," A Mitigation model for TCP SYN flooding with IP Spoofing", IEEE-InternationalConference on Recent Trends in Information Technology, ICRTIT 2011 , pp. 251-256.

ix. D.J. Bernstein, "SYN Cookies" 1997 [online] ,http://cr.yp.to/syncookies.html

x. Stopforth, Riaan: Techniques and countermeasures of TCP/IP OS fingerprinting on Linux Systems, Thesis, University ofKwaZulu-Natal, Durban, 2007

xi. TCP SYN flooding and common mitigation RFChttp://www.tcpipguide.com/free/t_TCPConnectionPreparationTransmissionControlBlocksT-2.htm

xii. Wireshark,www.wireshark.orhttp://www.frozentux.net/ipsysctltutorial/chunkyhtml/tcpvariables.html

xiii. D.Kirkland,TCPprotocol',http://manpages.ubuntu.com/manpages/aunty/man7/tcp.html

xiv. Eddy, W., "TCP SYN Flooding Attacks and Common Mitigations," Internet-Draft (work in progress), draft-ietf-tcpm-syn-flood-00, July 2006.

xv. BERNSTEIN, D. J. Syn cookies. http://cr.yp.to/syncookies.html.

xvi. BORMAN, D. Bsdi implementation of syn cache. ftp://ftp.bsdi.com/private/ 44-syn-diffs.gz.