CS 458/658 - Computer Security and Privacy **Fall 2018**

Lecture 4: September 20, 2018

Lecturer: Ian Goldberg *Notes By: Harsh Mistry*

4.1 Operating Systems

- An operating system allows different users to access different resources in a "shared way"
- The operating system needs to control this sharing and provide an interface to allow this access
- Identification and authentication are required for this access control

4.1.1 Protected Objects

- Memory
- Data
- CPU
- Programs
- I/I devices
- Networks
- OS

4.1.2 Separation

- Physical Separation - Use different physical resources for different users
- Temporal Separation - Execute different users' programs at different times
- Logical Separation - User is given the impression that no other users exists
- Cryptographic Separation - Encrypt data and make it unintelligible to outsiders

4.1.3 Sharing

- Sometimes users do want to share resources
- As a result, OS should allow "flexible" sharing"

4.1.4 Memory and Address protection

- Prevent one program from corrupting other programs or data, operating system and maybe itself
- Often, the OS can exploit hardware support for this protection, so its cheap
- Memory protection is part of translation from virtual to physical addresses
- Protection Techniques
  - Fence Register
    - Exception if memory access below address in fence register
    - Protects operating system from user programs
    - Single-user OS only
  - Base/Bounds register pair
    - Exception if memory access below/above address in base/bounds register
    - Different values for each user program
    - Maintained by operating system during context switch
    - Limited flexibility
  - Tagged architecture
    - Each memory word has one or more extra bits that identify access rights to word
    - Very flexible
    - Large overhead
    - Difficult to port OS from/to other hardware architectures
  - Segmentation
  - Paging

4.1.4.1 Segmentation

- Each program has multiple address spaces
- Different segments for code, data, and stack
- Virtual addresses consist of two parts: Segment Name and offset within Segment
- OS keeps mapping from segment name to its base physical address in Segment Table
- OS can (transparently) relocate or resize segments and share them between processes
- Segment table also keeps protection attributes
- Advantages
  - Each address reference is checked for protection by hardware
  - Many different classes of data items can be assigned different levels of protection
  - Users can share access to a segment, with potentially different access rights
  - Users cannot access an unpermitted segment
- Disadvantages
  - External fragmentation
  - Dynamic length of segments requires costly out-of-bounds check for generated physical addresses
  - Segment names are difficult to implement efficiently

4.1.4.2 Paging

- Program (i.e., virtual address space) is divided into equal-sized chunks (pages)
- Physical memory is divided into equal-sized chunks (frames)
- Frame size equals page size
- Virtual addresses consist of two parts: Page number and offset within page
- OS keeps mapping from page # to its base physical address in Page Table
- Page table also keeps memory protection attributes
- Advantages
  - Each address reference is checked for protection by hardware
  - Users can share access to a page, with potentially different access rights
  - Users cannot access an unpermitted page
  - Unpopular pages can be moved to disk to free memory
- Disadvantages
  - Internal fragmentation
  - Assigning different levels of protection to different classes of data items not feasible

4.1.4.3 x86

- x86 architecture has both segmentation and paging
- Memory protection bits indicate no access, read/write access or read-only access
- Most processors also include NX (No eXecute) bit, forbidding execution of instructions stored in page

4.2 Access Control

Memory is only one of many objects for which OS has to run access control. In general, access control has three goals :

- Check every access: Else OS might fail to notice that access has been revoked
- Enforce least privilege: Grant program access only to smallest number of objects required to perform a task
- Verify acceptable use: Limit types of activity that can be performed on an object

4.2.1 Access Control Structures

There are 4 different types of control structures

- Access control matrix
- Access control lists
- Capabilities
- Role-based access control

4.2.1.1 Access Control Matrix

- Set of protected objects : O
- Set of subjects : S
- Set of rights : R
- Access control matrix consists of entries $a[s, o]$, where $s \in S, o \in O$ and $a[s, o] \subseteq R$
- Access control matrix is rarely implemented as a matrix, instead an access control matric is typically implemented as
  - A set of access control lists
  - A set of capabilities
  - or some combination