Outline
- Cellular network structure
- Radio resource control
- Security threats and defenses
- Middleboxes

Cellular Network Structure
- Standards -- lots of acronyms and versions of standards, and lots of marketing nonsense from cellular service providers
  - 1G
  - 2G -- GPRS was 2.5G, EDGE was 2.75G
  - 3G
  - 3G with HSPA (High Speed Packet Access) -- marketed as 4G, but really 3G with enhancements
  - 4G LTE
- Components
  - UE = user equipment
  - eNodeB = base station
  - MME = Mobility Management Entity -- signaling messages for UE, nodeB, and GWs
  - S-GW = Serving Gateway -- router that forwards between eNodeB and P-GW
  - P-GW = Packet data network Gateway -- communicates with the outside world

Radio Resource Control (RRC)
- Determines how much power UE uses to transmit
- Three states: IDLE, FACH (forward access channel), DCH (dedicated access channel)
  - Power consumption: none, low, high
  - Data rate: none, low, high
- Transition between states based on whether you have data to send and when you last sent data
  - IDLE to DCH if you have any data to send -- takes 2 seconds to change radio to that state
○ DCH to FACH if you are idle for some time (5 seconds)
○ FACH to DCH if you have more than some amount of data queued to send
○ FACH to DCH if you are still idle for some time (12 seconds)

● Trade-off between power consumption and delay
  ○ Go to FACH or IDLE faster => save more power, but need to add delay to transition back more frequently

● Behavior of applications can really impact power/delay trade-off
  ○ If you send data every 8 seconds, you'll be in high power mode for 5 seconds, low power mode for 3 second and delay sending of next packet by 1.5 seconds
  ○ If you send bursts of data every 1 minute, you'll stay in high power mode longer initially, but you'll only transition to low power mode and then idle once => only incur delay at beginning of burst and spend less time in lower power mode
  ○ Research study examined how real application’s decisions impact power consumption and delay in practice
    ■ Pandora internet radio use to fetch full song when playback of song started, but would fetch advertising image every 60 seconds
    ■ Incurred cost of going to high power state every 60 seconds
    ■ If you fetch enough images for duration of song when song is downloaded, then you don’t need to go to high power state again and avoiding wasting power sitting in DCH and FACH state

● Example:
  ○ Assume a device wants to transmit a total of 60KB of data, and the available bandwidth is 80Kbps (i.e., it takes 1s to transmit 10KB of data).
  ○ How many Joules of energy are consumed in 90s if a device wants to transmit 20KB of data at time 0s, 30s, and 60s?
  ○ How many Joules of energy are consumed in 90s if a device wants to transmit 60KB of data at time 0s?

● Takeaway: layers make things convenient, but they abstract away details that can sometimes be useful

Network Security

● What network-related attacks must we defend against?
  ○ Unauthorized access to hosts (e.g., SSH)
    ■ Often depends on carefully crafted packets or data
  ○ Sending malicious code to hosts (e.g., viruses, worms)
  ○ Denial of Service (DoS) -- causes bottlenecks that prevent legitimate access
    ■ If many hosts are used to launch the attack (e.g., hosts in a botnet), it’s called a distributed denial of service attack (DDoS)
    ■ Often, set up lots of TCP connections but not send any data -- consumes host resources to perform handshake and maintain connection state
  ○ DNS hijacking -- resolve domain names to IP address for servers with malicious code or phishing sites
  ○ Route hijacking -- send BGP announcements for prefixes you do not own or cannot reach
  ○ Eavesdropping on data

● How do we protect against these attacks?
  ○ Encryption -- make sure data remains confidential
  ○ Authentication -- identify and assure origin of information; e.g., communicating with your bank
  ○ Middleboxes -- firewalls, intrusion prevention systems
Software on end-hosts -- anti-virus, firewall

Middleboxes

- Systems in the “middle” of the network that examine and block packets and flows
  - Middle = on the path between pairs of communicating hosts
  - Packets forced to pass through the middlebox -- based on physical network topology or using SDN
- Basic firewalls
  - Apply simple rules to decide whether to forward or block packets
  - Rules are based on fields in packet headers -- e.g., source/destination IPs, transport layer protocol, source/destination ports
  - Default rule to either forward or block if no other rules match
- Advanced firewalls
  - Maintain some state about active connections
    - E.g., current state of TCP connection -- SYN sent, SYN+ACK sent, established, FIN sent, etc.
    - Rules based on both packet headers and current state
  - Application-aware
    - Extra protection for services that should not be blocked (e.g., HTTP)
    - E.g., check if HTTP POST method is allowed
    - E.g., check if client is requesting a web page from a domain that is on a blacklist because it is known to host malware
    - Often acts as a proxy -- terminates TCP connection from client and establishes separate TCP connection to server
- Intrusion detection/prevention systems
  - Performs deep packet inspection
    - Look at payload of packet, not just headers, to decide if it should be blocked
    - Knows the format of packets for many different transport and application protocols -- HTTP, SSH, SSL/TLS, FTP, NFS, FTP, NTP, etc.
    - Cannot perform deep packet inspection on traffic that is encrypted!
  - Maintains state about active connections
    - Connection info such as src/dst IP, src/dst port, and TCP connection state
    - Reassembled payloads -- e.g., HTTP reply may be split among multiple packets, so the packets are reassembled into a single memory region that contains the entire reply
  - Uses a set of signatures (i.e., rules) to detect malicious traffic
    - Specific sequences of packets -- e.g., TCP SYN+ACK after TCP FIN
    - Keywords in payloads -- e.g., “root”
    - MD5 sum of payload -- compare against database of MD5 sums for known malware
    - Large numbers of packets to one host in a short period of time
      - DoS
        - Port scan -- look for hosts which have sockets listening on a particular port and will accept connections; host may be running an outdated version of software that has a known vulnerability
    - IDS just raises alerts, while IPS raises alerts and blocks traffic
    - Speed of signature matching can significantly impact latency and throughput
      - Want efficient pattern matching algorithms
      - Sometimes use custom hardware
      - Can be easily parallelized