OS Basics for Security

- Goals: restrict access to register, RAM & limit processing time.
- How to restrict access to memory:
  o Segment Register
    1. 0 to 1000 process A and 1000 to 2000 process P
    2. Real CPU don’t work this way
  o We use virtual memory: TLB – virtual address table (translation look aide buffer)
    1. Address are divided to 2 parts, first part address translated, 2nd is offset
    2. 32 bit address 20—which page || 12 which offset of the page
    3. The offset have to account for the page size, \(4^{12} = 4096\) (average page size)
- How to load OS into the virtual memory at boot time
  o OS has access to all of memory
  o At start x86 machines start in real mode – no TLB: has access to all memory as physical address
  o TLB set to default entries
  o Address entries for itself in the TLB
  o Switch into “Protected Mode”
- TLB manipulation is privileged
  o Privileged register indicate if TLB instructions and Interrupt Handler instructions are allowed.
  o OS has privilege ==1.
  o OS can set privilege =0;
- How to run an Application
  o Idea 1
    1. Clear all entries from TLB
    2. Add current application entries to the TLB
    3. Set clock interrupt 100 mil second
    4. Set Privileged bit = 0
  o Interrupt restore private bit = 1
  o Ok loads few entries in TLB for its reload
  • Support for read only page for the OS is needed, so application can’t change it
  o In real systems PID is used and every application is in the TLB as well as the OS
  o OS can’t be read or access by applications.
  o Applications use Interrupts in order to interact with I/O like the hard drive