CS 370 Notes
week 8

Software Vulnerabilities

- potentially exploitable flaws in Programming
- OWASP tracks top 10's
  - ↳ injection
  - ↳ XSS (many others)
  - ...
- Categories to organize vulns include
  - → input check
  - → code / logic
  - → interactions with OS, other prog
  - → program output
- consider security at architecture time, not an afterthought
- attacks take advantage of implicit assumptions made by the programmer

User Input Violations

- <u>whitelist</u> the input to make sure it is valid.
- Code injection, command injection, SQL injection
- XSS: inject code into a webpage such that on other's instances, it is considered code

OS interaction Vulnerabilities

- typically involves poor management of memory.
- race conditions where multiple threads accessing same location
- deadlocks: both threads/programs are wait on other
- errors are inconsistent acceross executio
- environment variables are an input
  - ↳ changing PATH or IFS variables (delimiter)
- Least Privilege related vulnerabilities
  - ↳ when compromised, too many privs
- limit duration of escalated privileges
- Journaling File systems keep track of versions
- Race Conditions: two things want access to one thing. lockfiles.
- temporary files names can be guessed and linked to important files.

Buffer Overflows

- stack smashing: overwriting the important stuff on the stack to change control flow
- 3 steps
  1] injection 2] control flow 3] Execute code
- defenses by level above

compile time {
1] - memory / operation safe languages
- safe coding
- extensions or safe libraries

{2] - stack Protection (2)

run time {3] - NX (3) - ASLR (2,3) - Guard pages (2,3)

- Stack Protections
  - → stackguard: canary
  - → return addres Defender: copy ret
- guard pages differentiate memory areas.