

Operating System Security

CSH5 Chapter 24
“Operating System Security”
William Stallings

1

Copyright © 2011 M. E. Kabay. All rights reserved.

Topics

- Information Protection and Security
- Requirements for Operating System Security
- Protection Mechanisms
- File Sharing
- Trusted Systems
- Windows 2000 Security
- Windows 7 Security*



* Not in chapter:
added by M. E. Kabay

2

Copyright © 2011 M. E. Kabay. All rights reserved.

Information Protection & Security (1)

Overall protection policies

- No sharing
 - ❑ Every process completely isolated
 - ❑ Virtualization illustrates this approach
- Sharing originals of program or data files
 - ❑ Read-only access to program
 - ❑ Sharing data requires locking mechanisms
- Confined, or memoryless, subsystems
 - ❑ No transfer of protected information across boundaries
 - ❑ E.g., server and client are partitioned from each other
- Controlled information dissemination
 - ❑ Security classes for data and users determine access
 - ❑ Widely used



3

Copyright © 2011 M. E. Kabay. All rights reserved.

Information Protection & Security (2)

- Operating security concerns grouped
- Access controls
 - ❑ Regulating user access to total system, subsystems, data
 - ❑ Regulating process access to resources & objects in system
- Information flow control
 - ❑ Within system &
 - ❑ To users
- Certification
 - ❑ Proving that access & flow control perform to specification
 - ❑ Demonstrating that measures actually enforce data protection and security policies

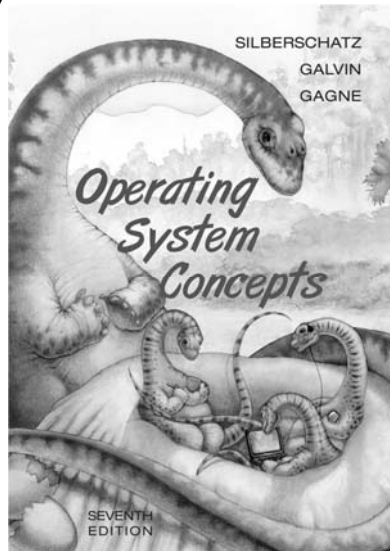


4

Copyright © 2011 M. E. Kabay. All rights reserved.

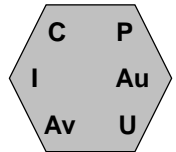
Requirements for Operating System Security

- Requirements
- Computer System Assets
- Design Principles



Requirements

- Confidentiality
 - Restrict access to authorized parties
 - Prevent disclosure even of *existence* of data
- Integrity
 - Control over who can make which changes to what system assets
 - RWALX (read, write, append, lock, execute) including save, delete, changing
- Availability
 - Timely access to resources with authorization
- Authenticity
 - Verify identity of user



Computer System Assets



- Hardware
 - Accidental & deliberate damage or alteration (e.g., switches, hardware settings)
 - Theft
- Software
 - Availability – deletion, disabling
 - Corruption – changing functionality (malware, accidental write)
 - Control – preventing unauthorized copying
- Data
 - Unauthorized access for reading or writing (especially personally identifiable information)
 - Data integrity & data destruction
 - Inference (data mining and data correlation)



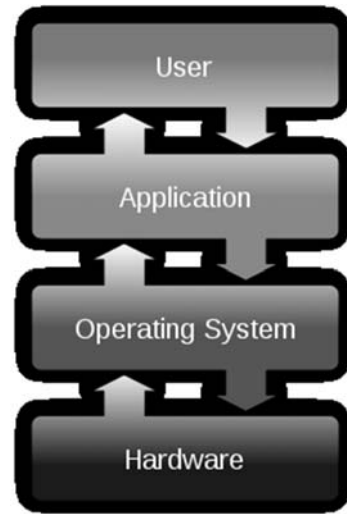
Design Principles

- Least privilege
 - Default no access; explicit granting of permissions
- Economy of mechanisms
 - Small, simple security tools
 - Include in initial design, not as add-ons
- Acceptability
 - Meet functional requirements AND keep overhead to minimum
 - Do not interfere *unreasonably* with operations
- Complete mediation
 - All access must be checked by security processes
- Open design
 - Do not depend on *secrecy* of the design or implementation (Kerkhoffs' Principle)
 - Allow for expert review, open discussion



Protection Mechanisms

- Overview
- Protection of Memory
- User-Oriented Access Control
- Data-Oriented Access Control
- Protection Based on an OS Mode



Overview of Protection Mechanisms

- Resources being shared in multiprogramming environments
 - ❑ CPU, Memory, I/O devices, Programs, Data
- Spectrum of OS protections
 - ❑ No protection – run sensitive procedures at different times
 - ❑ Isolation – all processes completely separate, with no shared resources
 - ❑ Share all or share nothing – public or private
 - ❑ Share via access limitation – every access verified for specific user & specific object
 - ❑ Share via dynamic capabilities – allow dynamic creation of sharing rights for objects
 - ❑ Limit use of an object – functional limitations (read, write, print, statistical measures vs individual data)



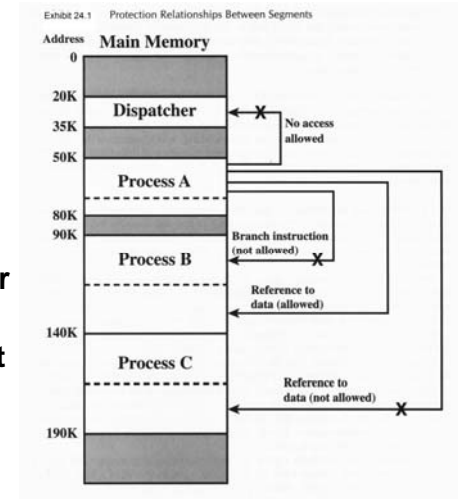
Protection of Memory (1)

- Protection main memory crucial for multiprogramming environment
 - ❑ Processes must not overwrite each other's data
 - ❑ Must not read private data
- Virtual memory supports protection
 - ❑ Memory segmentation or paging basis for defining objects to be protected
 - ❑ Segmentation allows applications to declare segments as sharable or nonsharable
 - ❑ Segments have defined length in addition to base address: can enforce bounds restrictions
 - ❑ Paging more difficult because memory management data not available to programmer



Protection of Memory (2)

- Fig 24.1 (➔) shows how OS can control access in paged memory
- Hardware can implement memory protection
 - ❑ E.g., IBM S370 under OS/390
 - ❑ Every page has 7-bit storage control key
 - ❑ OS checks key for allowed operations



User-Oriented Access Control

- Distinguish between
 - ❑ **Identification**: provision of an identifier (e.g., userID)
 - ❑ **Authentication**: ascertaining binding between identifier & user of identifier
- User logon is I&A
 - ❑ Identification (provide userID) &
 - ❑ Authentication (provide some other bound information – see later chapters on I&A)
- Once process(es) established for user, can use data-oriented access control

I&A



Data-Oriented Access Control (1)

- Assign access profile to userID once logon complete on specific system
- OS can restrict / grant access to objects on system as function of profile
- **Access matrix model** includes
 - ❑ **Subject** (e.g., user ID that creates a process and conveys its privileges to the process)
 - ❑ **Object** (anything definable to which access can be controlled; e.g., files, records, fields, programs, hardware, memory structures, ...)
 - ❑ **Access right** (how specific subject can interact with particular object; e.g., RWALX)



Data-Oriented Access Control (2)

- Most frequent implementation of data-oriented access control uses **access control lists (ACLs)**
 - ❑ For each object, list users & allowed access modes
 - ❑ Can specify **groups** of users
 - ❑ Usually includes **default** mode for unlisted users
- **Capability tickets** apply to users
 - ❑ For each user, list authorized objects & access modes
 - ❑ Users may lend or give them to other users (delegation)
 - ❑ But dispersal increases need for authenticity of the tickets
 - ❑ Therefore OS often holds ticket in central store



Protection Based on an OS Mode (1)

- Processors support multiple (at least 2) modes of operation
 - ❑ **More privileged mode**
 - ✓ **System, control, or kernel*** mode
 - ✓ Permits R/W of control registers, direct I/O, memory management, process control
 - ❑ **Less privileged mode**
 - ✓ **User mode**
 - ✓ Normal mode for user processes

*Kernel is part of OS w/ critical functions



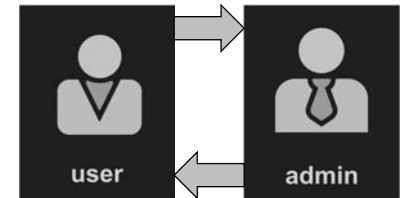
Protection Based on an OS Mode (2)

Exhibit 24.3 Typical Kernel Mode Operating System Functions



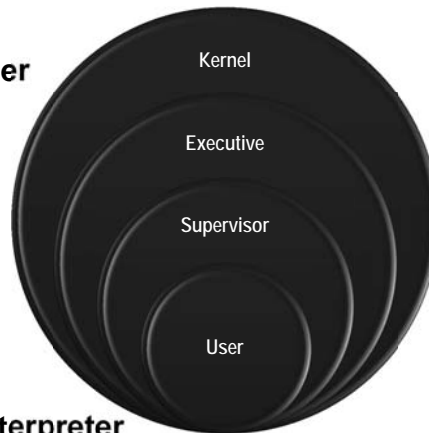
Protection Based on an OS Mode (3)

- How does the processor know which mode to use?
 - ❑ Bit in process control block (or equivalent)
- How is the mode changed?
 - ❑ Execute instruction to flip mode bit
 - ❑ Switch into privileged mode upon entering system routine
 - ❑ Switch into user mode at end of system routine



Ring-Protection Structure

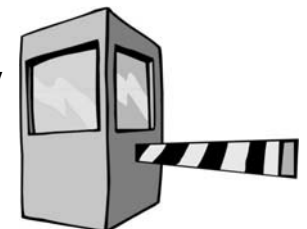
- Many OSs implement a ring-structure for privileges
- Process may access
 - ❑ Data in same ring or lesser
 - ❑ Services in same ring or higher
- Example from VAX VMS
 - ❑ Kernel: memory management, interrupt handling, I/O
 - ❑ Executive: file, record management
 - ❑ Supervisor: command interpreter
 - ❑ User: normal program execution



File Sharing: Access Rights

- Can control access to range of file information and functions; e.g.,
 - ❑ None: not even knowledge of existence
 - ❑ Knowledge: file exists, owner
 - ❑ Execution: run program
 - ❑ Read: input from file
 - ❑ Append: output to end of file
 - ❑ Update: modify existing records only*
 - ❑ Write: add, change, delete records**
 - ❑ Change protection: usually owner only
 - ❑ Delete: destroy file
 - ❑ Lock: flag for concurrency control***

NOTES:
 *In text, author does not limit "update" to this function only.
 ** Author refers to write function as part of "update"
 ***Not mentioned in text



Trusted Systems: Multilevel Security

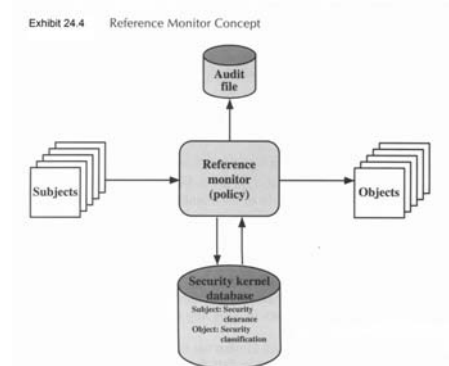
- Categories of security requirements; e.g.,
 - ❑ Top Secret, Secret, Confidential, Unclassified
 - ❑ Corporate-officers-only, Company-confidential, General-release
- Fundamental: higher-classification data must not be released to lower-classification group without reclassification
- Rules
 - ❑ No read up (simple security property): read only at equal or lower level
 - ❑ No write down (*-property): write only at equal or higher level



See CSH5 Chapter 9 for more details on security models

Trusted Systems: Reference Monitor

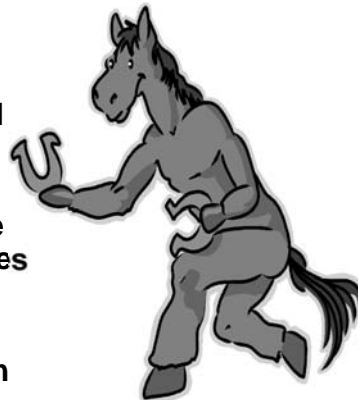
- Enforces security rules
- Properties
 - ❑ Complete mediation: all accesses
 - ❑ Isolation: protected against unauthorized modifications
 - ❑ Verifiability: provable correctness
- Computer Security Center of National Security Agency established to help evaluate and certify trusted systems



See CSH5 Chapter 51 for more details on trusted systems evaluation

Trusted Systems: Trojan Horse Defense

- Trojan horse programs attempt to subvert security by tricking higher-privilege user into executing harmful code
- Some Trojans such as keyloggers attempt to store privileged information in files that can be accessed by unprivileged users
- But a reference monitor can prevent write-down (*-property) and thus stymie the Trojan data collection



Windows 2000 (& Later) Security

- Introduction
- Access-Control Scheme
- Access Token
- Security Descriptors



Introduction to W2K Security

- Windows 2000 (W2K) OS
 - ❑ Released Feb 2000
 - ❑ Successor to NT
 - ❑ Followed by XP, Server 2003, Vista, 7
- Access control uniformly applied
 - ❑ Processes, files, flags, windows....
- Uses 2 entities
 - ❑ Access token for each process
 - ❑ Security descriptor for each object



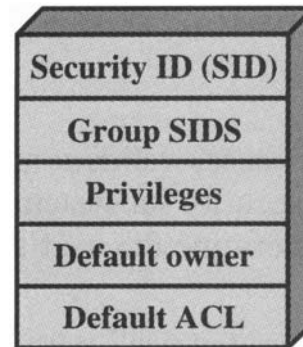
W2K Access-Control Scheme

- User logs on with userID/ password
- User process has access token created
 - ❑ SecurityID (SID)
 - ❑ Child processes inherit SID
- Functions of access token
 - ❑ Consolidates all security information for fast validation
 - ❑ Lets process modify own security parameters without interfering with other processes
- Security descriptor
 - ❑ Associated with each object
 - ❑ Includes ACL



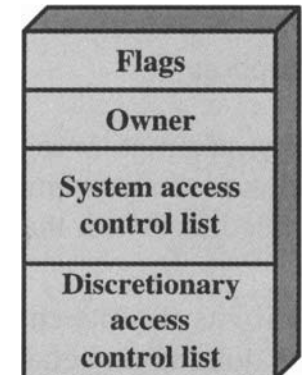
W2K Access Token

- SecurityID: unique identifier
- Group SIDs: list of groups to which user belongs
- Privileges: list of services available
- Default owner: who owns a new object created by user
- Default ACL: access control list available by default to new object created by this user



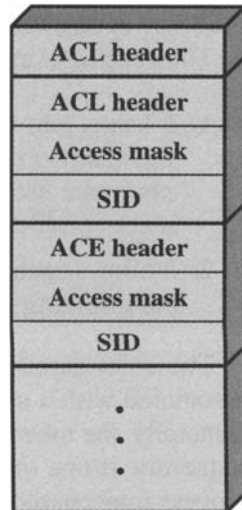
W2K Security Descriptors

- Flags: what's in the SD
- Owner: SID
- System ACL (SACL): which operations generate audit records
- Discretionary ACL (DACL): which users and groups can access object for which operations
- Any new process can receive SID of creator



W2K ACLs

- Overall header defines structure
- Access Control Entries (ACEs)
 - Specific SID (or group SID)
 - Access mask defining rights
- Object manager reads SID & scans object's DACL for match
 - Match shows right for process to access object



W2K Access Mask Standard Access Types

- Access mask bits define allowable modes
 - Synchronize: can make object part of wait
 - Write_owner: modify owner of object
 - Write_DAC: modify protection
 - Read_control: get the security data for object
 - Delete: destroy object

See Exhibit 24.7 for additional bits in Access Mask



W2K Access Mask Generic Access Bits

- Define general-purpose access modes
- Can be applied to any object
- Types are
 - Generic_all: allow all access
 - Generic_execute: run code
 - Generic_write: any form of output to object
 - Generic_read: input from object



W2K Access Mask Special-Purpose Bits

- Access_System_Security
 - Allows process to modify audit & alarm control
 - Access token must have appropriate privilege enabled
- Maximum_allowed
 - Alters algorithm for granting privilege to user
 - If off, security monitor scans entire list to locate privilege requested or end of list
 - If on, monitor limits privilege to a defined maximum



W2K Options for Access

- Attempt to open object for all possible accesses
 - But may be denied
 - Even though enough access available for needs
- Open object with specific access every time required
 - Reliably get access
 - Increase overhead due to extra table entries
- Open object with maximal access allowed for object
 - But may grant more than needed
 - Can lead to security issues with bad code



33

Copyright © 2011 M. E. Kabay. All rights reserved.

Application-Level Usage of W2K Security

- Applications can apply W2K security to specific objects; e.g.,
 - Database server can attach descriptors to elements of DB
 - Add special DB-specific functions such as JOIN
- OS checks access rights as usual



Windows 7 Security

- ASLR
- DEP
- BitLocker-to-Go
- IE8
- UAC
- Crypto improvements



made by microsoft

Windows 7 Security (1)

- ASLR
 - Address Space Layout Randomization
 - Unpredictable location of DLLs in RAM
 - Much harder for malware to target code
- DEP
 - Data Execution Prevention
 - Restrictions on buffer overflow attacks
- BitLocker-to-Go
 - Encryption for any kind of data storage
 - Includes removable media



Windows 7

36

Copyright © 2011 M. E. Kabay. All rights reserved.

[ref 1]

Windows 7 Security (2)



- Internet Explorer 8 SmartScreen
 - ❑ Anti-phishing / anti-malware feature
 - ❑ Blocks known bad sites
 - ❑ Highlights actual URL of links in address bar (warning against phishing)[ref 1]
- UAC
 - ❑ User Account Control
 - ❑ Distinguish between admin and normal user
 - ❑ Set domain environment to “Always notify” so “users will be prompted to input their passwords to perform high-risk administrative actions”[ref 2]



Windows 7 Security (3)

- Crypto improvements
 - ❑ Swap file easily encrypted
 - ✓ XP and earlier allowed swap file erasure
 - ✓ But could add 10 minutes to shutdown
 - ✓ W7 allows swap-file encryption
 - ❑ All modern encryption methods supported
 - ✓ Suite B: AES, ECDSA (Elliptic Curve Digital Signature Algorithm), ECDH (Elliptic Curve Diffie-Hellman, SHA2)
 - ✓ See < <http://tinyurl.com/3xs28uz> >
 - ❑ Encrypting File System improved
 - ✓ Control user actions – keylengths, ciphers, force backups of keys

Windows 7 Security



References:

- [1] Bradley, T. (2009). “Pros and Cons of Windows 7 Security.” *PCWorld* (Nov 23, 2009). < <http://tinyurl.com/yfarf6z> >
- [2] Grimes, R. A. (2010). “Expert’s Guide to Windows 7 Security.” *InfoWorld* (Sep 30, 2010). < <http://tinyurl.com/37zkskp> >



DISCUSSION