Lecture notes for 4.1-4.2 Memory Protection

[wrapup anonymity]
[wrapup crypto]
Cryptographic hash collisions
- Book mentions that in 160-bit SHA1 attacker can find P1, P2 with same hashes in 2^63 steps, which is weaker than **expected 2^80** steps
- Why 2^80? Why not 2^159 (½ of 2^160)?
- Birthday problem: collision in sqrt(2^160) = 2^80
  - In a set of n items, we have n^2 comparisions (and possible matches), not 2n comparisons

Operating system mediation
- Control access by users/processes to resources
- Resources may be other users & processes, also files, hardware, etc.

Memory protection
- Multiprocess, multiuser systems
- Each running process can access only its own memory
  - Cannot access memory of other processes
  - Cannot access kernel memory
  - Limits damage of attack against a vulnerable process
  - Limits damage of malicious process
- Kernel can access any process' memory
- Kernel allows processes to share memory only if user permissions allow

Separation
- Keep one user (or role) separated from another user (or role)
  - Example: Military general needing to access both classified data and unclassified email
    - Convenience: wants to use single system
- Physical separation
  - Different users use different physical objects
    - General has two computers on his desk
- Temporal separation
  - Different users execute at different times
    - General reboots into secure OS
- Cryptographic separation
  - Users execute simultaneously but use encrypted computation to remain unintelligible to other users
- Logical separation
  - Users execute simultaneously, OS gives users isolated view

Granularity
- No protection
- Total isolation
- Share all or share none
- Share via OS-mediated access
- Share but limit use (some PDFs disallow printing)

Memory protection implemented via virtual memory / MMU
- Segmentation
- Paging

Paged virtual memory
- Every process views memory as contiguous, often larger than physical memory
  - Usually 2^32 or 2^64 addresses
- Operating system maps virtual pages onto physical memory frames
- Each process has own mapping
- OS will not map a virtual page for process A to a physical page for process B unless memory is shared
- Process A cannot access process B's memory because it has no way to name the memory
- Page tables managed by OS
- Processor MMU uses page tables to resolve virtual addresses to physical addresses
- Used by: Windows, OS X, Linux

Page protection bits
- RWX bits on memory pages
- Limits type of access to addressable memory
- Non-exec stacks help prevent basic code injection via stack buffer overflow

Operating systems without memory protection
- DOS
- Old mainframe systems
- Some embedded systems

Kernel fence
- Kernel resides in a portion of each process' virtual address space
- 32-bit Linux 2.6: Lower 3 GB for process, top 1 GB is kernel
  - True for each process
- Processes can cross fence only in limited ways
  - Corresponds to x86 privilege ring transition
- Windows, OS X similar
- DOS had no fence (no virtual memory...)
  - Any process could alter DOS
  - Viruses spread by hooking DOS interrupt handlers via direct kernel alteration