http://www.cs.sunysb.edu/~rtjohnso/teaching/cse509-sp07

Reviews: rtjohnso+509r @ cs.sunysb.edu
All else: rtjohnso@cs.sunysb.edu

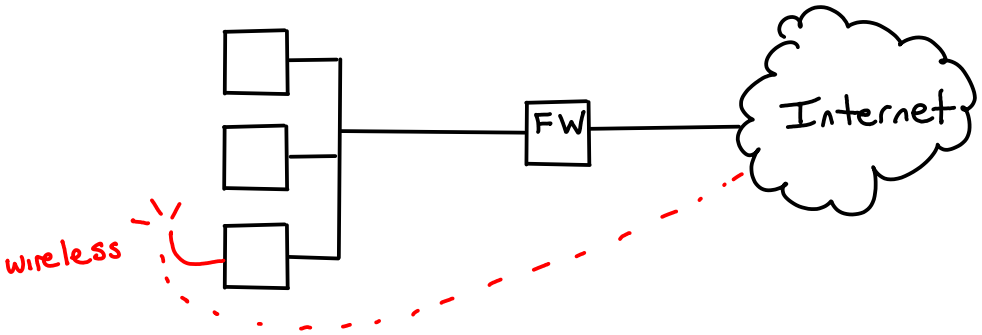
Thursday's reading is now fixed

Rule: Least shared mechanism

- code
  - downside
    - less auditing of code
    - expertise required
  - upside
    - less trust
    - limit damage
- data

Rule: Psychological Acceptability

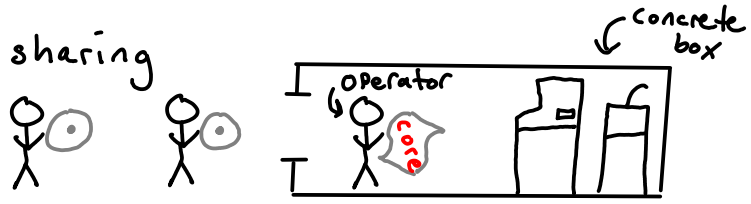
Unusable security is unused security



Early security research: Isolate

Current security research: Controlled sharing

No sharing

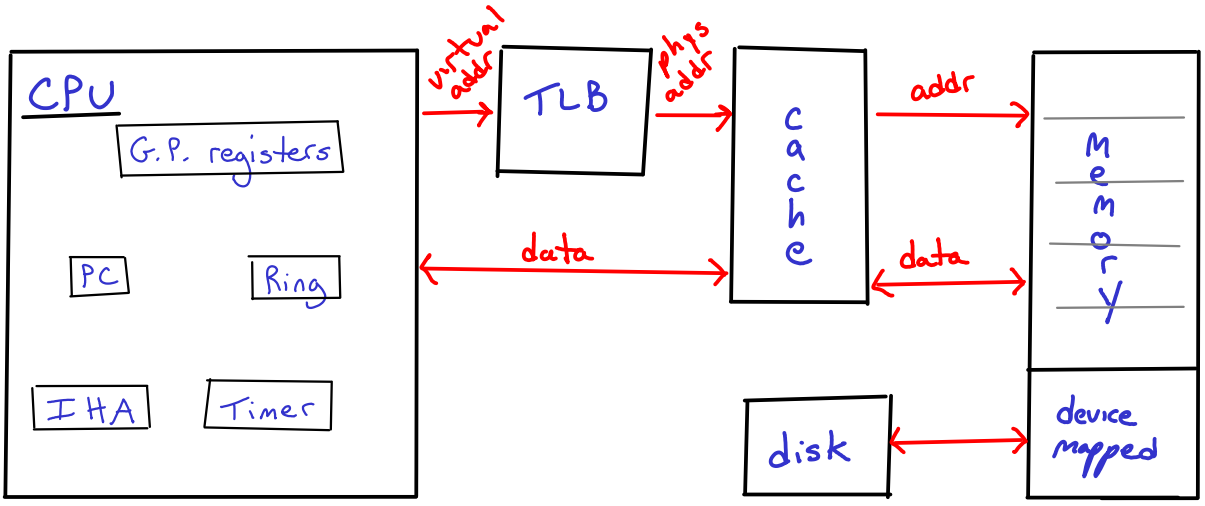


What does this achieve?

- confidentiality
- integrity
- availability

Assumptions

- operator is infallible
- operator is not malicious
- no network
- computer has no state
- enough paper
- electricity
- room is sealed
- denial of service



- never put device in the TLB

If Ring == 0: can execute any instruction
Else: can only execute "normal" instructions

TLB - maps pages

virt	phys	perm
1234	0012	R

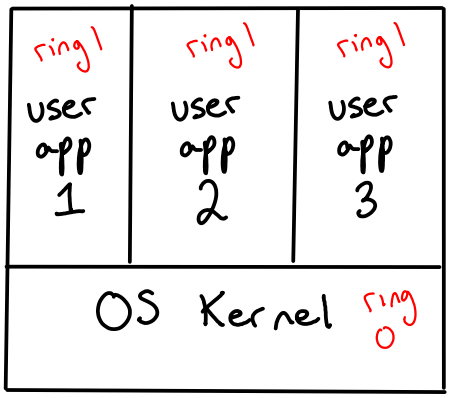
New instructions

load TLB, clear TLB } should not be available to untrusted programs

drop Priv } "normal" instruction

syscall } "normal" instruction
- Ring := 0
- Jumps to IHA

write IHA } priviledged



On system boot
- CPU is in Ring 0
- loads and executes OS

On OS Kernel boot
- sets IHA

To run program, OS Kernel
- load program into some physical pages
- places entries in TLB
- drop Priv
- jump to program

Goals

- memory insulation
  - confidentiality
  - integrity
  - availability ?
- device isolation
  - built on memory protection
- CPU insulation
  - confidentiality + integrity (save/restore full CPU state)
  - availability (Timer)

Have we achieved "full" isolation? No.
- timing leaks
- memory usage leaks

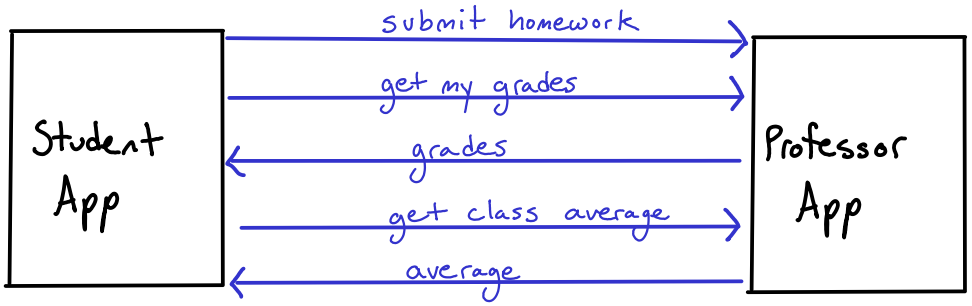
Non-interference

Very inefficient

Sharing

- OS "sendmsg" command
  - destination
  - data
- OS will prepend unforgable src address to your message
- another syscall command: recvmsg

Homework Grading System



Student trusts Professor

Student and Professor trust OS