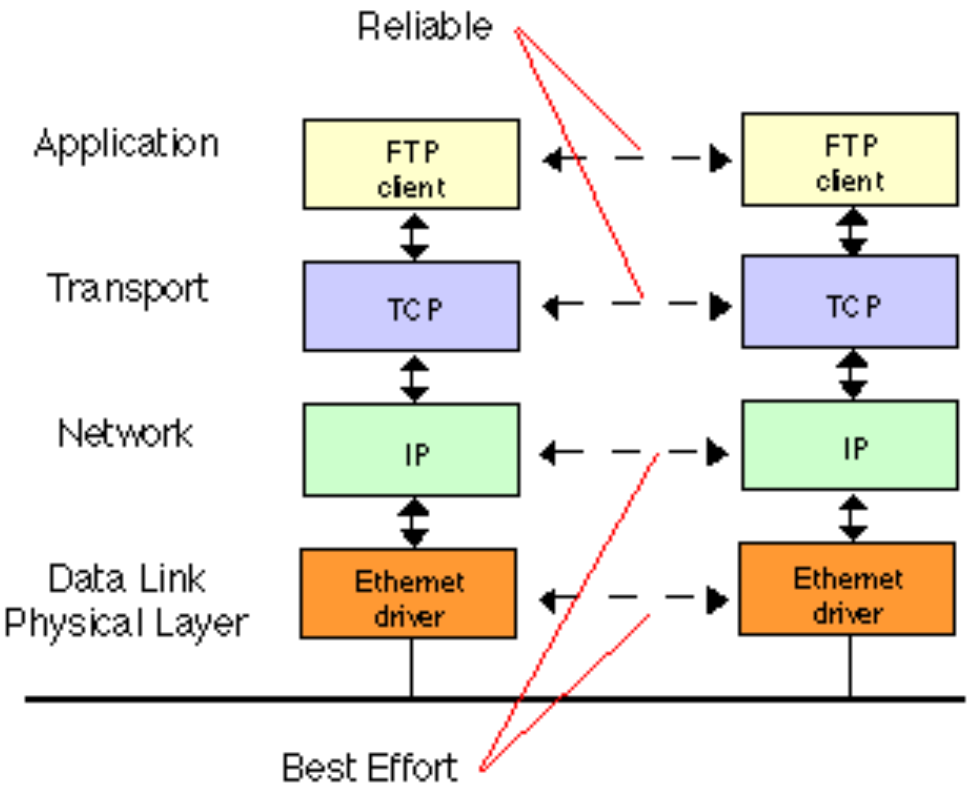
ECE509
Cyber Security :
Concept, Theory, and Practice

Lecture 3: Network Security – Part 1

Fall 2023

Relevant Network Layers



*From http://www.erg.abdn.ac.uk/users/gorry/course/images/ftp-tcp-enet.gif

UDP Client/Server Programming

- C:
  - http://www.cs.ucsb.edu/~almeroth/classes/W01.176B/hw2/examples/
- Java:
  - http://www.cs.uic.edu/~troy/spring05/cs450/sockets/socket.html

Common Terminology

- **NIST:** National Institute of Standards and Technology
- **CAN:** Candidate Vulnerabilities
- **CVE:** Common Vulnerabilities and Exposures
- **CVSS:** Common Vulnerabilities Scoring System
- **IETF:** Internet Engineering Task Force
- **RFC:** Request for Comments
- **STD:** Internet Standard
- **IANA:** Internet Assigned Numbers Authority

Address Resolution Protocol (ARP)

- Used to discover mapping of neighbouring Ethernet MAC to IP addresses.
  - Need to find MAC for 192.168.1.3 which is in your interfaces subnetwork
  - Broadcast an ARP request on the link
  - Hopefully receive an ARP reply giving the correct MAC
  - The device stores this information in an ARP cache or ARP table

ARP cache poisoning

- Anyone can send an ARP reply
  - The Ingredients to ARP Poison,
  - http://www.informit.com/articles/article.aspx?p=29750&seqNum=5
- Classic Man-in-the-middle attack
  - Send ARP reply messages to device so they think your machine is the one that has the IP address in the ARP request message
  - Better than simple sniffing because packets will get to you regardless of sniffing.
- Solutions
  - Encrypt all traffic
  - Monitoring programs like *arpwatch* to detect mapping changes
    - Which might be valid due to DHCP

Transport layer

- UDP
  - Best effort delivery
  - Connectionless
- TCP
  - Reliable
  - Establishes connections and monitors deliveries

UDP Header

Source Port	Destination Port
UDP Length	UDP checksum

TCP Header

Source Port								Destination Port
Sequence Number								
Acknowledgement number								
HDR Len		URG	ACK	PSH	RST	SYN	FIN	Window Size
Checksum								Urgent Pointer
Options (0 or more words)								

UDP - Datagram Transport

- User Datagram Protocol (UDP)
  - A best-effort delivery, no guarantee, no ACK
  - Lower overhead than TCP
  - Good for best-effort traffic like periodic updates
  - No long lived connection overhead on the endpoints
  - Connectionless
- Some folks implement their own reliable protocol over UDP to get “better performance” or “less overhead” than TCP
  - Such efforts don’t generally pan out
- TFTP, DNS, VOIP, P2P Data protocols use UDP
- Data channels of some multimedia protocols, e.g., H.323 also use UDP

Ports

- Ports dynamically “bind” IP packets to a process
- Port range 0 - 65535
- Both TCP and UDP use ports for
  - Transport address selection
    - To identify which service or application to communicate
  - Multiplexing
    - To allow multiple connections per host
- Relationship with Socket

Ports (cont'd)

- Applications are associated with ports (generally just destination ports)
    - IANA organizes port assignments http://www.iana.org/
- Server listening on destination port
  - TCP and UDP have distinct ports, but services usually use the same number for both
- Source ports generally dynamically selected
  - Ports under 1024 are considered well-known ports
  - Would not expect source ports to come from the well-known range
- Scanners probe for listening ports to understand the services running on various machines
- Most Operating Systems allow only privileged processes to open the ports below 1024
    - HTTP 80 TCP
    - SMTP 25 TCP
    - DNS 53 UDP
    - HTTPS 443 TCP

Transport Flow

- **Transport Flow ::** a sequence of packets sent between a source/destination pair and following the same route through the network.
- <src_ip, dst_ip, src_port, dst_port,>
- Total combinations 2^{32} X 2^{32} X 2^{16} X 2^{16} = 2^{96}
- What's the problem with this **BIG** number?
- With a computer operating at 2^{12} instructions per second, and assume the year has 2^{25} seconds, it will take 2^{62} number of years to finish
  - assuming each combination can be done in one instruction; unrealistic assumption.

UDP Issues

- All lower layer issues, with similar attacks
  - IP spoofing
  - IP and link layer broadcast (amplification)
  - IP fragmentation
  - ARP spoofing
  - Link layer
- New possibilities
  - Network services and applications can be contacted and attacked with UDP packets that exploit the lower level issues
  - Traffic amplifying applications

UDP Amplifier Attack

- Fraggle
  - Broadcast UDP packet sent to the "echo“ (Port 7) service
  - All computers reply (amplification)
  - Source IP was spoofed, victim is overwhelmed

UDP Ping-Pong

- Chargen service (Port 19) replies with a UDP packet to any incoming packet
  - It sends arbitrary characters to the sender until the host closes the connection
- Spoof a packet from host A's chargen service to host B's chargen service
  - Computers keep replying to each other as fast as they can
- Variants use the echo service on one of the hosts
  - Or even the same host (CVE-1999-0103)
    - a.k.a. UDP bomb, UDP packet storm

UDP Ping-Pong (Cont'd)

- Any service or application that issues a UDP reply no matter what is the input packet (e.g., error message) is vulnerable
  - daytime (port 13)
  - time (port 37)
- Do you know of another UDP service that answers no matter what?

Example Hosts Vulnerable to UDP Ping-Pong

- Routers and firewalls!
- Cisco IOS 11.x had *chargen* and *echo* enabled by default
  - Date
- Other services
  - Quote of the day (RFC 865)
  - Active Users (RFC 866)
  - Daytime (RFC 867)
  - UDP Kerberos v5 (port 464)
  - Any service that responds (e.g. with an error message) to any packet

Amplification Using UDP Packets

- Key: Applications that reply with large packets to small requests
  - e.g., games
    - BattleField 1942
    - Quake 1 (CAN-1999-1066)
    - Unreal Tournament
- Hosts can be attacked by using these applications as amplifiers, with forged source IP packets

Exploits Through UDP

- Resource Exhaustion
  - Windows 98 and Windows 2000 Java clients allow remote attackers to cause a denial of service via a Java applet that opens a large number of UDP sockets, which prevents the host from establishing any additional UDP connections, and possibly causes a crash.
    - CAN-2001-0324
- Sniffing and Spoofing
  - NAI Sniffer Agent allows remote attackers to gain privileges on the agent by sniffing the initial UDP authentication packets and spoofing commands.
    - CAN-2000-1159

Exploits Through UDP (more)

- Exploitation of other flaws (anonymous)
  - Interactions between the CIFS Browser Protocol and NetBIOS as implemented in Microsoft Windows 95, 98, NT, and 2000 allow remote attackers to modify dynamic NetBIOS name cache entries via a spoofed Browse Frame Request in a unicast or UDP broadcast datagram.
    - CAN-2000-1079
- Traffic amplifiers
  - DNS allows remote attackers to use DNS name servers as traffic amplifiers via a UDP DNS query with a spoofed source address, which produces more traffic to the victim than was sent by the attacker.
    - CVE-1999-1379

Exploits Through UDP (more)

- Self-connection
  - Quake 2 server allows remote attackers to cause a denial of service via a spoofed UDP packet with a source address of 127.0.0.1 (loopback IP address), which causes the server to attempt to connect to itself.
    - CAN-1999-1230
    - Similar to UDP bomb

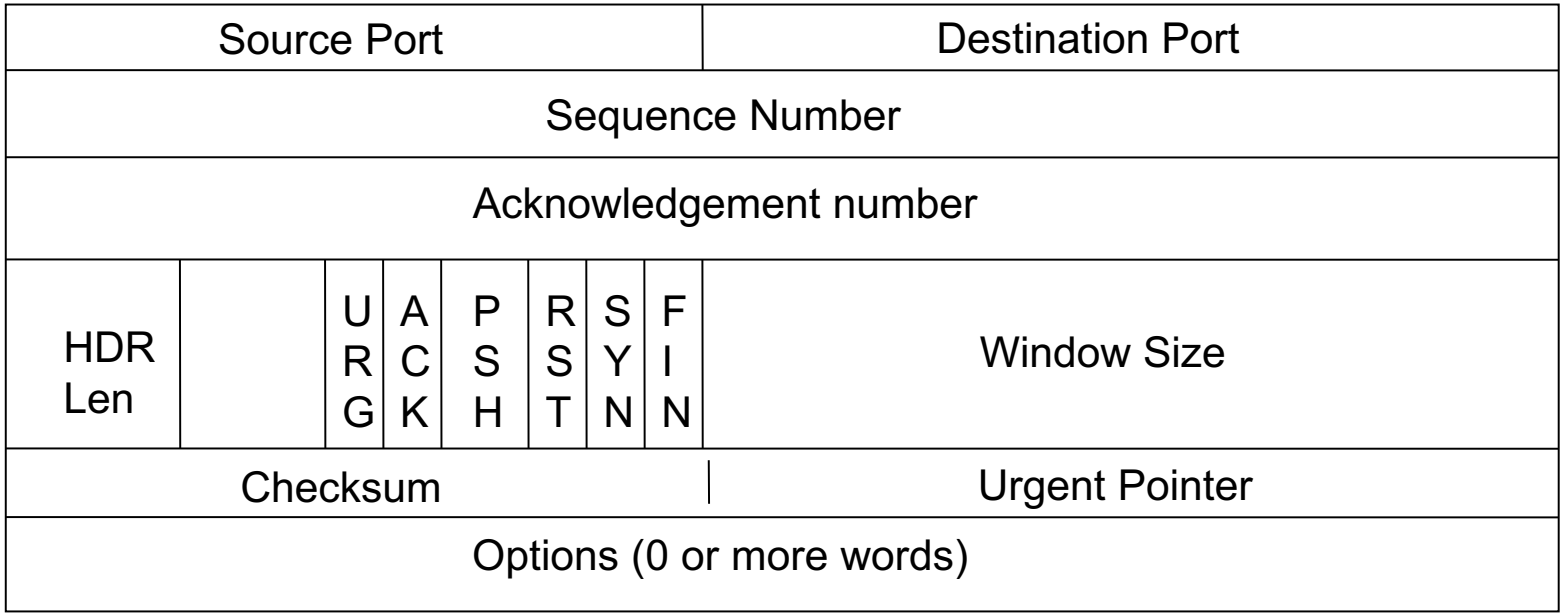
Discussion and Conclusion

- UDP does not in itself introduce new vulnerabilities, but makes the exploitation of IP layer vulnerabilities easy.
  - Makes applications more difficult to design to prevent amplification and ping-pong effects
- When is UDP needed?
    - DNS
      - Normal hosts query DNS servers using UDP in practice
        - » UDP also used for other DNS functions (more on this later)
    - Streaming video, Voice-over-IP
- Is your LAN used to attack a third party via UDP?
    - Did some computers in your LAN get compromised?

TCP - Reliable Streams

- Transmission Control Protocol (TCP)
  - Guarantees reliable, ordered stream of traffic
  - Such guarantees impose overhead
  - A fair amount of state is required on both ends
  - Connection oriented
    - Similar to packages requiring signatures at delivery
- Most Internet protocols use TCP, e.g., HTTP, FTP, SSH, H.323 control channels

TCP Header


Source Port
Destination Port
Sequence Number
Acknowledgement number
HDR Len
URG
ACK
PSH
RST
SYN
FIN
Window Size
Checksum
Urgent Pointer
Options (0 or more words)

TCP Flags

- one-bit
- Synchronize flag [SYN]
  - Used to initiate a TCP connection
- Acknowledgement flag [ACK]
  - Used to confirm received data
- Finish flag [FIN]
  - Used to shut down the connection
- Push flag [PSH]
  - Do not buffer data on receiver side – send directly to application level
- Urgent flag [URG]
  - Used to signify data with a higher priority than the other traffic
    - I.e Ctrl+C interrupt during an FTP transfer
- Reset flag [RST]
  - Tells receiver to tear down connection immediately

Initial Vulnerability

- Establishing connections
  - SYN flood attack
- Is this packet relevant?
  - Initial sequence number predictability
  - RST attacks
- TCP Scanning
  - Tcptraceroute
- Refers to the network, TCP is "Stateless"
  - Phone conversations require a network path to be established
  - TCP does not change network paths
  - Each packet is independent of others
- Clients and servers maintain states, which makes them vulnerable to resource exhaustion attacks

TCP Connections

A client A wants to set up a TCP connection to a server B

- A sends SYN with its sequence number X
- B replies with its own SYN and sequence number Y and an ACK of A’ s sequence number X+1
- A sends data with its sequence number X+1 and ACK’ s B’ s sequence number Y+1

- This establishes that packets can be sent both ways and provides the proof to both hosts.

TCP SYN Scans

- If someone sends you a SYN packet for a port that is closed,
  - you are supposed to respond with a packet with RST and ACK flags (“I got your message but I don’t want to talk to you”).
- Sending SYN packets to find out which ports are open on which machines is known as port scanning
  - Source IP address may be spoofed to hide the true source

TCP ACK Scans

- Bypass firewalls that only allow "established" connections
  - they block incoming packets with the "SYN" flag)
  - Doesn't work if the firewall builds a table of outgoing connections
    - e.g., Network Address Translation, a.k.a. IP masquerading
- Response is a RST packet whether the port is closed or open
- Allows attackers to find out which IP addresses are in use, similar in function to an ICMP ping

TCP FIN Scans

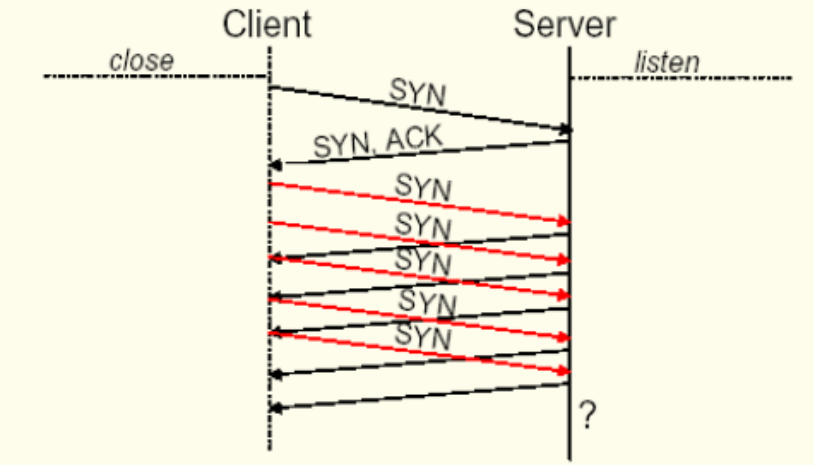
- RFC says:
  - open port, do not respond
  - closed port, respond with RST/FIN
- Some implementations respond with a RST on open ports
- Another way to map services on a host

Defending Scans

- Issue: spoofed IP addresses
  - Option 1: Don't reply
  - Option 2: "Active" defence
    - In a SYN scan, if you send a SYN/ACK for every packet, you could force the attacker to complete the connection to gain information
      - Makes it more difficult to spoof the source IP
      - Slows down scanners
        - » But if 1 in 100 packets is not spoofed, it slows down your server 100 times more than the scanner!
      - However:
        - » Increases traffic, bandwidth consumption
        - » May have undesired effects
        - » Replies sent to spoofed IP addresses
        - » Are you unwittingly attacking them?

SYN flood

- A resource DoS attack focused on the TCP three-way handshake
- Denial of service when an attacker sends many SYN packets to create multiple connections without ever sending an ACK to complete the connection, aka SYN flood.



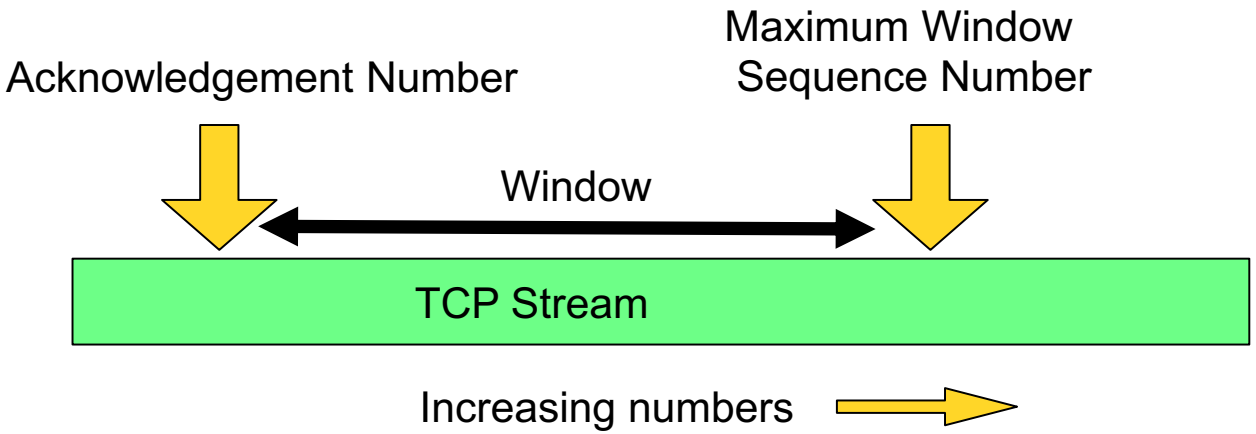
  - This leaves B with a bunch of half open connections that are filling up memory
  - Firewalls adapted by setting limits on the number of such half open connections.
  - Keeping track of each half-open connection takes up resources

TCP Reliability

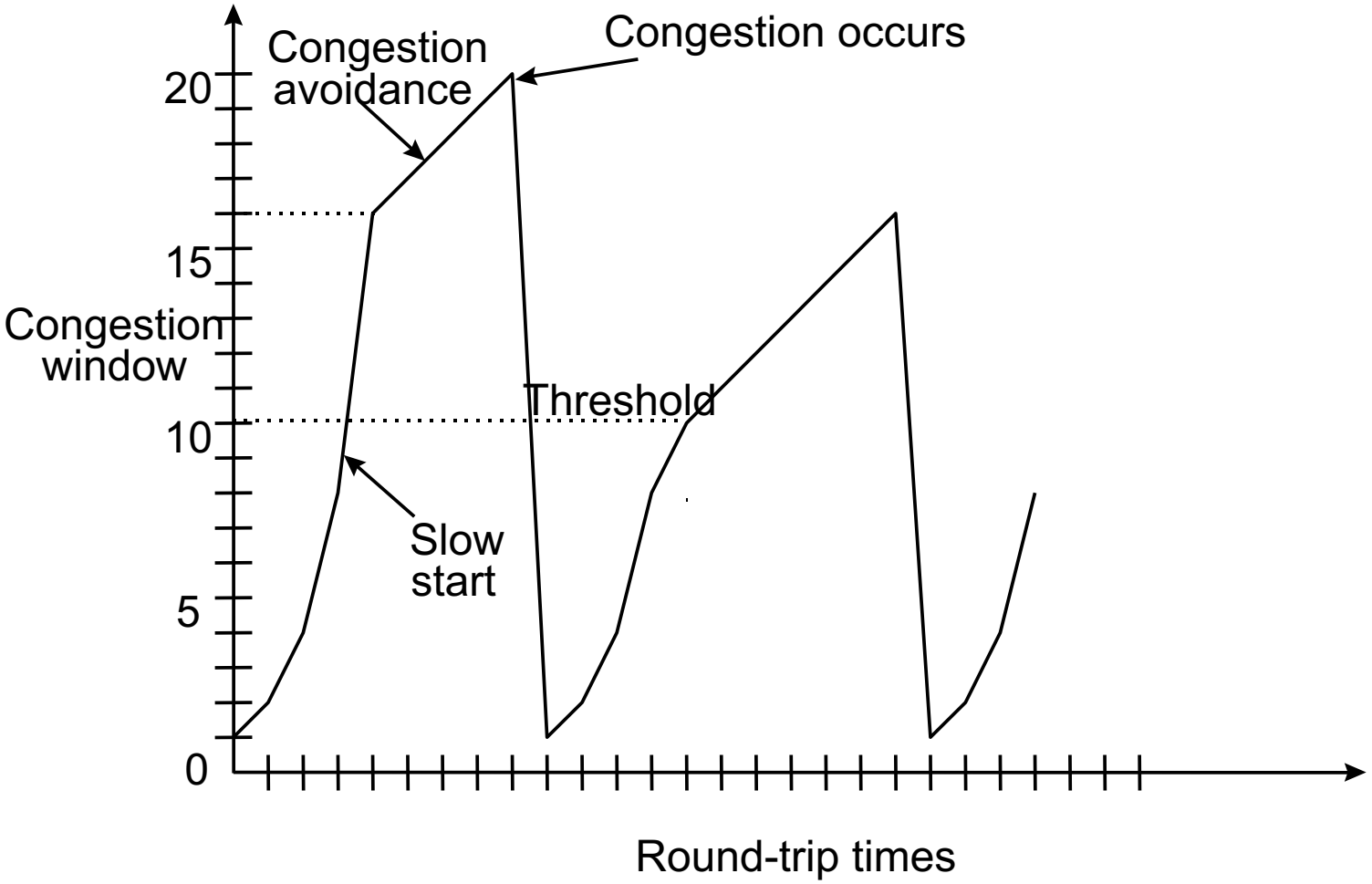
- A TCP connection is a *stream*
- Each TCP packet contains a *stream segment*
- A sequence number is associated to each byte
  - Packets have a single field for the sequence number
    - e.g., refers to the sequence number of a specific byte, according to a convention described in the RFC
- An ACK is required for each byte
  - If an ACK is not received in a certain amount of time, data is retransmitted
  - An ACK packet serves as an ACK for all bytes up to the byte indicated by the ACK's sequence number
- Receiver uses sequence numbers to correctly reorder segments and remove duplicates

TCP Flow Control

- How much can a sender send at a time?
  - The more can be sent, the more efficient the network is
    - Fewer header bytes, media contention delays, etc...
- TCP "Window"
  - With every ACK, the receiver indicates how many more bytes it is prepared to receive



Congestion Control



Protection against SYN Attacks

- Client sends SYN
- Server responds to Client with SYN-ACK cookie
  - sqn = f(src addr, src port, dest addr, dest port, rand)
  - Normal TCP response but server does not save state
- Honest client responds with ACK(sqn)
- Server checks response
  - If matches SYN-ACK, establishes connection
    - "rand" is top 5 bits of 32-bit time counter
    - Server checks client response against recent values

See http://cr.yp.to/syncookies.html

SYN Cookies (cont'd)

- Difference between server's ISN and client's ISN
  - top 5 bits: t mod 32, where t is a 32-bit time counter that increases every 64 seconds;
  - next 3 bits: an encoding of an MSS selected by the server in response to the client's MSS;
  - bottom 24 bits: a server-selected secret function of the client IP address and port number, the server IP address and port number, and t.

TCP Sequence Numbers

- Every new connection gets a new initial sequence number (ISN)
  - For both sides of the connection
  - ISNs are exchanged (jargon: streams are "synchronized") in the initial SYN handshake
  - Is this a real random number?
- TCP packets with sequence numbers outside the window are ignored
  - This makes attacks on TCP applications harder than if they used UDP
- Sequence numbers allow reconstruction of correct order of packets
- How to hijack a TCP connection?

Finding the Sequence Number

- Sniffing
- MAC address man-in-the-middle attack
- Source-routed IP packets (to setup a M.I.M. attack)
- ICMP redirects
- If the above is not possible, try to predict the initial sequence number
  - Connect yourself, examine how sequence numbers are generated (e.g., last one + 128 000)
  - Make a guess based on observations

ISN Vulnerability

- Predictable
  - Symantec Raptor Firewall 6.5 and 6.5.3, Enterprise Firewall 6.5.2 and 7.0, VelociRaptor Models 500/700/1000 and 1100/1200/1300, and Gateway Security 5110/5200/5300 generate easily predictable initial sequence numbers (ISN), which allows remote attackers to spoof connections.
    - CAN-2002-1463
  - Cisco switches and routers running IOS 12.1 and earlier produce predictable TCP Initial Sequence Numbers (ISNs), which allows remote attackers to spoof or hijack TCP connections.
    - CVE-2001-0288

TCP RST Flag

- TCP reset (RST) flag is used to abort TCP connections, usually to signify an irrecoverable error
  - Receiver deletes the connection, frees data structures
- RST messages are accepted only if they fit inside the sequence number window
  - Prevents delayed RST messages from previous connections to affect the current connection

TCP RST Attack

- Send a RST (TCP RESET flag) packet with a spoofed IP address to either side of a valid connection
  - Need to guess a sequence number inside the appropriate window
    - Or sniff traffic to know which number to use
  - The range can be guessed fairly efficiently for RST attacks
  - Sequence numbers: 32 bits
  - Window size: up to 16 bits
  - Number of guesses 16 bit address space
    - 65535 RST attempts, ~ 4 min on DSL connection
    - Faster connection or zombies, faster RST
    - This is the brute force RST attack

Hijacking a TCP Session

- Idea: all that is required to mess up someone else's TCP session is guessing or knowing the sequence numbers for their connection.
  - Only need to fall within the needed range, exact guess not needed
- Attackers needs
  - Ability to forge TCP/IP packets.
  - Initial sequence number
  - Knowledge that a TCP connection has started (but not the ability to see it)
  - When the TCP connection started
  - Ability to redirect responses to you
    OR continue the conversation without responses to you while achieving your goal
- Thought to be too hard, but exists in the wild.