Network Security

- A big open network invites many attacks
  - Authentication / confidentiality
  - Liveness (uptime of services)
  - Privacy
- Today: attacks on core network protocols

- Internet core:
  - Open network consisting of many ISPs
  - Peering agreements between ISPs provide connectivity
  - Many protocols:
    - TCP, BGP, DHCP, UDP, telnet, FTP, IMAP
    - Defined by RFC interfaces
    - Many implementations of these
    - Changing the protocol is much harder than changing implementation
- There are now many secure higher layers such as:
  - Kerberos, SSH, SSL
  - Enabled by the advent of cryptographyj
- Core network still lacks security
  - Mostly concerned with liveness
  - Hosts manage end-to-end security instead of the actual network
  - This design is successful, but important to look at attacks

Forging IP Addresses

Remote Login

- Example internet application: remote login (1980)
- Uses telnet to open a TCP connection to the login program
- Attacker can:
  - Steal the password by snooping on network
  - Modify data in flight
  - Inject false data
  - Re-direct conversation with routing
- rlogin deals with this by not using a password to authenticate
  - Destination only allows hosts in its hostfile to log in
  - Attacker could pose as the client's IP address
  - But TCP communication is both ways, so if the attacker lies about the source, then the server's replies will not go back to the attacker

TCP Setup

- On initial handshake, the client sends a sync request to the server with its sequence numbers
- Server responds with its sequence numbers and an acknowledgement of the client sequence numbers
  - Ack is important so the client knows that this is coming from the real server
  - The client only has a guarantee that its destination packet reached the correct server
    - Does not have guarantee that its receiving messages from the real server
  - Client sends back an acknowledgement of the server's sequence numbers and the data it wants to send

TCP Sequence Number Attack

- Adversary wants to simulate a connection to the server by a client C
- Attacker sends a TCP packet with src = C
- Server will send sequence numbers to C
- Attacker has to guess the sequence numbers sent to C to be able to communicate with the server
  - Could try guessing based on guessing how the server does sequence number management
  - Attacker could do an actual connection to the server to learn about what kinds of sequence numbers the server is currently using and then guess the next one
- The real client will receive the actual packet and might think that it is an old packet
  - Real client might send an abort packet to shut down this connection
  - The attacker might have to race to get there before the RST arrives
- Security risks:
  - If authentication is based on IP address, SN attacker can pretend to be a host in the trusted user list
    - Breaks rlogin; key problem was assuming that the TCP layer's sender field was legit
    - IP authentication is no longer used for remote login
  - Denial of service attack:
    - Can constantly send reset packets
    - Servers actually accept reset packets for a large window of sequence numbers so you don't even need to get the sequence number exactly right
      - You can just guess
    - These can be used to target TCP connections between BGP routers
      - Causes routers to assume link failure and affect traffic
  - Hijack existing connections
    - Can inject data into an existing connection
    - Wait for someone to log in and then hijack
- Mitigation:
  - End to end encryption (i.e. SSL as used in the next lecture)
  - ISPs can filter packets with obviously forged IP source addresses
    - Hard for complex networks

Hardening TCP against forged IP source addresses

- Need to make it harder for attacker to guess next ISN (initial sequence number)
- Can't choose them completely randomly or else you violate TCP speec
  - Need to avoid recently used sequence numbers for same host / port pair
- Can't do random increments because we don't want to wrap around very often
  - Need to keep these increments small
- Have to be careful about how we generate random numbers since random number generators can be reverse-engineered
- Main idea that is implemented in most OS is to use a random offset for each src/dest pair
  - ISN = ISN_old + SHA1(srcip, srcport, dstip, dstport, secret)
- Whole idea is to prevent attacker from being able to make an ordinary connection and using that info to try to guess the ISN for another client

Liveness

SYN Attacks

- SYN flooding: first high-profile DoS attack
  - SYN = synchronization request (initial packet)
  - Server must check client's ACKs to it sending back sequence number
    - The original implementation of TCP kept the states for these "half-open" connections
    - Kept it for minutes if client is slow / network lossy
    - Only willing to remember 50 half-open connections to avoid OOM
    - Silently ignored new connections if it already had 50 waiting

- Attack sends SYN packet with forged random IP addresses
  - Most of these will never respond and the server just waits for them to send a third packet when they never will
  - Server begins to ignore legitimate connection requests
  - Don't even need to send these that fast because servers keep half-open connections for minutes

---

- Mitigation: SYN cookies
  - Make the server stateless until it receives the third packet
  - Tricky because the half open-state was what helped ensure source IP wasn't forged by checking the packet had the right ACK
  - Use a bit of cryptography
    - Encode server-side state into sequence number
      - ISN = SN_c + timestamp || SHA1(src/dest addr/port, secret, timestamp)
      - Timestamp is coarse-grained (minutes) to ensure that clients have enough time to submit back a packet
      - ISNs are per client so attacker can't guess for a forged IP address
      - Upon receiving the third packet, the server just has to recompute to see if this is a feasible packet which should have only been known by the client that receives it legitimately
  - Successfully blunted low-rate SYN-flooding DoS attacks

Bandwidth Amplification

- Attacker's goal is to overwhelm server / link so legit traffic is discarded
- Send ICMP echo request (ping) packets to broadcast address of a network
  - Used to be that you would get an ICMP echo reply from all machines on network
  - If you fake a packet with a victim's address, then the victim gets all replies
  - If you find a subnet with 100 machines on a fast network, then you get 100x amplification on your attack
- Fixed by routers blocking directed broadcast (packets sent tobroadcast address)

- Modern variant:
  - Use DNS because with a small query, a server might send back a large respones
    - DNSSEC makes this even worse since it contains lots of signatures
    - DNS runs over UDP so source address is completely unverified
  - Hard to fix this because DNS has to respond to anyone
- Generally, can be mitigated via a DoS protection service (i.e. Akamai)
  - These have lots of available bandwidth and you tunnel requests into them first
  - They can filter out DoS attack traffic (i.e. DNS amplification responses)

Routing Protocol Attacks

- Generally, routing protocols are overly-trusting of participants
- ARP: within a single Ethernet network sending via MAC addresses of routers
  - To send an IP packet, you just need the MAC address
  - In this protocol, you broadcast a request for target's mac and anyone can listen to / send a reply with no authentication
  - You can just impersonate people
- DHCP: within a single Ethernet network sending via IP address
  - Client asks for IP address by sending a broadcast request
  - DHCP server responds with no auth
  - Adversary can imeprsonate DHCP server to new clients on the network
- BGP: Internet-wide
  - Huge routing system
  - Any BGP participant router can announce route to any IP address

    - You can announce you have a path to MIT and people will route through you
        - You can inspect / modify traffic and then forward to MIT
    - Fixes:
        - Trusted database of who is allowed to announce what IP prefixes
        - This database is a weak point
- The open Internet makes it easy for attackers to gather useful info
    - Can probe random hosts to see if they are running vulnerable software / protocols
    - Can scan entire Internet (2^32 addresses) in just 45 minutes

Improving Security

- Firewalls: partial fix but widely used
    - Issue if adversary is within firewalled network
    - Hard to determine if packet is malicious
    - Hard to authenticate src / dst fields
- Cryptographic security on top of TCP / IP can be used but is hard