# TCP based SYN Flood Attack - Analysis, Detection and Prevention

Hardik K. Molia[1], Sohil M. Gambhir[2], Mahesh D. Titiya[3]
[1, 2, 3]*Department of Computer Engineering, Government Engineering College, Rajkot, Gujarat, India*

***Abstract*— TCP - Transmission Control Protocol is a logical vehicle to transfer data between two processes running on two different computers. TCP is a connection oriented and reliable protocol. The three way handshaking based TCP connection establishment process invites a special Denial of Service (DoS) attack, called SYN Flooding attack. A SYN Flood attacker sends**

**a large number of TCP connection establishment requests to overburden the target system. In real life, the target system is usually the server. As a result of the attack, server remains unavailable and unresponsive for legitimate users too. The Denial of Service nature of the attack targets the availability of the system. This paper analyses and discusses various methods to detect and prevent such attack.**

***Keywords*— TCP, DoS, SYN Flood, SYN Cache, SYN Cookie**

## I. INTRODUCTION

A Denial of Service (DoS) attack targets the availability of the system to its legitimate users. A DoS attack tries to overburden the system by consuming enough resources like computation power and storage space unnecessarily. The main goal is to prevent legitimate users from accessing the system because of temporarily or indefinitely unavailability. TCP – Transmission Control Protocol is a unicast, connection oriented and reliable protocol which acts as a logical vehicle to transfer data between a client and a server. TCP establishes a connection between a process running on a client and another process running on the server by three way handshaking process. SYN Flooding is a DoS attack which exploits the three way handshaking process. A SYN Flood attack sends a large number of connection establishment requests to the server to make server processes unable to reply to the future legitimate client requests, causing unavailability of the services running on the server for them. This paper discusses variants of SYN Flood attack and various countermeasures [1][2][3].

### *A. TCP – Connection Establishment:-*

TCP establishes a unicast and full duplex connection via a series of messages called three way handshaking process. Initially, server in itiates passive open by opening and binding a port at which it listens for new connections. A client can initiate active open by initiating three way handshaking process as shown in Figure. 1 [1].

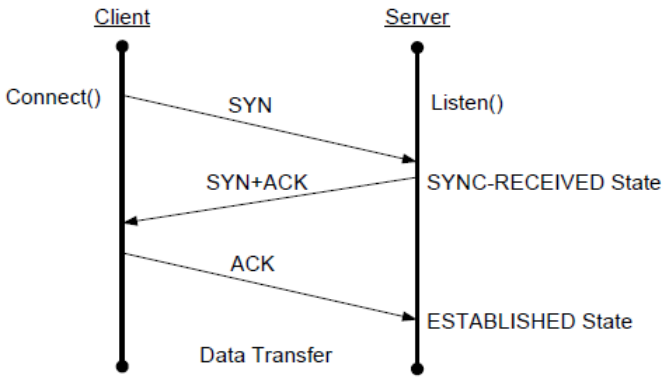


Figure 1. Three Way Handshaking Process

The process can be explained with three steps [1].

1. SYN: A client sends a SYN to the server with a segment's sequence number X.
2. SYN-ACK: Server replies with a SYN-ACK with a segment's sequence number Y and acknowledgement number X+1.
3. ACK: Client sends ACK back to the server with a segment's sequence number X+1 and acknowledgement number Y+1.

## II. SYN FLOOD ATTACK

TCB - Trans mission Control Block is a data structure used to store parameter information about an active TCP connection. The size of TCB varies from 280 bytes to 1300 bytes. Whenever a server TCP receives SYN from client, it sends SYN-ACK back to the client and enters into the SYN-RECIVED state corresponding to the connection. SYN-RECEIVED state indicates half open status of the TCP connection. Increase in the number of active TCP connections increases the number of active TCBs too. Server operating system defines backlog parameter to limit the number of active TCBs (Number of active TCP connections) as per the memory management policies. SYN Flood Attacker initiates the attack by sending large number of connection establishment requests with SYN messages which cause server to reach to the backlog limit soon. As a result, subsequent legitimate connection requests are not proceed until few of the TCBs are discarded (half open TCP connections are terminated). TCP remains in SYN-RECIVED state to wait for ACK from the client to complete the connection establishment process. It is natural

that the TCP should wait for ACK for some time as network may delay or loss ACK. The question is what if TCP never gets ACK from the client? This question is the base behind SYN Flood Attack. Attacker ensures that the server TCP never gets ACK for those half open connections which are init iated by the attacker. TCP has a timeout timer for half open connections. On time out, TCP terminates all active half open connections which may allow subsequent legitimate clients to establish connections. But unfortunately, attacker continuously floods the server to keep unavailability alive [1][2][3]. Various attack scenarios are shown in Figure 2 [5].

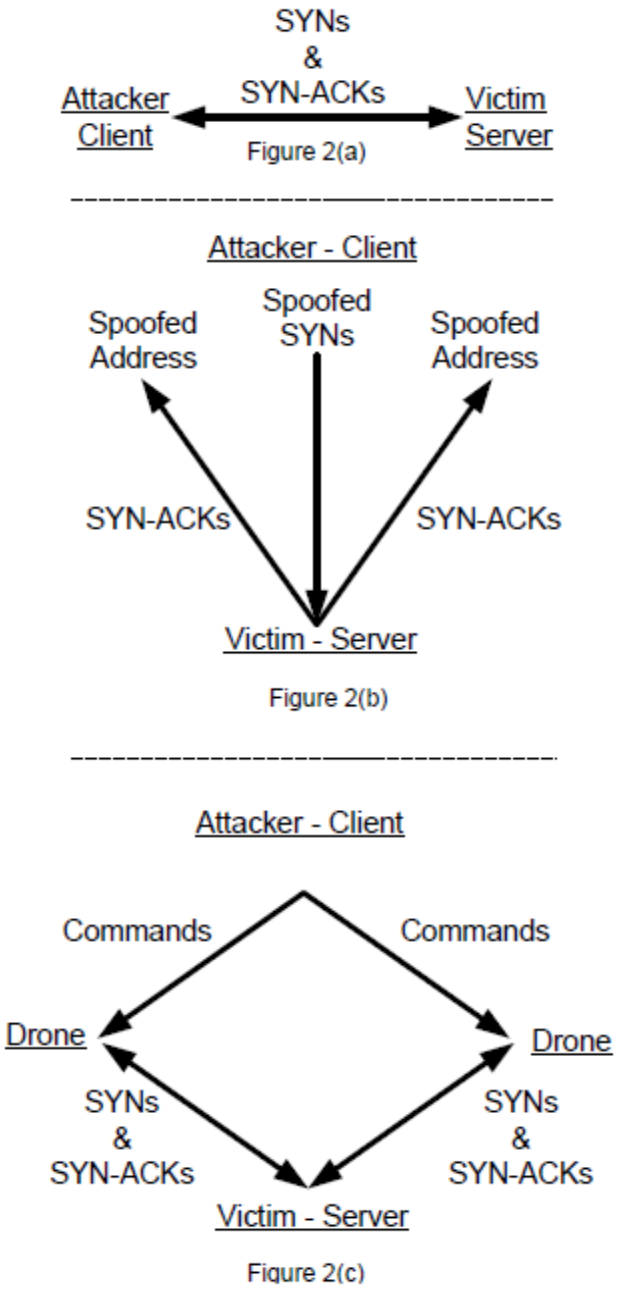


Figure 2. Various SYN Flood Attacks

*A. Direct Attack*

An attacker sends a large number of SYN messages from its own IP address only. This method can be easily implemented by continuous calling TCP connect( ). Attacker must ensure that its system never replies to the corresponding SYN-ACK messages. Any negative reply of the SYN-ACK message can move server TCP out of the SYN-RECEIVED state which discards corresponding TCB to terminate the connection. Attacker accomplish this through its own firewall by setting either outgoing traffic filter to allow only SYN messages for a victim server (Block any other outgoing traffic like ACK, ICMP, RST etc) or incoming traffic filter to discard any SYN-ACK message from the victim server [2][3]. Figure 2(a) shows this attack [5].

As the attacker uses its own IP address, server detects the attack due to a large number of requests from the same IP address. Such attack can be easily prevented by blocking all the traffic from the source IP address of the attacker. Such functionalities are provided with most of the reactive firewalls [2][3].

*B. IP Spoofing based Attack*

IP Spoofing is a way of creation or modification of IP packets with a forged source IP address. The main purpose is to hide the identity of the actual source or to impersonate someone else. IP Spoofing is used for DoS attack where the attacker wants to hide its own identity as well as there is no value of the responses coming from the victim server. Attacker needs to forge only those IP addresses which will never reply to SYN-ACK message in any manner. If attacker uses only one spoofed IP, it will be comparatively easy for server to detect and prevent the attack. Usually attacker changes the spoofed IP periodically from the set of selected IP addresses. An intelligent attacker ignores those IP addresses which are either non-routable or from the reserved – unused IP scheme to difficult detection [2][3]. Figure 2(b) shows this attack [5].

*C. Distributed Attack*

A single IP Spoofing attacker is difficult to detect but after certain efforts, it can be possible to trace the source from which spoofing is being performed. A distributed version of SYN Flooding attack uses a set of systems called drone machines to attack all together. Each drone machine can do direct attack or IP spoofing based attack[2][3]. Figure. 2(c) shows these attacks with Drone machines are doing direct attacks [5].

## III. END – HOST COUNTERMEASURES

*A. Increasing Backlog*

Backlog parameter can be increased by allocating more memory to store TCBs. Ultimately, it increases the number of simultaneously active TCP connections. Standard TCP's data structures and algorithms are not designed to deal with large

backlog. Searching a large backlog is one of the issues. At the same time, there is no scope of much defense as ultimately backlog parameter cannot be infinite [2][3].

*B. Reducing SYN-RECEIVED Timer*

After sending SYN-ACK, TCP waits for corresponding ACK up to SYN-RECEIVED Timer. On t imeout, TCP terminates the half-open connection and discards corresponding TCB. If this timer value is reduced, connections belonging to the attack will be terminated earlier. Subsequently, legitimate connections will get chance to start connection establishment process. This method is also ineffective when legitimate client's messages are delayed or lost due to network congestions. Occasionally, TCP will terminate legitimate half-open connections before their messages will be delivered. At the same time, once resources become free for new connections, attacker's SYN messages will get chance to flood the server again [2][3].

*C. Terminating Oldest Half-Open Connection*

Whenever backlog becomes full, TCP terminates the oldest half-open connection and removes corresponding TCB. The space can be used for newly requested connection. This solution works well when the legitimate connections can be established fully in lesser time than the time required to fill the entire backlog with the attack. This method fails when the attack rate is very high and/or backlog size is small [2][3].

*D. SYN Cache*

This method is based on postponing allocation of full TCB until a connection is fully established. A partial TCB is allocated for every half-open connection. Partial TCBs are stored in global hash table. Once connection is fully established, partial TCB is transferred to the backlog memory. Server TCP selects some random secret bits for every SYN message. Secret bits, source IP and source Port are hashed together to find hash value which specifies bucket to store partial TCB inside the hash table. Every hash table bucket has limited space. Once filled, the oldest entry is deleted for new insertion. The concept of selection of secret bits prevents the attacker from targeting a specific hash table bucket [4].

*E. SYN Cookies*

SYN Cache minimizes the amount of status information which is being allocated init ially on receiving SYN. SYN Cookies initially allocates nothing on receiving SYN. SYN Cookies allocates required TCB only once the connection is completely established. So no status is stored as far as TCP remains in SYN-RECEIVED status. Legitimate clients will reply with ACK and for them TCBs will be allocated. To track the information, SYN Cookies encodes all the required information into a Code called SYN Cookie and sent along with the SYN-ACK message as an ISN-Initial Sequence Number to the client. So instead of sending any randomly generated ISN, SYN Cookies method carefully constructs. The connection state can be reconstructed from the corresponding ACK message as it contains ISN+1 value which is SYN Cookie+1 as an acknowledgement number. Some time dependent information must be encoded to prevent replay attack. One such implementation is shown here [4][5].

*Encoding*

1. Set parameter t = Slowly Incrementing Timestamp
2. Set parameter m = Maximum Segment Size
3. Set parameter s = Value of Hash Function calculated over Server IP, Server Port, Client IP, Client Port and value of parameter t. Size of s is 24 Bits
4. The SYN Cookie (In itial Sequence Number) can be calculated as follow.

| t Mod 32 | m | s |
|---|---|---|
| 5 Bits | 3 Bits | 24 Bits |

*Decoding*

1. Subtract 1 from the Acknowledgement Number received from ACK Message.
2. Compare current time with t for connection expiration.
3. Recalculate s to verify the validity of SYN Cookie.
4. Extract m to set Maximum Segment Size field in corresponding TCB.

*F. Hybrid Approach*

A scheme can be a combination of two or more approaches. For example, SYN Cache and SYN Cookies methods can be used together. For exa mple, if SYN Cache becomes full, further connection requests can be entertained by SYN Cookies method [1][2].

## IV. NETWORK BASED COUNTERMEASURES

*A. Filtering*

Packet filtering can be used to prevent the attack at certain level. Ingress filtering blocks outside packets with IP addresses inside the network to defense against outside attacker. Egress filtering blocks inside packets with IP addresses outside the network to defense against inside attacker [5][6].

*B. SYN-ACK Spoofing Firewall / Proxy*

This method is also called Firewall / Proxy as a relay. A Firewall / Pro xy can send spoofed SYN -ACK message to the client. Firewall splits the end-to-end connection between client and server into two connections to and from the firewall. The splitting protects server as it never sees any SYN message directly, neither from the legitimate client nor by the attacker. As far as firewall implements any End-Host (TCP-

based) defense method, it can protect all the servers which are located behind it. This method is shown in Figure 3 [5].

Figure 3(a) shows a non attack scenario. Whenever a legitimate client tries to establish a TCP connection, corresponding SYN message will be received by the Firewall rather than directly by the server itself. Firewall will send Spoofed SYN+ACK message back to the client. The client will assume it has received SYN+ACK message from the server and so it will send ACK message to complete the connection establishment process. On receiving ACK, Firewall will consider the client as a legitimate client and will establish a connection with the server by sending Spoofed SYN and subsequently Spoofed ACK message. This way a TCP connection will be split across client, firewall and server [5].

Figure 3(b) shows an attack scenario. An attacker will send SYN messages continuously which will be received by the Firewall. Firewall will send corresponding Spoofed SYN+ACK messages too. But as attacker will never reply with ACK message, Firewall will never consider attacker as a legitimate client and so will never establish corresponding connection with the server [5].

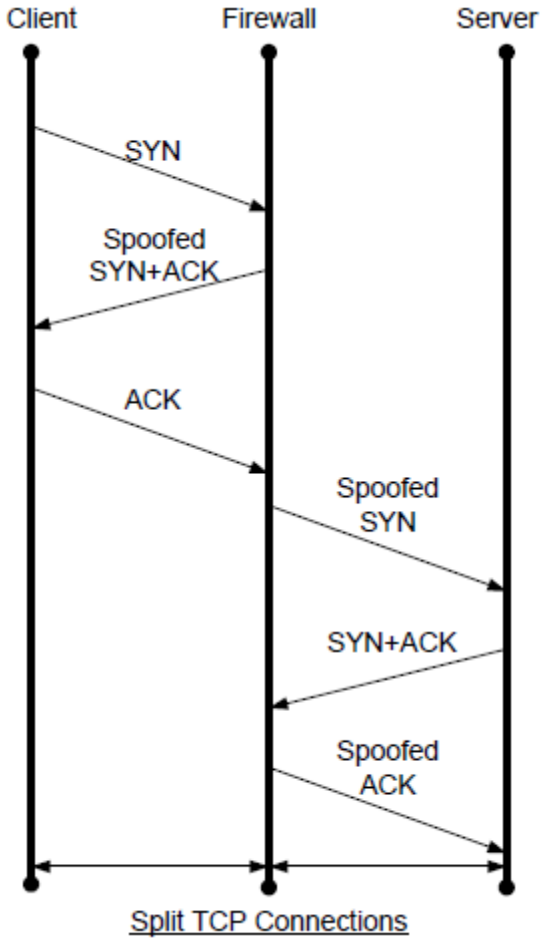


Figure 3(a). Non Attack Scenario

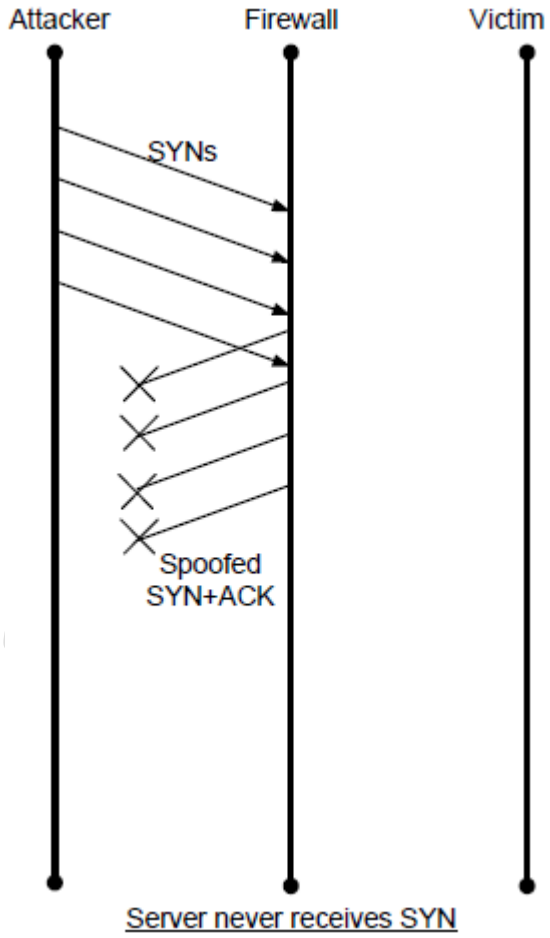


Figure 3(b). Attack Scenario

*C. ACK Spoofing Firewall / Proxy*

This method is also known as Firewall / Pro xy as a semitransparent gateway. A Firewall / Pro xy can send spoofed ACK message to the server. This method sends a spoofed ACK in response of SYN-ACK message immediately. Firewall waits for a legitimate ACK from the client, if it doesn't receive within limited time, it informs server to free TCB by sending a spoofed RST message. This method is shown in Figure 4 [5][7].

Figure 4(a) shows a non attack scenario. A client will send a SYN message which will be directly received by the server through the firewall. Server will generate SYN+ACK message which will be sent to the client via the firewall. Once firewall will receive SYN+ACK message, it will immediately send a corresponding Spoofed ACK message to the server. Later on when client will send ACK message, firewall will simply sends it to the server too [5].

Figure 4(b) shows an attack scenario. An attacker will send a SYN message which will be directly received by the server through the firewall. Server will generate SYN+ACK message which will be sent to the attacker via the firewall. Once firewall will receive SYN+ACK message, it will

immediately send a corresponding Spoofed ACK message to the server. But as attacker will never reply to the SYN+ACK message with a ACK message, Firewall will terminate the Spoofed connection to the server by sending a Spoofed RST message after a timeout[5][7].

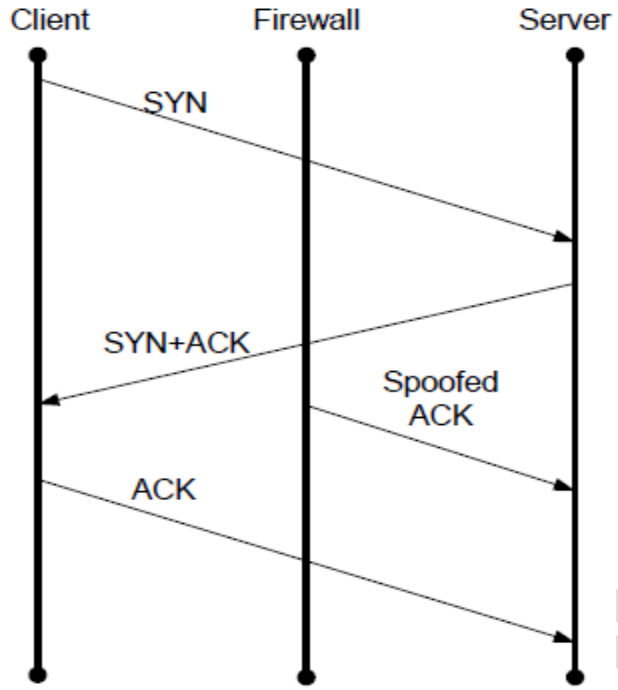


Figure 4(a). Non Attack Scenario

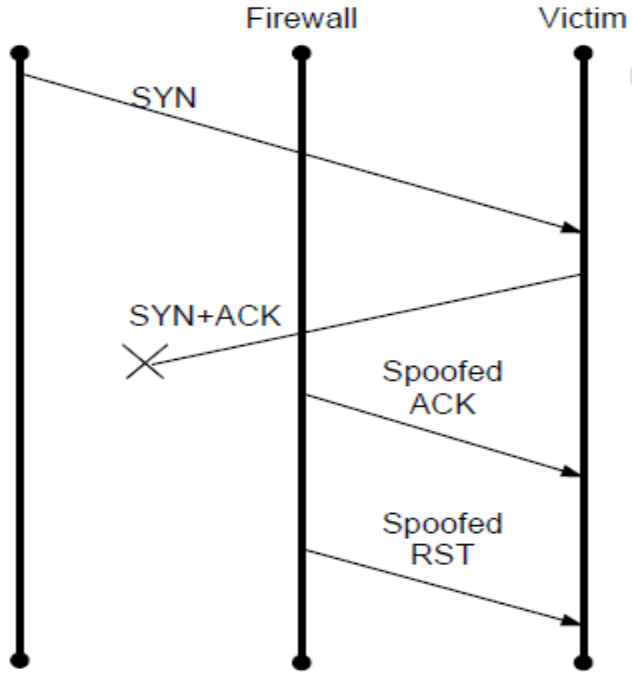


Figure 4(b).Attack Scenario

## V. CONCLUSION

This paper has described SYN Flood Attack and its various forms as a Denial of Service attack for TCP. Various countermeasures are broadly classified under host based solutions and network based solutions. No solution has been accepted as a standard yet. Some of the solutions are already available in commercial security products. Future research work can be done by using Artificial Intelligence based techniques to detect and prevent such attacks.

## REFERENCES

[1]. CERT , CERT Advisory CA-1996-21 T CP SYN Flooding and IP Spoofing Attacks.
[2]. Touch, J., Defending T CP Against Spoofing Attacks, Internet-Draft (work in progress), draft-ietf-tcpm-tcp-antispoof-05.
[3]. Eddy, W., TCP SYN Flooding Attacks and Common Mitigations, Internet-Draft (work in progress), draft -ietf-tcpm-syn-flood-00.
[4]. Lemon, J., Resisting SYN Flood DoS Attacks with a SYN Cache, BSDCON 2002.
[5]. Wesley M. Eddy, Verizon Federal Network Systems, Defenses Against T CP SYN Flooding Attacks, The Internet Protocol Journal - Volume 9, Number 4
[6]. 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.
[7]. Geetha, K.; Sreenath, N. ,SYN flooding attack — Identification and analysis, Information Communication and Embedded Systems (ICICES), 2014.