Firewalls and Intrusion Detection Systems: Part 2

Gaurav S. Kasbekar

Dept. of Electrical Engineering

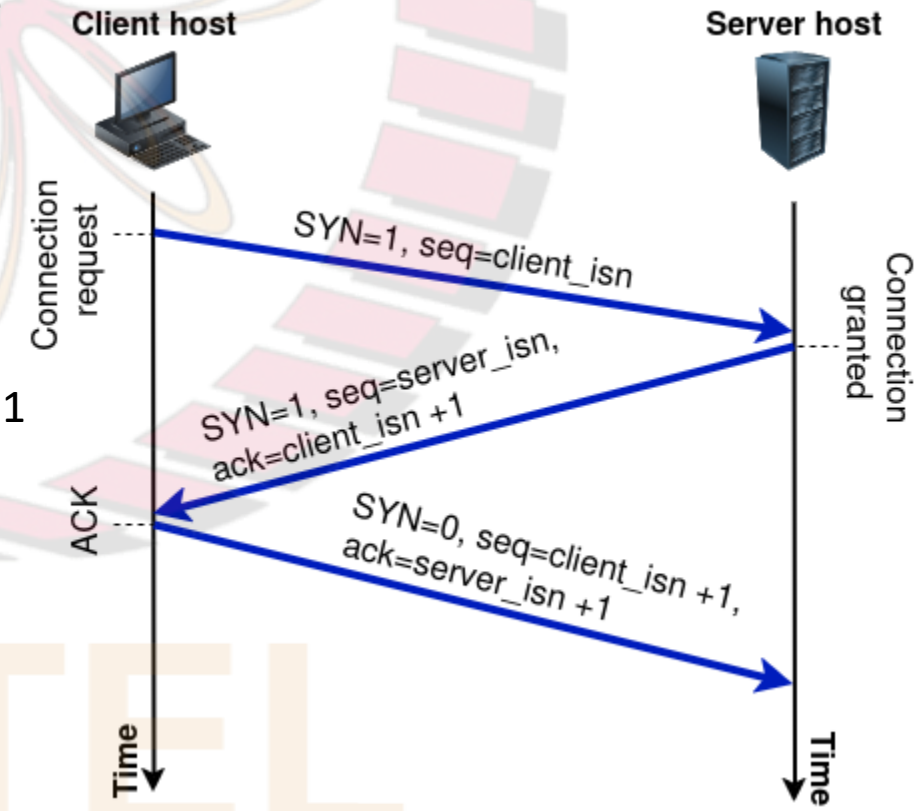
IIT Bombay

References

- J. Kurose, K. Ross, "*Computer Networking: A Top Down Approach*", Sixth Edition, Pearson Education, 2012
- B.L. Menezes, R. Kumar, "*Cryptography, Network Security, and Cyber Laws*", Cengage Learning India Pvt. Ltd., 2018
- C. Kaufman, R. Perlman, M. Speciner, "*Network Security: Private Communication in a Public World*", Pearson Education, 2nd edition, 2002

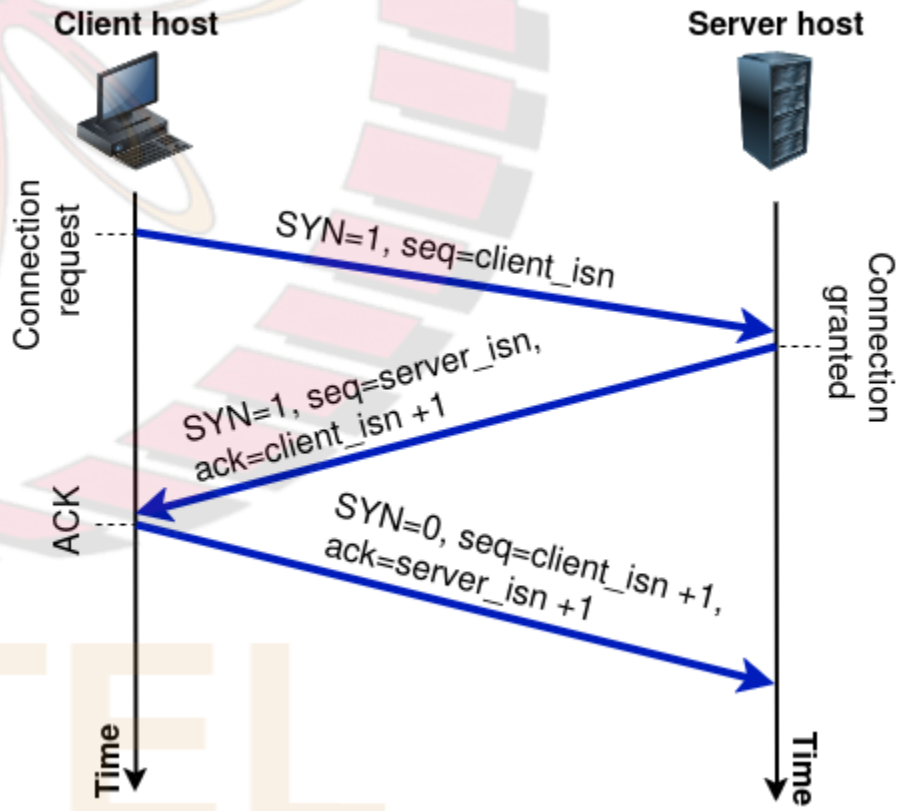
TCP Three-Way Handshake

- Fig. shows how a TCP connection is established
- Client-side TCP process first sends a special TCP packet to server-side TCP process
  - Contains no application-layer data
  - But one of the flag bits in the TCP header, the SYN bit, is set to 1
    - Hence, this packet referred to as "SYN" packet
  - Also, client randomly chooses an initial sequence number (client_isn) and puts it in sequence number field of TCP header
- Once SYN packet arrives at server, the server-side TCP process:
  - Allocates TCP buffers and variables to the connection
  - Sends a packet to client-side TCP process indicating that connection is granted
  - This packet contains no application-layer data
  - SYN bit of this packet is set to 1
  - Acknowledgement field set to client_isn+1
  - Finally, server chooses its own initial sequence number randomly (server_isn) and puts it in sequence number field of TCP header
  - This packet referred to as "SYNACK" packet



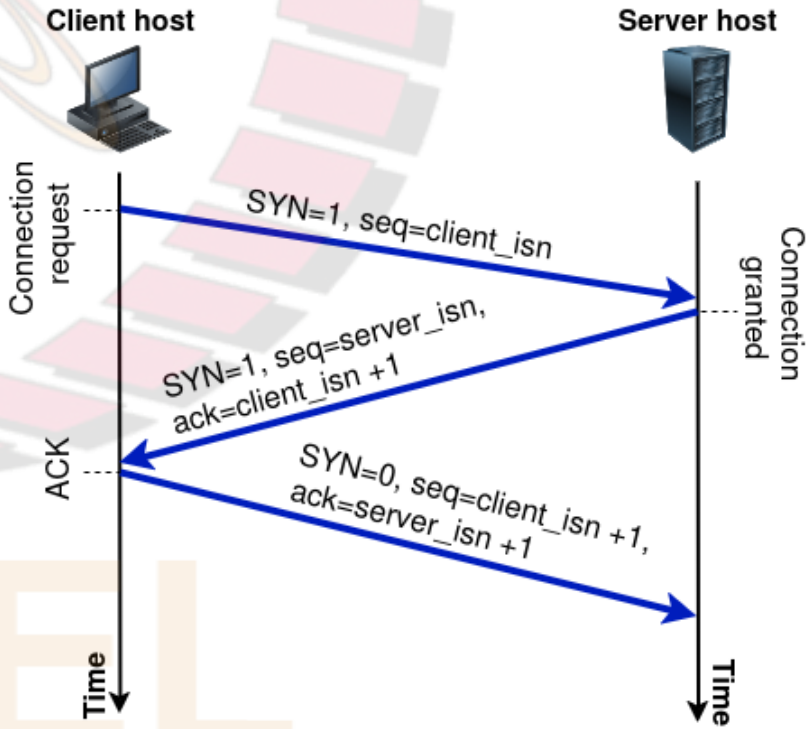
TCP Three-Way Handshake (contd.)

- Upon receiving the SYNACK packet:
  - ❑ the client also allocates buffers and variables to the connection
  - ❑ client then sends the server another packet
  - ❑ this last packet acknowledges the server's connection-granted packet (the client does so by putting the value server_isn+1 in the acknowledgment field of the TCP header)
  - ❑ SYN bit is set to zero
  - ❑ This third stage of the three-way handshake may carry client-to-server data in the packet payload
- Once these three steps have been completed, the client and server hosts can send packets containing data to each other
- In each of these future packets, the SYN bit will be set to zero



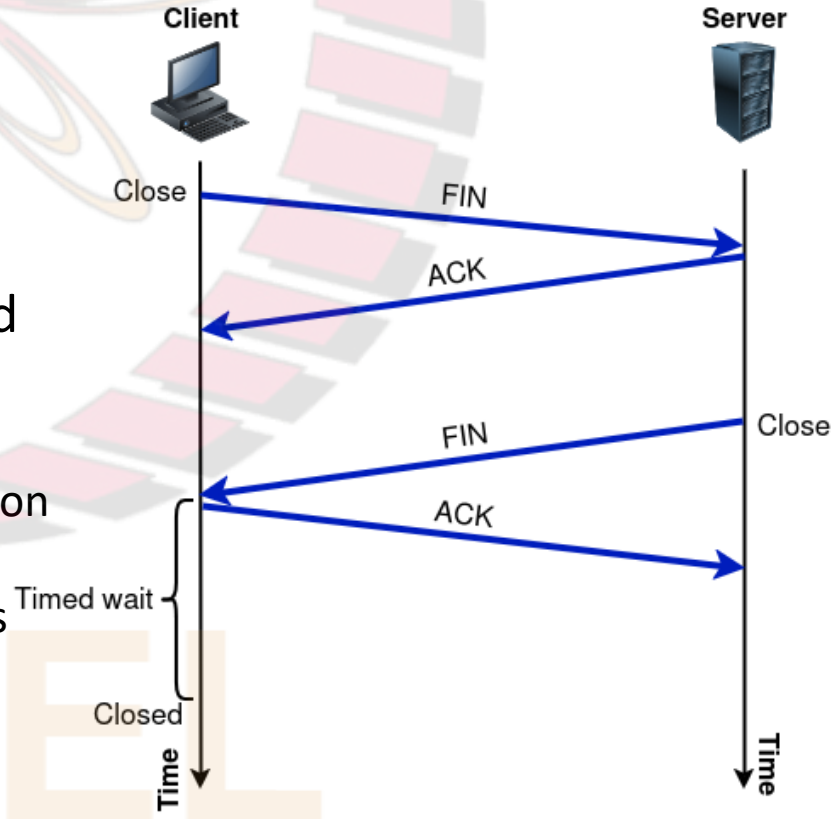
TCP Three-Way Handshake (contd.)

- Reason for using three-way handshake instead of directly starting the exchange of data packets:
  - ❑ each side randomly selects an initial sequence number (ISN) and needs to notify the other side of its selected ISN
  - ❑ in first packet, client notifies server of its selected ISN, client_isn
  - ❑ in second packet, server acknowledges receipt of client_isn as well as notifies client of its selected ISN, server_isn
  - ❑ in third packet, client acknowledges receipt of server_isn
- Reason for using random ISN instead of starting from 0 each time:
  - ❑ suppose initially there is a connection between a client and server, which ends, and then another connection is established
  - ❑ random ISN for each connection prevents surviving packets from old connection from being confused with those from new connection



TCP Connection Termination

- Either the client TCP or server TCP process can end the TCP connection
- When a connection ends, the resources (i.e., buffers and variables) in the hosts are deallocated
- The case where client decides to close the connection shown in fig.
- First, client sends a packet to server in which the FIN bit in TCP header is set to 1
- Server sends an ACK in return
- Server then sends a packet with FIN bit set and client ACKs it
- At this point, all the resources in the two hosts are deallocated
- Why is second message (ACK) not combined with third message (FIN)?
  - ❑ since processes on the two sides must independently close their half of the connection
  - ❑ if only one side closes the connection, then it means that it has no more data to send, but is still available to receive data from other side



SYN Flood Attack

- Recall:
  - a server allocates and initializes connection variables and buffers in response to a received SYN
  - the server then sends a SYNACK in response, and awaits an ACK packet from the client
- If the client does not send an ACK to complete the third step of this three-way handshake, eventually (often after a minute or more), the server will terminate the half open connection and reclaim the allocated resources
- This TCP connection management protocol is vulnerable to a Denial-of-Service (DoS) attack known as the *SYN flood attack*
- In this attack, the attacker(s) send a large number of TCP SYN packets, without completing the third handshake step
- With this deluge of SYN packets, the server's connection resources become exhausted as they are allocated (but never used) for half-open connections; legitimate clients are then denied service

Defence Against SYN Flood Attacks Using Cookies

- A defence against SYN flood attacks known as SYN cookies is now deployed in most major operating systems
- Works as follows
- When server receives a SYN packet
  - It does not know if the packet is coming from a legitimate user or is part of a SYN flood attack
  - So, instead of creating a half-open TCP connection for this SYN, the server creates an initial TCP sequence number that is computed by applying a cryptographic hash function to a quantity derived from source and destination IP addresses and port numbers of the SYN packet, as well as a secret number only known to the server
    - This carefully designed initial sequence number called “cookie”
  - Server then sends the client a SYNACK packet with this special initial sequence number
- Importantly, *server does not remember the cookie or any other state information corresponding to the SYN*

Defence Against SYN Flood Attacks Using Cookies (contd.)

- A legitimate client will return an ACK packet
- When the server receives this ACK, it must verify that the ACK corresponds to some SYN sent earlier
  - Since the server maintains no memory about SYN packets, this is done using the cookie
  - Recall: for a legitimate ACK, the value in the acknowledgment field is equal to the initial sequence number in the SYNACK (the cookie value in this case) plus one
  - Server can then run the same hash function using the source and destination IP address and port numbers in the ACK packet (which are the same as in the original SYN) and the secret number
  - If the result of the function plus one is the same as the acknowledgment number in the client's ACK packet, the server concludes that the ACK corresponds to an earlier SYN packet and is hence valid
  - The server then creates a fully open connection and allocates resources
- On the other hand, if the client does not return an ACK packet, then:
  - The original SYN has done no harm at the server, since the server hasn't yet allocated any resources in response to the original bogus SYN
- Later, we will study an alternative technique to defend against a SYN flood attack, which uses an Intrusion Detection System

Domain Name System (DNS)

- Internet hosts are addressed in multiple ways:
  - using hostnames (e.g., www.google.com, timesofindia.indiatimes.com), which are easy to remember for humans
  - using IP addresses (e.g., 121.7.106.83), which are hierarchically assigned for facilitating routing
- DNS is a directory service that translates hostnames to IP addresses
- Components:
  - a distributed database implemented in a hierarchy of DNS servers, and
  - an application-layer protocol that allows hosts to query the distributed database
- DNS protocol runs over UDP and uses port 53