Security in Computing & Information Technology

Lecture 6

Operating System Security

Lecture Schedule

Foundations

1. Introduction
2. Vulnerabilities, Threats, Attacks

Basic mechanisms

3. Security mechanisms, Elementary cryptography
4. Authentication
5. Access control

Major computing security areas

6. Operating systems
7. Databases
8. Networks
9. Web
10. Mobile computing

Applications

11. Privacy
12. Internet banking

Lecture Topics

- Security issues in OSs
- OS security mechanisms
- Security in ordinary OSs

Operating System (OS)

- A collection of system programs which manages the operation of a computer
  - Controls the resources of a computer
    - Time (CPU, disk scheduling)
    - Space (main & secondary storage)
    - Process synchronisation
      Process: running instance of a program
      Separate processes can run the same program code
    - Accounting information
  - Provides a base on which applications can be built
- Shields the user/programmer from the intricacies of the hardware
- Presents a user-friendly interface
  - Execution environment file manipulation, I/O handling …
  - Error detection and handling

OS Functions

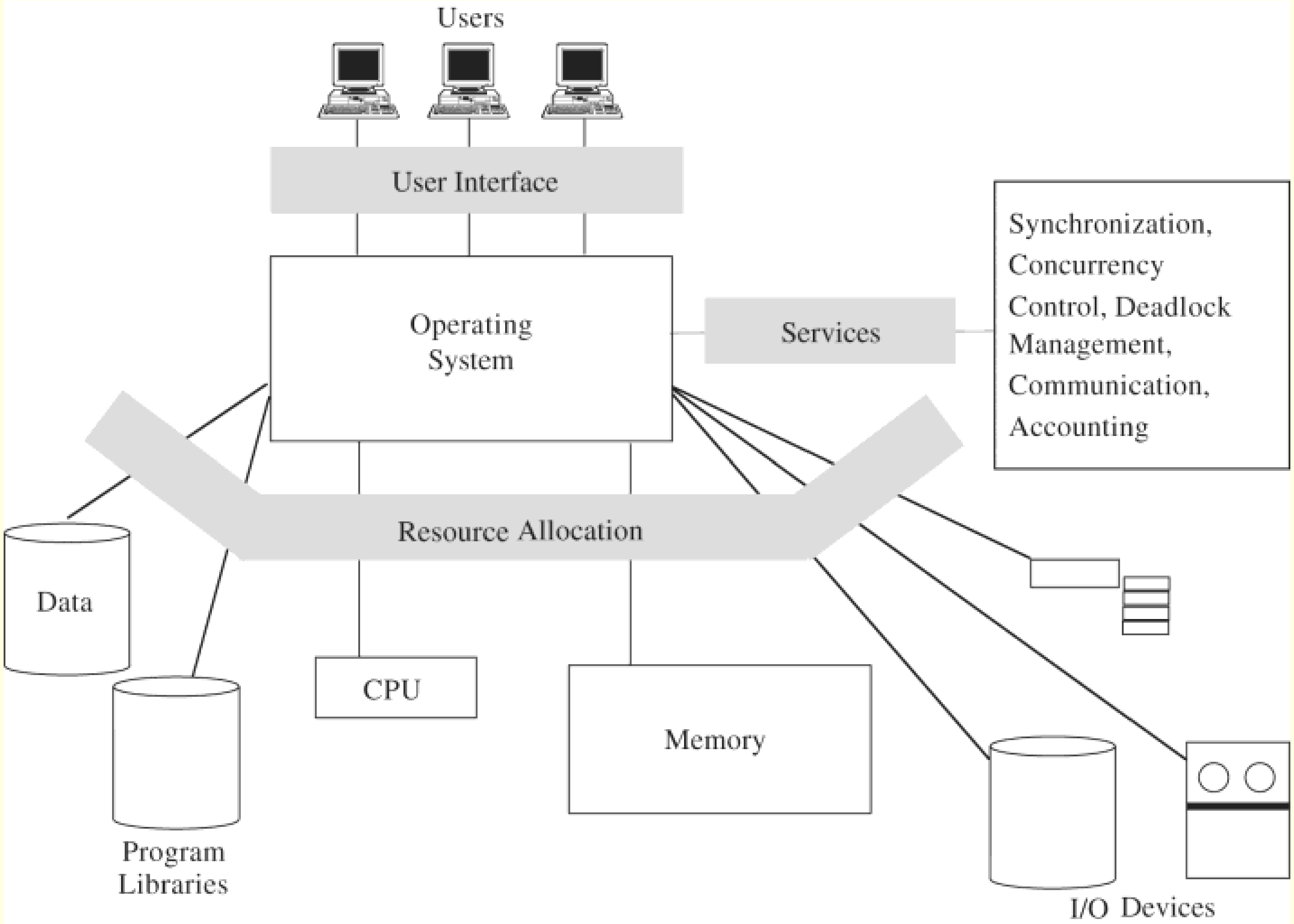


Image source: Pfleeger & Pfleeger, Security in Computing

OS Evolution

Major phases	Technical innovations	Possible attackers	Example OS
Open shop	The idea of OS	Anyone	IBM 201
Batch processing	Tape batching, First-in-first-out scheduling	Machine operators	Mainframe computing
Multiprogramming	Processor multiplexing, resource scheduling	Tasks on the same computer	Unix, VMS
Distributed systems	Networked resources	Users on the same network	Unix, Windows
Internet (cloud)	Virtualization	Users connected to the Internet	Azure, Chrome OS
Pervasive computing	Resource constrained devices	Users connected to the Internet	iOS, Android

Basic OS Security Features

- Authentication of users
- Protection of resources
  - Hardware
    - Memory
    - Sharable I/O devices (e.g. disks)
    - Serially re-usable I/O devices (e.g. printers)
    - Network connections
  - Software
    - Sharable programs and procedures
    - Sharable data
    - Interprocess communication
- Enforcing policies
  - Allocation and access control to general objects
  - Enforcement of sharing
  - Guarantee of fair service

Basic OS Security Principles

- Least privilege
  - Assign the least amount of privileges needed to complete the task
- Economy of mechanism
  - Small and simple mechanisms reduce opportunities for attacks
- Open design
  - Security should not depend on obscurity of the mechanism
- Complete mediation
  - Every access has to be checked
- Permission based
  - Fail-safe defaults (default is denial of access)
- Separation of privilege
  - Program divided into parts, each part runs with least privileges
- Least common mechanism
  - Programs cannot corrupt each other's state
- Easy to use
  - User's security expectations should match the mechanisms available

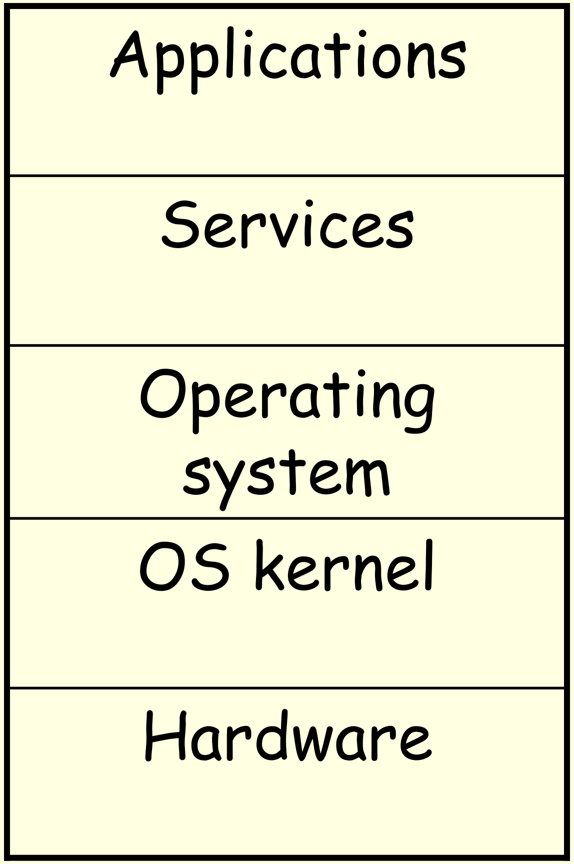
Protection Methods

- Protection based on OS
  - Many CPUs provide hardware support for user mode and system mode
  - Allows some quick access control decisions (e.g. done by hardware)
- User-oriented access control
  - User profile assigned after authentication
  - Used e.g. to grant access to a system
- Data-oriented access control
  - Access control considers both data accessed and user identity
  - Used e.g. to access databases

OS Security Methods

- Separation
  - Physical
    - Different processes use different resources
  - Temporal
    - Different processes run at different times
  - Logical
    - Processes do not see anything related to other processes (sandboxing)
  - Cryptographic
    - Processes conceal their internal working in a way that makes them incomprehensible for others
- Control of sharing
  - Allow sharing without security compromise
  - Granularity of objects & control

Protection Layers



Applications

Services

Operating system

OS kernel

Hardware

- Security has to be provided at each layer
- Each layer should have one or more security mechanisms
- Security services can be (and are) shared by different processes in a layer
- Sharing security services can undermine protection between sharing entities

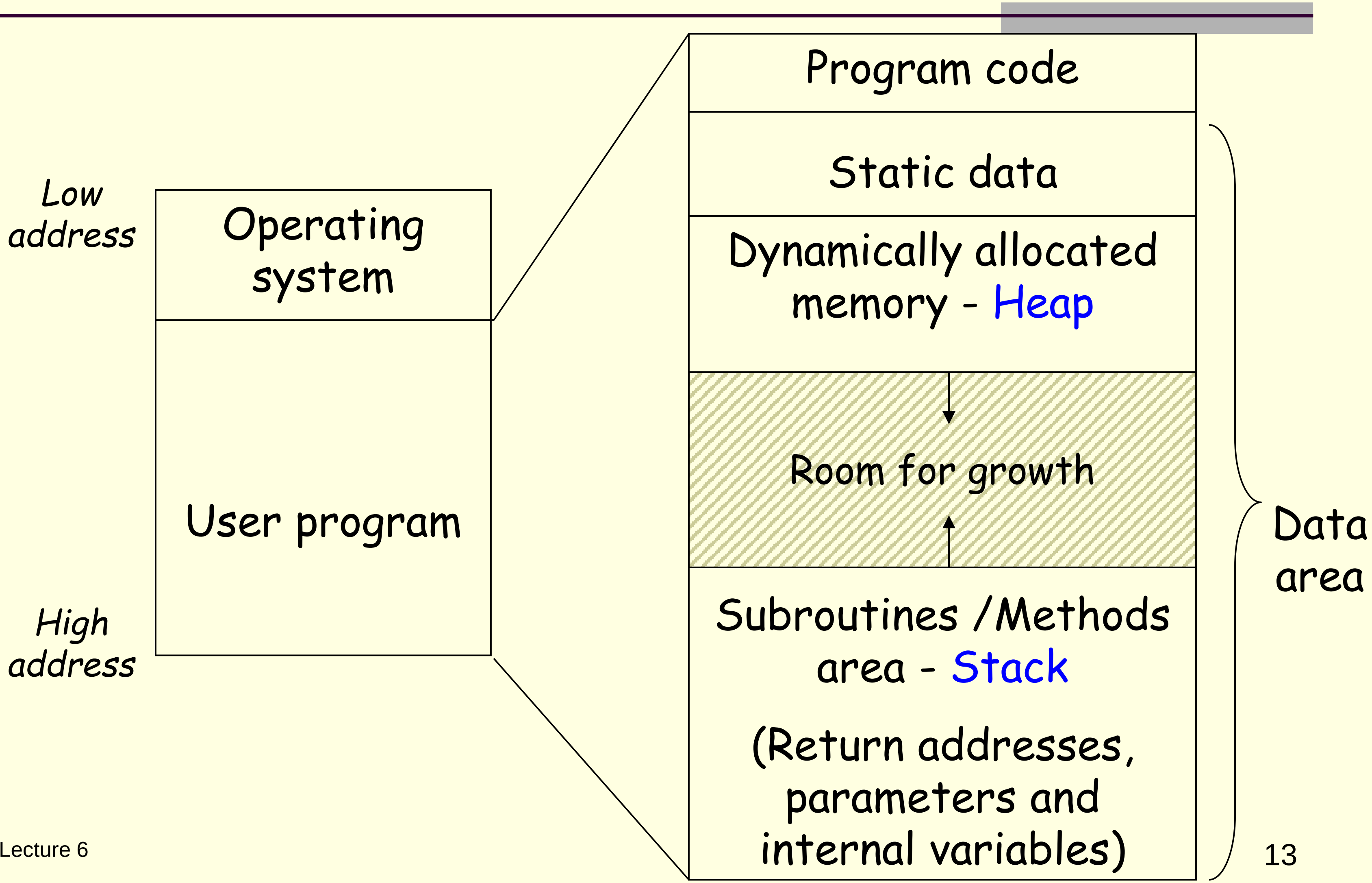
Memory Organisation (1)

- Each process has its own memory space not accessible by others
- The memory space for shared access is separate
- Access rights are associated with each part of the memory
  - Access rights depend on the OS
  - Illegal memory accesses raise exception

    E.g. Windows XP: `STATUS_ACCESS_VIOLATION`

    `(0xFC: ATTEMPTED_EXECUTE_OF_NOEXECUTE_MEMORY )`
- Virtual memory
  - Memory larger than the available physical main memory, some part of it is stored on disk
  - Memory references have to be translated to physical addresses
  - Swapping: bringing in (or out) chunks of memory (pages)

Memory Organisation (2)



Low address

Operating system

User program

High address

Program code

Static data

Dynamically allocated memory - Heap

Room for growth

Subroutines /Methods area - Stack

(Return addresses, parameters and internal variables)

Data area

Memory Protection

- Memory management
  - Processes should not be able to read/write memory belonging to another process
- Protection levels
  - System (OS) area: accessible only by the OS
  - User area: accessible by user programs & OS
  - Some OSs also have sublevels of the above two
- Protection methods
  - Segmentation
    - Memory is divided into segments
    - Each segment has its access rights
  - Paging
    - Virtual memory with fixed segment size (one page)
  - Capability-based addressing
    - Access to objects is controlled

      Programs may execute in the same memory space
    - The concept is used by object-oriented systems

      E.g. Java Virtual Machine

Memory Protection Limitations

- Memory is protected only while in use (allocated)
  - Once memory is released, the information may be available
    - Example: tar files on Solaris 2.0 contained segments of /etc/passwd: the tar utility was looking up some user information before doing the real work
- Disk will contain virtual memory pages even after the computer is turned off

Files and File Systems

- File
  - User-level unit of storage on external media (e.g. disk)
    - Identified by its name
  - Resides in permanent storage
- File system
  - Maintains files (create, delete)
  - Organises files
  - Provides tools to manipulate content
- Directory (folder)
  - Container of files
  - Can hold other directories
  - Typically it is a file itself

File Information

- File systems maintain various information (aka properties) about files
  - The information maintained depends on the actual OS
- Typical properties
  - Name
  - Date of last modification
  - Type
    - E.g. text, binary, executable, …
    - Usually indicated by the extension in the name (e.g. file.java, file.txt, file.exe …)
  - Size
  - Attributes
    - Hidden, compressed, …

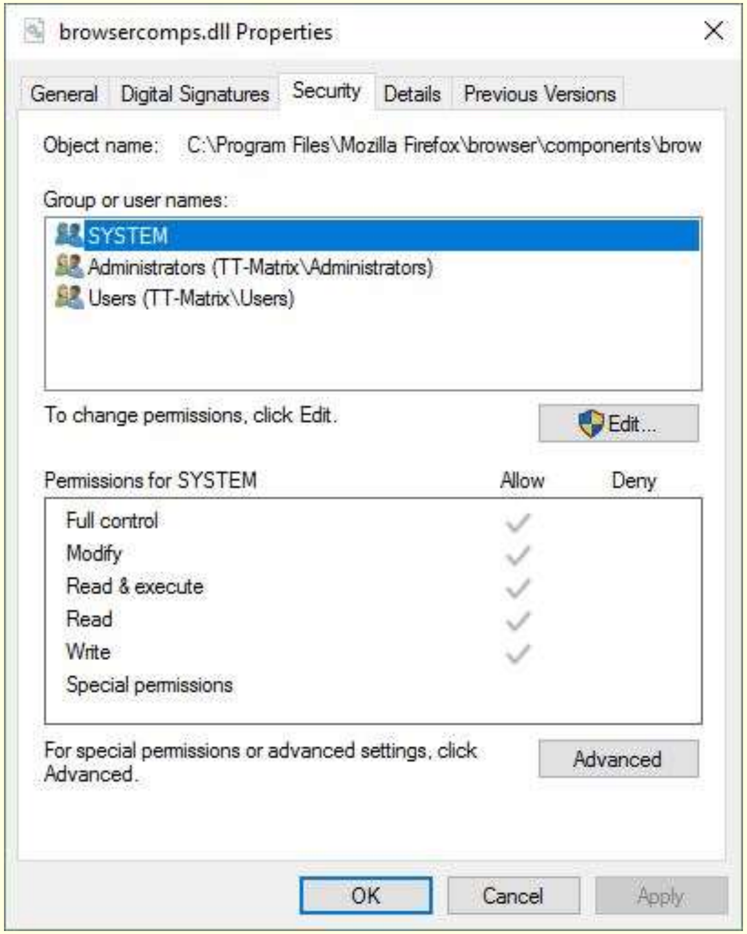
File Protection

- Access control based on
  - subject

    who wants to access the file
  - operation intended

    what type of access it is
- User groups
  - OSs define/allow to define user groups
  - A group can share files and other resources
  - Some OSs have predefined groups

    E.g. Windows: administrators, power users



browsercomps.dll Properties

Object name: C:\Program Files\Mozilla Firefox\browser\components\brow

Group or user names:

- SYSTEM
- Administrators (TT-Matrix\Administrators)
- Users (TT-Matrix\Users)

To change permissions, click Edit.

Permissions for SYSTEM	Allow	Deny
Full control	✓	
Modify	✓	
Read & execute	✓	
Read	✓	
Write	✓	
Special permissions		

For special permissions or advanced settings, click Advanced.

Permission Inheritance

- Permission: a particular type of access right (e.g. read)
- Permissions can be
  - *Assigned directly*
  - Inherited from a parent directory, process etc
  - Inherited permissions can be overridden by directly assigned permissions in most systems

Temporarily Acquired Permissions

- Console-based
  - Change user ID or group ID
    - Unix: `setuid` and `setgid` commands
  - Execute a command as another user
    - Unix: `sudo` command
- Program-based
  - Privileged execution
    - E.g. Java `doPrivileged()` method
      - Perform an operation when the invoked code has the permission to do it but the invoking code does not

File System Security Issues

- File protection may have no effect if the volume is accessed from a different system
  - Example: USB memory Sandisk: U3 protection works under Windows, but not under Linux
- Encrypted file systems
  - Require encryption key management, data granularity management

File System Reliability

- Destruction of files can be a greater disaster than destruction of other parts of the computer
- OSs have support for repairing slightly damaged file systems
    - Windows `chkdsk`, Unix `fsck` utilities, etc
- Storage system failures
    - Hard disks have bad blocks (due to manufacturing defects or operational problems)
        - Bad block list: Information about bad blocks, maintained by the file system

          Stored e.g. as a special file (cannot be deleted)
    - Interconnections (e.g. via network) can also cause file system problems
        - Network mounted file systems ("network drives")
    - Performance failure
        - The hardware cannot deliver the data in time

Redundant Array of Independent Disks RAID

- Method to divide and replicate data among multiple disks
- Improves performance, reliability or both
- Key concepts
  - Replication (mirroring): writing identical data to more than one disk
  - Striping: dividing data among several disks
  - Error correction: additional, redundant data is stored to help recovery of damaged data
- RAID levels
  - Define different services

    E.g. RAID 0: striping, but no replication or error correction,

    RAID 1: exact replication of a disk, …
- Problems
  - Disk failures are usually not independent
  - Equipment compatibility issues

Security in Ordinary OSs: Unix

- Unix basic components
  - OS kernel
  - Processes

    Each process
    - runs a program
    - has its own address space
    - is associated with a user who runs the process
- Security layers
  - Trusted base
    - Consists of kernel + some process run by the superuser (root)
    - Has full access to system resources
  - Other users
    - Have limited access to resources, according to the user's privileges

Unix Elements

- Subjects
  - Identified by a user ID (uid) and a group ID (gid)
- System resources (objects)
  - All objects, such as secondary storage, I/O devices, network are represented as files
    - Many of them are not files in the usual sense, e.g. physical devices, symbolic links
  - The semantics of operations (e.g. execute) may be different for different types of objects
  - Traditional permissions
    - Read, write, execute

Unix Protection

- Method: Combination of access control lists (ACLs) and capabilities
  - Objects have ACLs represented by protection bits
  - Capabilities established at authorisation time
- Discretionary access control
  - Each object has a protection state
    - Defines the operations that the system's subjects can have on the object
  - A set of operations are available to modify that state
  - Processes run by the file's owner can use those operations to modify the protection mode bits

Unix Authorisation

- Mediation is not complete
  - Controls each access to files by processes

  But

  - Access to certain objects does not require authorisation (e.g. network communication)
- Authorisation is granted or denied in the file **`open`** operation
  - If granted, the kernel creates a file descriptor that describes possible future operations – a form of capability
- The superuser (root) has automatic authorisation for any operation
- Time-of-check-to-time-of-use interval is a vulnerability

  E.g. userID can be changed by **`setUID`** in the meantime

Security in Ordinary OSs: Windows

- Basic principles are similar to Unix, but many details are more complex
- Subjects – similar to Unix
  - Users identified by a security ID (SID)
    - Concatenation of a statistically unique system ID and user ID
    - Certain SIDs (representing generic users) are constant across all Windows operating systems
- Objects – different from Unix
  - Can be of many types, including user-defined ones
    - Kernel objects: accessible by the OS kernel only (e.g. physical devices)
    - Executive objects: used by applications and services
  - Permissions
    - Many types that reflect object and operation variety
    - Include user-defined types

The Windows Security Model

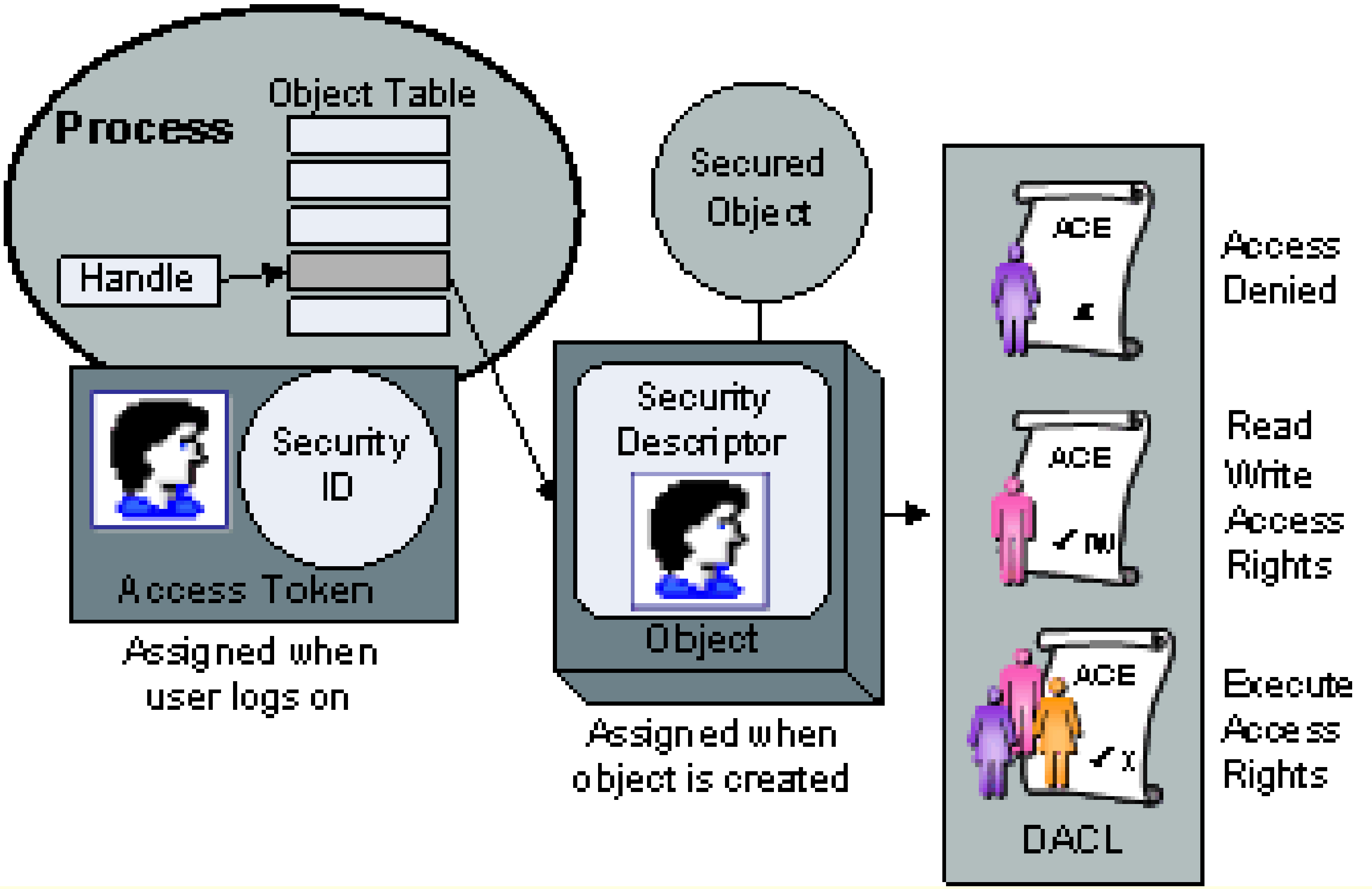


Image source: https://technet.microsoft.com/en-us/library/bb496995.aspx/

Windows Protection

- Trusted base
  - All system services and processes run by Administrator
- Discretionary ACL
  - Stores access control entries (ACEs)
  - Has positive (allow) and negative (deny) rights
  - Child objects can inherit ACE of the object
  - An object having no DACL can be accessed by anyone

DACL

Access Control Entry (ACE) 1	
Principal SID	Alice
ACE type	Deny
Access rights	Read, Execute
Inheritance flag	
Access Control Entry (ACE) 2	
Principal SID	Bob
ACE type	Grant
Access rights	Read, Write
Inheritance flag	

Windows Authorisation

- Access token
  - Identity and permissions of the user account running the process
- Authorisation process
  - The Security Reference Monitor (SRM) searches the ordered ACL
  - The search stops when the requested access is explicitly allowed or denied
- Mediation
  - Object manager
    - Centralised resource access broker
    - Its tasks include the verifying that a process has the right to use that object

UNIX & Windows Security Model Differences

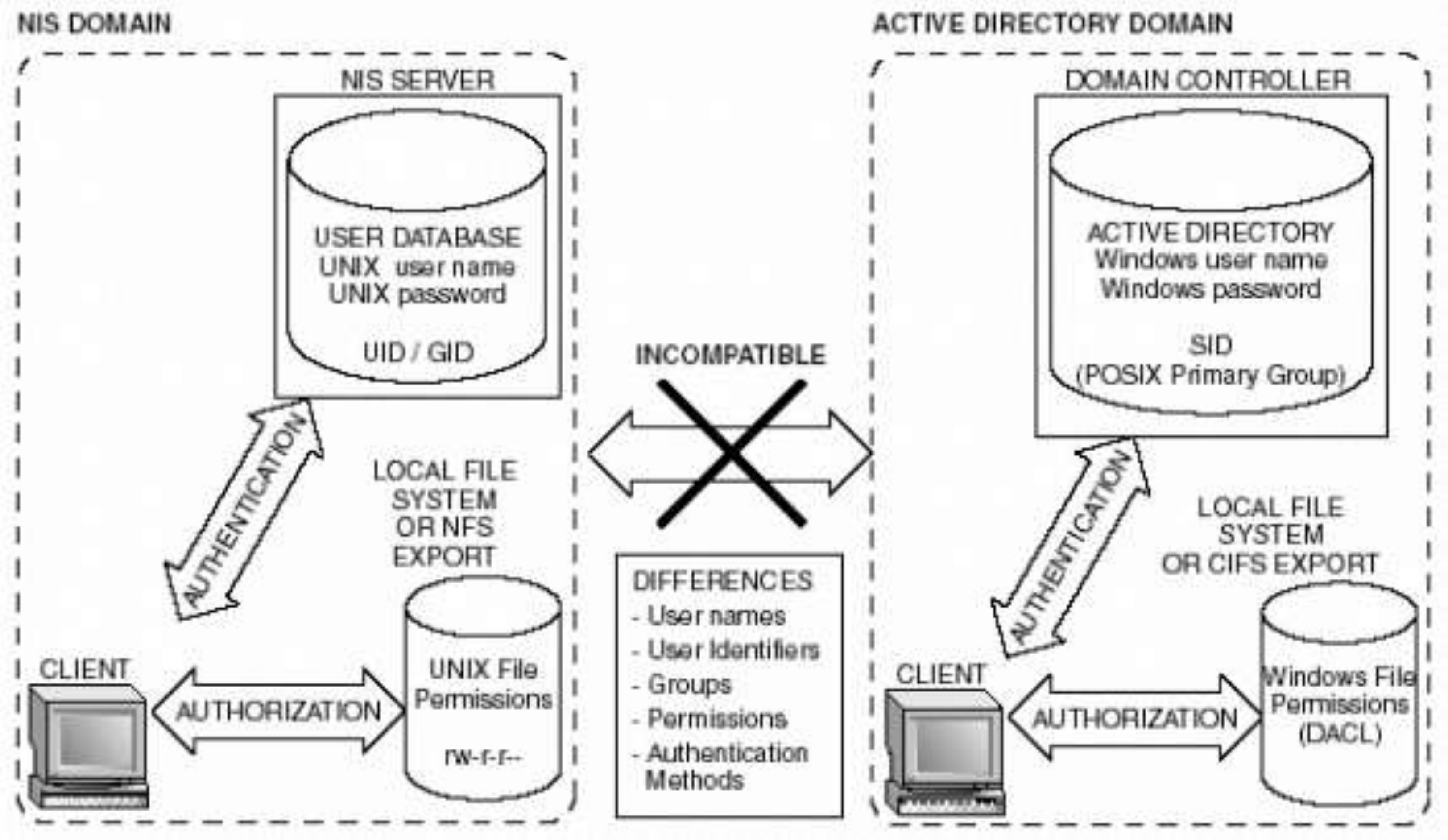


Image source: https://technet.microsoft.com/en-us/library/bb496981.aspx/

Summary

- OS: the basic interface between user and hardware
  - OS has to provide security for both
- OS focuses on memory protection
  - Main memory: data and working space
  - Secondary memory (disk): files
- Different operating systems use similar security principles, but the mechanisms can be very different