Network Attacks

Vectors
- Network protocol issues or features
- Bugs in server programs
- Bugs in client programs
- Bugs in websites or web programs (but not the actual web server)

Network Protocols
70s/80s were trusting times in computing, so TCP/IP has very few security features built in

Examples:
- No default encryption - packet sniffing attacks
- No verification of addresses - "spoof" addresses
- No verification of infrastructure servers - forge responses

Common attacks
- Flood an Ethernet switch with random source MAC addresses
  - Confuses the switching algorithm so it starts broadcasting all data
  - Enables packet sniffing
  - Fix: don't allow one cable to have multiple MACs

Reflected Distributed Denial of Service Attack
- Attacker generates a packet with forged source IP address that requests a connection
  - May use many machines to generate packets

```
Attacker -> request from Bob
           Server -> request granted -> Bob
Waiting for Bob to say "ok" ("half open" state)

Server runs out of slots for half-open connections
Can't process ANY connections
```
Fix: SYN cookies eliminates need for half-open state by sending a cryptographic "cookie" w/ that info.

**ARP Cache Poisoning**
ARP is used to translate IP address \( \leftrightarrow \) MAC address
Trick into using malicious MAC address for an IP addr.

**DNS Cache Poisoning**
Same idea but for domain names & IP addr.
Fix - DNSSEC allows digital certificates for domain name \( \leftrightarrow \) IP address lookups.

**Operational Security Protections Against Protocol Attacks**
1) Keep up to date w/ systems
2) Firewalls
   - Watch for spoofed addresses, bad flag combos, etc.
   - Disable access for computers that appear to be doing common network protocol attacks
3) Protocol-based intrusion detection systems

Best practice is to use all three (as much as feasible) to have layers of defense ("defense in depth")

**Bugs in Programs**
**Three Main Types**
- Mistake in program code/logic
- Exploitable feature of program
- Malicious code (intentional)

**Mistakes in code/logic**
Very common issue w/ programs
Some languages have syntactic features which are very visually similar, but logically different

C/C++/Java examples:
- \( \text{if } (x = y) \) vs \( \text{if } (x == y) \)
- Set \( x \) to be same value as \( y \)
- Accidentally use wrong logical operator
  - \& \& AND of two items
  - \& bitwise AND
Accidentally use wrong logical operator
  & instead of two items & bitwise AND
  | instead of two items | bitwise OR

Don’t sanitize user input
  no limits on number of letters typed
  no checks on values typed

Feature gets used "unexpectedly"
  no logic error, but didn’t foresee what users will do

Malicious code
  not much to do except be careful where you get programs

Classic examples:
  Brain Virus - late 1980s
    floppy disk virus
    infects boot sector of disk
    gets run when disk put into reader
    puts itself into RAM on vulnerable system
    copies itself to new floppy when disk is put into reader

Morris Worm - Nov 2, 1988
  self-replicating code using bug in email servers
  Robert T. Morris Jr created it as a grad student
  convicted in 1990
  also tried common usernames & passwords
  caused at least 10% of systems to crash or go offline
  caused development of CERT & several security groups

Code Red Worm - mid 2001
  buffer overflow in web page distributed by default with Microsoft web servers
  allowed any code after buffer to be run as user running Microsoft web server (often Administrator)
bug had been patched one month before 1st worm, but few had applied patch

Lessons: Patching is not done as often as it should
Web servers shouldn't run as Admin
Even poorly coded worms can spread quickly

Zeus Virus/Trojan - within last 5 years
web bug - caught from visiting website with malicious code or clicking email attachments
targetted at banking credentials
use mix of key logging, browser-in-the-middle, & session hijacking
evades anti-virus & malware detection software

Lessons: Malware writers are tricky, can't count on antivirus software & patches alone

How Can We Protect Ourselves?
1) Keep up to date on patches for all software
2) Have anti-virus & malware detection software
   and keep it up to date too
3) Be careful what software you install & where you get it
4) Consider having a dedicated system for secure transactions or a "throw-away" virtual machine for random web browsing

"Botnets"
These are compromised machines w/ backdoors on them that respond to command & control networks
Used in generic Distributed Denial of Service attacks sold to go to a site