

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

- whitelist 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 across executions
- environment variables are an input
 - ↳ changing PATH or IFS ^{delimiter} variables
- 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.

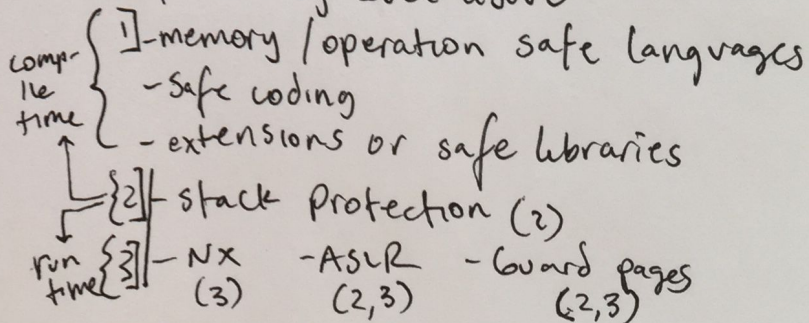
Buffer Overflows

- 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



Stack Protections

- stackguard: canary
- return address Defender: copy ret
- guard pages differentiate memory areas.