Designed by scratch in the 90s. At that time they could use knowledge and power when UNIX was designed.

1

Although NT's design principles remains the same, but it's much more complex features (including security). Ex. Windows introduced MAC, Linux introduced it later. Windows, after years, introduced a User environment very robust thanks to a separate VM with a full Linux kernel.

Learning objectives

- Review of the Windows Architecture
- Windows security model
- Discretionary Access Controls
- Mandatory Access Controls
- Vulnerabilities
- System hardening

2

References

The online chapter 26 "Windows Security" and the section 12.7 from the Computer Security – Principles and Practice (Pearson, fourth edition), W. Stallings, L. Brown

The chapter on Windows from the book "Operating System Concepts", 10th edition, Silberschatz, Galvin and Gagne

The Windows documentation: https://docs.microsoft.com/en-us/windows/

3 Windows is not an open S. OS. So we do not have a complete documentation. What we call windows is just the last version of ①, delivered in 1993. The objective of Microsoft was to replace ②, which was the first OS for computers. MS-DOS was produced for IBS for their producers. MS-DOS was thought at the time for single user machines,

Windows

First version called Windows NT ①

- NT stands for "new technnology", to mark difference from old MS-DOS ②
- delivered in 1993
- initially meant to support both OS/2 and POSIX API
- but then moved to Win32, due to the popularity of old Windows 3.0

Many versions over the years (XP, Vista, 7, 8, 10 and now 11…)

Still many old versions in use, often vulnerable and unpatched

and was starting to introduce the concept for personal comp. Mainframes worked with multiple users, that's why Unix was created with protective measures in mind. So MS-DOS no security interface. Apple was also developing its OS, similar because no concept of users, no protections exc.

4 Microsoft wanted to provide a system with a better UI, Win32 UI applied over MS-DOS. So system extremely vulnerable also for faults and problems. Microsoft hired a team of programmers working for another OS, to work without constraint about legacy.

Initially they wanted to support OS/2 and POSIX API, but users preferred the Win32 interface more so they worked with that.

Problem: people were attached to their applications that could run on the unprotected systems they had so transition was hard.

Still, Windows is used on a lot of servers, and may be used as frontend of systems and old systems might still have old versions.

[Win32 = APIs on system for the OS on MS-DOS], for this reason, Windows NT was incompatible with POSIX.

Windows

Windows is a 64-bit Operating System → Intel or compatible microprocessors

Designed for X86-compliant processors (Intel, AMD…)

Main features now include:

- POSIX compliance and multiple subsystems
- security
- Hyper-V virtualization
- multiprocessor support
- extensibility and portability
- international support (support many languages)
- app store (as for other OSs like Android or macOS)
- … and backward compatibility with legacy MS-DOS and MS-Windows applications

Available in several versions, from low-power devices, to laptops to servers

External config can modify the distribution to do something or the other. So we have a uniform one.

5

Security Principles

→ Very extensive, no limitation in memory: can occupy as much memory as they want.

Access control lists (ACLs)

- implement Discretionary Access Control

Integrity levels (by means of)

- mechanism to specify capabilities for classes of users
- implement Mandatory Access Control

} limits even the admin capabilities

Several mitigations for exploits that include:

- file system and communications encryption [Modules for encryption] In modern machine, this feature is at HW
- Address-Space Layout Randomization, Data Execution Prevention, Control-Flow Guard, Arbitrary Code Guard

Windows Defender Exploit Guard and Application Control

- ensure only trusted applications can run.

Windows Defender Credential Guard

- defends credentials, isolates the Local Security Authority (LSA) by means of virtualization

↳ process that manages login, runs in a separate VM

6

Compatibility:

- Posix source code can be compiled to run on Windows

Extensibility:

- layered architecture by means of:
  - Remote procedure calls (RPCs)
  - Advanced local procedure calls (ALPCs)

Portability:

- most of the code written in C and C++
- only few processor-specific parts in assembly, isolated into the Hardware Abstraction layer (HAL)
  - *The HAL is isolated in a Dynamic Link Library (DLL)*

Design Principles

7

Performance

- high-performance message passing among Windows subsystems;
- preemptive scheduling:
  - optimizes response time of processes
  - supports symmetric multiprocessor architectures

International support

- localized for many languages via the national language support (NLS) API

Energy efficiency

- especially for mobile and portable devices

Reliability

- uses hardware and software protection for virtual memory and for OS resources

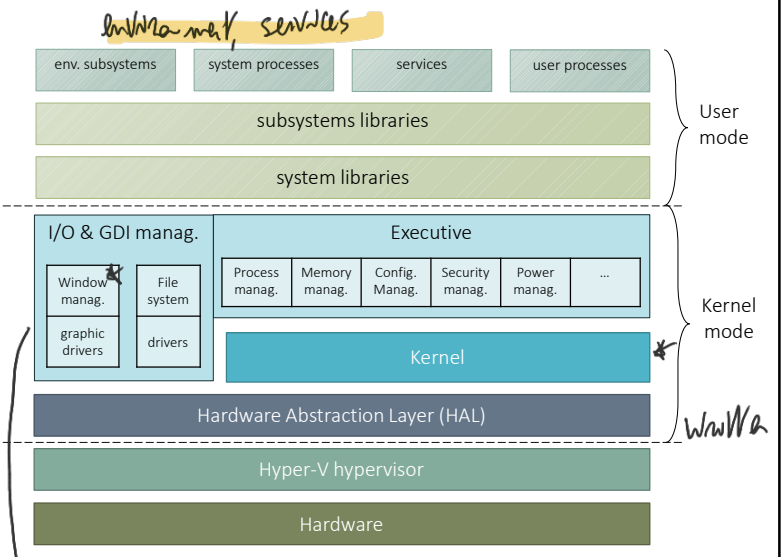
Design Principles

8

Above HW, there's Hypervisor on which Windows is installed.

*Kernel just means a component of the kernel, which is the implementation of the mechanism and executive implements the policy. All the managers are within the exception.
But Windows manager and File system, for performance reasons are connected directly to the HAL, but still use mechanism of Kernel.

Windows architecture



*Graphical UI that stands on the drivers.

This is written in C, only use the HAL for what cannot be expressed in C. They work directly on HW.

System components at each level operate under specific privilege layers

Beyond kernel and user mode, exploits virtualization (Hyper-V) to implement **virtual trust levels** (Win 10 feature):

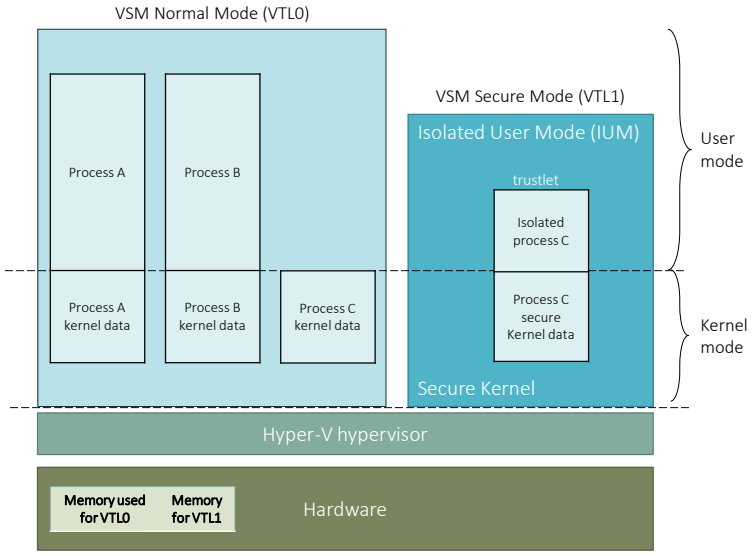
- here called normal world (VTL 0) and secure world (VTL 1), both with kernel and user mode
- this isolates VTL 0 from VTL 1
- the secure world also has a secure kernel and isolated user mode where trusted processes (trustlets) run
- the hypervisor runs in a special processor mode (VMX/VT-x Root Mode on Intel)

Windows Trust Levels

One concept: introduction of this layer of virtualization that lets the OS split in two separate VM, one more secure and one normal.

Two isolated Windows systems. The one in VSM, we have user processes and many OS processes.

Virtual trust levels (VTL)



But "trustlets" are isolated processes in the VSM. The hypervisor might even make a physical partition of the memory to avoid interference. For example process for the auth of the user, one runs in VM1 the other in VM0

11

In Windows the kernel is the low-level component of what we usually call "kernel"

Main roles:
- interrupt and exception handling
- thread scheduling
- low-level processor synchronization
- system recovery after a failure

component between the executive layer and HAL.

not huge, affordable

It is pinned in main memory (never paged out), and its execution cannot be preempted

To its purposes defines: → every component defines objects that abstract resources of the OS. The entire Sys is object oriented.
- *dispatcher objects* – include timer objects, event objects, semaphore objects, mutex objects, and thread objects; provides thread synchronization and dispatching functions
- *control objects* – include asynchronous procedure calls, interrupts, power notify & status, etc.

Two types of objects

The number of objects created are huge.

Kernel

12

Windows ①, a process is essentially a box containing many threads that can be scheduled.
↳ defines two type of threads: kernel threads, or user level threads are implemented using libraries = creation and scheduling and management doesn't require OS intervention.
Kernel level threads are known by the OS, so a lot more efficients.
So for ex: you cannot schedule a fiber over another core.

Process and Threads

Windows defines processes and threads

- the Windows API offers both kernel-level and user-level (fibers) threads.
- the kernel-level threads are scheduled by the kernel.

Process creation by means of `CreateProcess`:

- allocates the memory, loads code and libraries, initializes the first thread of the process;
- the process is managed by the kernel by means of its *handle*;
- the process can create threads by means of `Createthread`;

A process can allocate a kernel resource by means of the `Create` system call:

- the process obtains a handle to the resource
- the handle is local to the process (another process will get another handle for the same resource)
- the children of a process inherit all the handles to resources already acquired by the parent

↳ creates an object associated to processes.
↳ Sys call to create an object, CreateProcess = equivalent to a fork, but difference: fork doesn't take parameters. With this we have a lot of parameters.
↳ the result: it is a unique number to refer this object.

13

all these mechanisms are objects. Like inheritance with exec when creating childs, we have many mechanism, not like, objects with handlers and a child inherit handlers. You can also duplicate handler for an object and give it to somebody else.

Inter-process communications

The main interprocess communication mechanism is by sharing a kernel resource (e.g. pipe, mailslot, mutex...):

- a child shares the kernel resources with its parent
- at `Create` a process may give a name to the acquired resource, so that other processes can access the same resource by opening that name
- a process may use the `DuplicateHandle` function to pass the duplicate to another process (and thus share the resource)

Message passing is also possible:

- a thread can send a message to another thread or to a window
- every thread has its own input queue for incoming messages

14

Scheduling is ①: from POV of user, they can specify high level parameters translated in priority levels.

Threads scheduling

①

Pre-emptive, priority-based scheduling of threads:

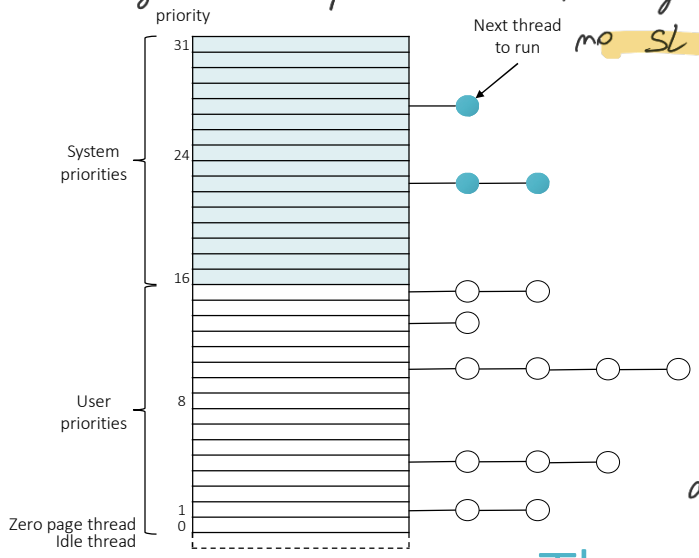
- multi-level feedback queues with 32-levels of priority
- priorities from 16 to 31 for system threads (kernel level)
- priorities from 0 to 15 for user threads
  - *basic priorities that can be attributed to user threads by the user:*
    1. `IDLE_PRIORITY_CLASS` (priority level 4)
    2. `BELOW_NORMAL_PRIORITY_CLASS` (NT priority level 6)
    3. `NORMAL_PRIORITY_CLASS` (level 8 — typical for most processes
    4. `ABOVE_NORMAL_PRIORITY_CLASS` (level 10)
    5. `HIGH_PRIORITY_CLASS` (level 13)
    6. `REALTIME_PRIORITY_CLASS` (level 24)
- priorities assigned dynamically to privilege interactive and I/O-buond threads against CPU-bound threads

There are jumps: this numerical level assigned may be changed over time, to give priorities depending on the needs of a thread.

15

Several queues for each priority level, round robin. We start with queues with highest priorities and we schedule thread. So sys threads have a high priority: no prob. of starvation for sys threads or user threads: sys threads might mostly be there for services asked by user level threads, so no UL scheduler, no SL scheduler. For ULT we also have

Threads scheduling



Note:

- Round Robin scheduling of the threads with the same priority
- A new thread starts with priority 8
- Priority raised up if :
  - thread reactivated after I/O operation (disk : +1, Serial line: +6, Keyboard: +8, Audio card: +8, …)
  - thread reactivated after waiting on a mutex/semaphore (+1 if in background, +2 if in foreground)
  - Thread didn't run for a given amount of time: priority goes to 15 for two time shares, against priority inversion
- Priority lowered if thread uses all time share (-1) [ex. CPU bound]
- When a window goes in foreground the time share of its threads is enlarged

a dynamic way to change priorities based on what the thread is doing, for ex. interacting with I/O.

16

KERNEL LEVEL THREADS BUT NOT NECESSARILY IMPLEMENTING SYS SERVICES (CAN BE USER LEVEL THREADS OR ...)

For MM: the part depends on HW; OS adds its own data structure. In modern architecture paging support is assumed.

because pages can be many, page tables are organized in multiple levels. 3-level pages for 32bit for ex.

Memory Management

Windows assumes the underlying HW platform supports virtual memory with paging

- page sizes defined by hardware (typically 4KB, can also have pages of 2MB or 1GB)
- multilevel page tables (number of levels depending on the addressing bit)
- working set page-replacement algorithm
- page states: valid, zeroed, free standby, modified and bad

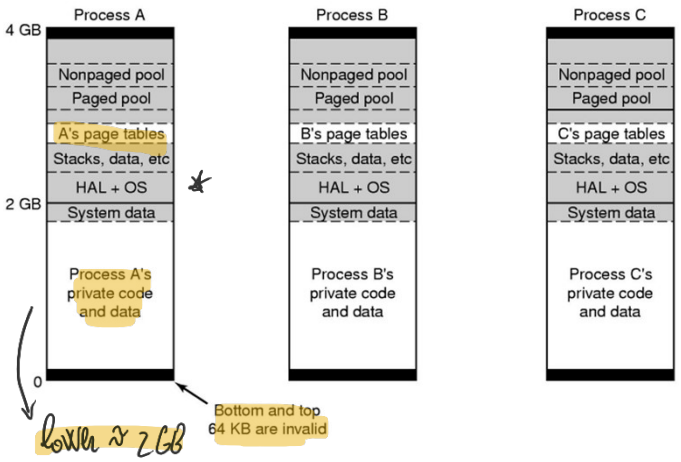
Virtual memory for a process up to 4GB in the 32-bit version and up to 128 TB in the 64-bit version

Physical memory up to 4 GB in the 32-bit version and up to 24 TB in the 64-bit version

↳ has also security implementation; VMemory has pages that are mapped to real pages. But we might not need it so allocate it for another process, that might need info! So for many pages there is a lifecycle that ends with zeroing page before beg assigned.

17 Ex of how VMem of a process in 32bit system is organized. Array of calls from 0 to 4GB.

Virtual Memory



Virtual space divided in two subspaces:

- low for user space
  - *in the 32-bit version it is of 2GB or 3GB*
- high for the kernel, and is shared among processes
  - *in the 32-bit version it is of 1GB or 2GB*

* white areas are private, shaded areas are shared (O.S. pages)

lower ½ 2GB used by process for its own purposes while process is running in UM.

* Maps sys calls, for ex: handler of calls.

18 Shared pages can be shared (if belong to the OS) kernel component for OS is a good candidate.

Process uses its own virtual memory to allocate data.

To allocate stack, heap, code exc, the Vmem is divided in two regions. Virtual pages are mapped as:

Virtual memory space unique, but divided into regions:

Each logical page can be:

free : if not assigned to any region
accessing it causes a fault for memory violation ↗ if they are allocated to regions

reserved : it's not yet in use but it's reserved to expand a region
For example, reserved to expand the stack of a thread
Accessing it brings the page in use ↪ there might be several regions to grow. To avoid problems we can reserve pages to use them in the future.

committed : if allocated to a region and in use

Memory Management

19

There are

System calls to manage virtual memory:

- `VirtualAlloc` reserves or commits virtual memory
- `VirtualFree` decommits or releases the memory

As in Unix, a process may map a file into its virtual memory

- two processes can share memory by mapping the same file into their respective virtual memories

The heap is a virtual memory region of reserved address space

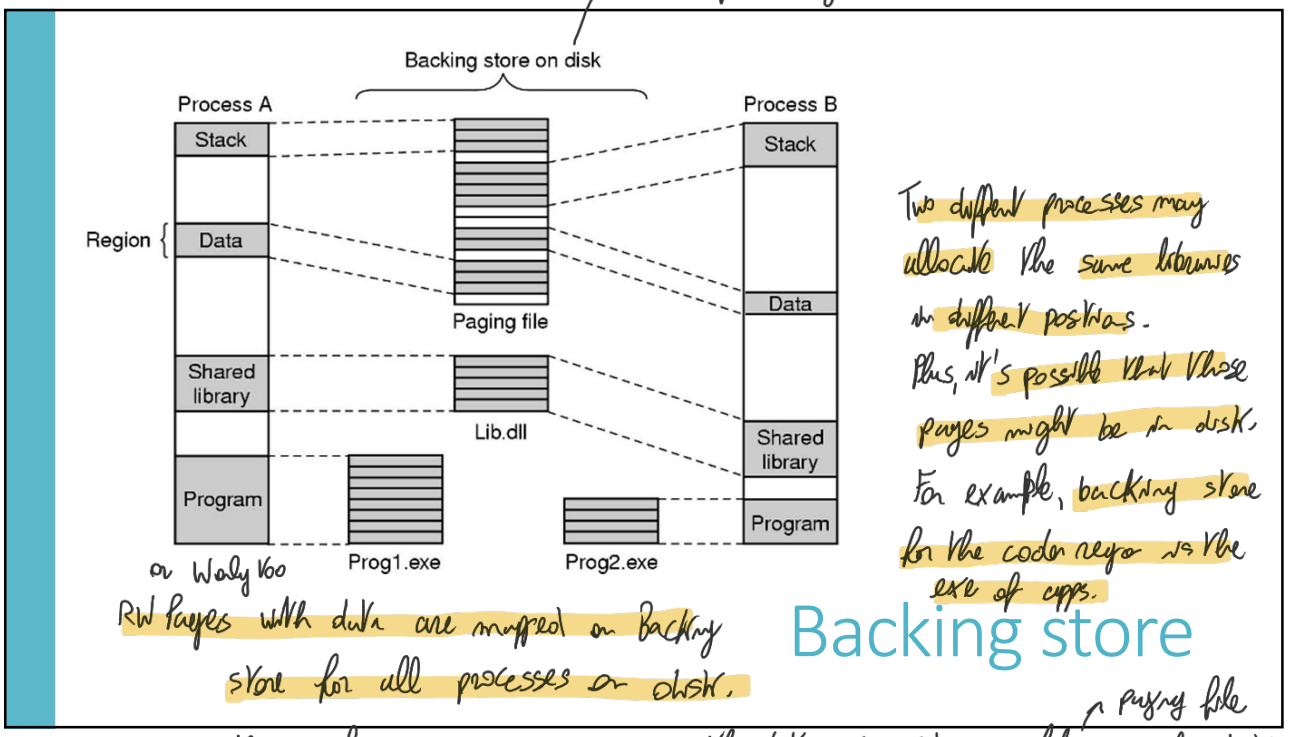
- by default a process starts with 1MB of heap
- heap shared by the process thread, hence its allocation is synchronized to prevent race conditions

Memory Management

20

example of regions and how processes organise their space.

Portion of memory on the disk



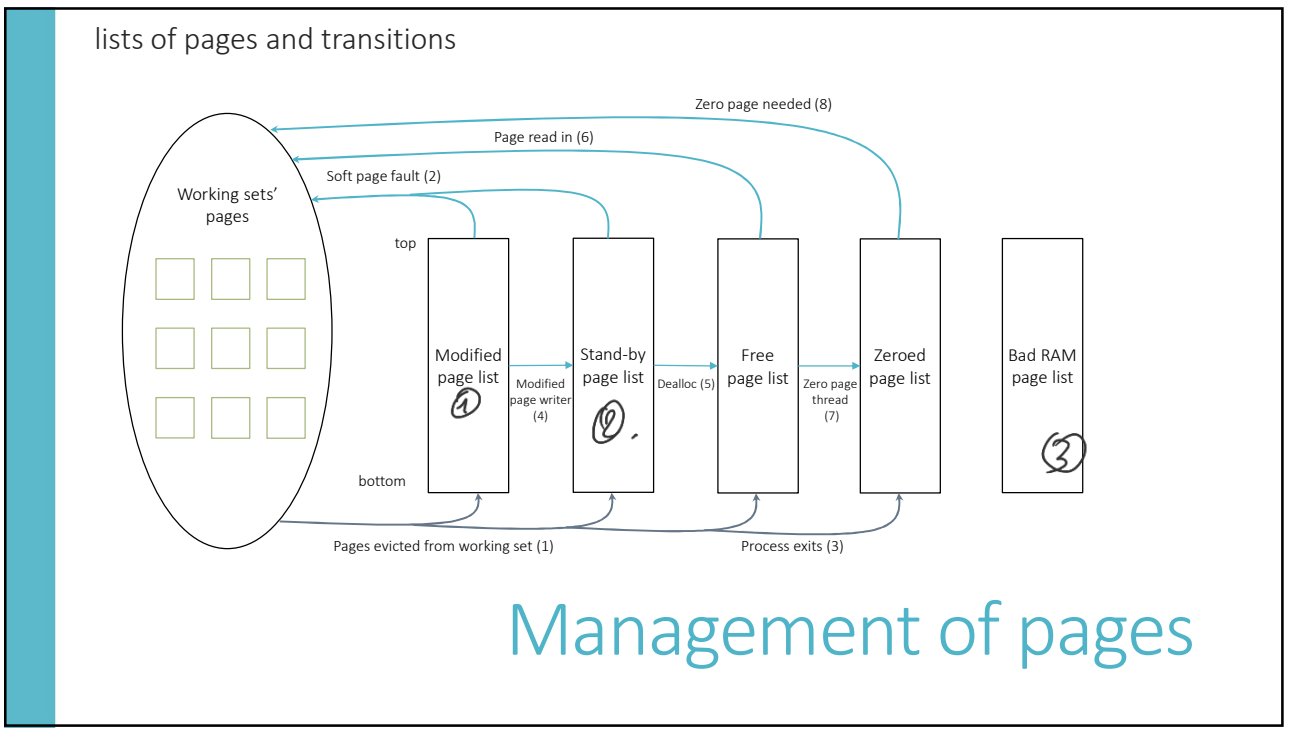
Two different processes may allocate the same libraries in different positions. Plus, it's possible that those pages might be in disk. For example, backing store for the code region is the exe of apps.

or Wally 160

RW pages with data are mapped on backing store for all processes on disk.

21

If we freeze a process, pages with data is saved in file (paging file) on the disk. If we inspect disk we can access to data on the disk.



22 life cycle of pages: every process keeps a working set, pages currently using that should be present in the memory. They might vary over time though! In this case, pages not used are evicted. If they've been modified they go in queue of ①, if not in the ②, because if modified you need to save them in the backing store. If process asks for them, we

can recover them with a SOFT PAGE FAULT and recover the more quickly. ① once written on disk goes in ②, becomes free page, gets zeroed* and can be used then. If the page is meant to be overwritten immediately, ① is not necessary. Pages can be completely pages it will never use ③

* in case of a malloc for new page

Manages the communication between applications and the device drivers (which also include the file system and network drivers)

Manages synchronization between device drivers and the rest of the OS:

- the devices operates at variable speeds and are asynchronous with the rest of the system
- the communication between the OS and drivers primarily done through I/O request packets (IRPs)
  - *IRPs mimic network communications*

The I/O system provides a layered driver model called stacks:

- Example: a **mouse driver** communicates to a **USB hub**, which communicates to a **USB host controller**, which communicates through a **PCI bus** to the rest of the computer hardware….
- … the stack consists of mouse driver, USB hub, USB host controller, and the PCI bus.
- all these layers in the stack communicate by means of IRPs.

I/O manager

23

Plug-and-Play (PnP) manager automatically recognizes when a new device is connected to the system:

- Loads the appropriate driver
- Keeps track of the devices connected and of the resources allocated to them

Power manager reduces the power consumption of the PC

- manages sleep and hibernation of the system and the different power levels of the processor
- when the PC goes to shut down or to low-power consumption it powers down also the attached devices while ensuring no data are lost
- Works with device drivers that support these functionalities

Plug and Play & Power managers

24

Important phase of OS: you start execute code in FW, that discovers where the OS is and load it in memory. If not secure, FW could execute a MW that could compromise the entire system. Happened in old MS-DOS. In modern systems, FW checks for the integrity of whatever is calling next. There are standards adopted by manufacturers ①

the boot loader architecture includes:

- a firmware-independent boot configuration and storage system called *Boot Configuration Data* (BCD)
- a boot option editing tool (BCDEdit.exe), which requires administrative privileges. In alternative use MSConfig.exe

booting through:

①
- Boot manager (bootmgr.exe) – generic, OS version independent
- Windows operating system loader (winload.exe) – specific of the OS & version
- Windows resume loader (winresume.exe) – resumes from hybernation

2 diff. loaders depending on the state of machine. If machine suspended ↓

boot sequence:

1. firmware boot loader (that calls)
2. UEFI (Unified Extensible Firmware Interface) application, provided by the SoC vendor (that calls)
3. Windows boot manager
4. once loaded, the kernel initializes all system processes

② standard for the specification of the loader that implement this.

- UEFI provides Secure Boot features that performs the integrity check by digital signatures of all firmware and boot-time components

FW has to identify on disk the place in which the OS is running. Standard is the way to locate loader on disk, checks for integrity and then runs.

System boot

* you need to resume machine from memory.

① 1st part of the chain is taken when you buy the HW, the second two added by windows.

[Backing store is a file on disk]

25

Review question

What are the main component of the Windows architecture? What is their role?

What are the trust levels?

What kind of scheduling adopts Windows?

What are the possible states of a page from the point of view of the memory manager?

26

Every component of OS defines abstraction of logical/physical resources as objects. Entire OS is object oriented.

Kernel objects

They are data struct. and operations.

An object is a data structure that represents a system resource
- Each component defines its object and exports routines to manipulate them
- No other component can directly access another component's objects, must use the exported routines

Each object has:
- a *header* – containing information about the object such as its name, type, and location and a security descriptor
  - *Object names are structured like file path names*
- a *body* – containing object attributes
  - *format of the attributes determined by the type of object.*

Three classes of objects:
- user – to support window management
- graphics device interface – to support graphics
- kernel – for all kernel related resources (memory, files, etc…)

Windows defines more than 25 types of kernel objects (see table for examples)

Examples of kernel object types
Files
Devices
Threads
Processes
Events
Mutexes
Semaphores
Registry keys
Jobs
Sections
Access tokens
Symbolic links
…

security descriptor → contains ACL → access control → so you can check if ops is allowed.

GUI

handle rendering and graphical output (ex. tools for drawing)

They are divided in classes that have diff. functions.

27

Creation of object returns an handle. We can create handles that can be shared, duplicated.

Kernel objects

objects are referred to and manipulated by means of handles:
- handles are process-specific:
  - a process must either create the object or open an existing one to obtain its handle
  - an object can thus have multiple handles
- the handle can be used to examine or modify the system resource → inspect, query or act on it. If obj is file, we can write on file.
- each handle refers to a (internally maintained) table that contain the address of the resource and the means to identify the resource type

handles are associated to access rights by means of Access Control Lists → because of handle with descriptor
- a process specifies access rights when it creates an object
- … and may change them later…

28

Kernel objects

All objects have the same structure (header and object-specific attributes)

Hence a single *object manager* can manage all objects to:

- create, destroy and manage life-cycle of objects
- keep object namespace database
- keep access rights to the objects and implement access control
- create object handles and to return them to the caller
- keep track of objects assigned to each process and maintain resource quotas
- creating duplicate handles
- closing handles to objects

29

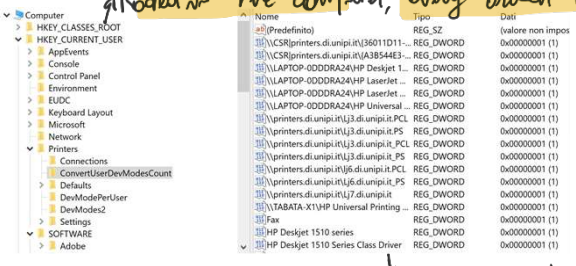
The registry

It is a hierarchical database that contains critical data of Windows and of the applications and services that run on it

it's so critical that windows creates a restore point before making any change to it…
…makes possible recovery if something goes wrong

Data structured in a tree

- each leaf in the tree is a registry entry and contains data
- each node in the tree is a key and it is a container for (sub)keys and data
- keys, subkeys and data are identified by their unique pathname in the tree
- some keys are entirely associated to a specific application



30

15

Main branches of the registry

- HKEY_CLASSES_ROOT – association of file types with programs and other configuration data
- HKEY_CURRENT_USER – the user profile of the user that is currently logged in
  - Includes environment variables, desktop settings, printers, applications preferences, …
- HKEY_LOCAL_MACHINE – information about the local computer system
  - Includes hardware, OS data, system memory, device drivers etc.
- HKEY_USERS – default user configuration and profiles of all users
- HKEY_CURRENT_CONFIG – contains no data, but just some pointers to the registry entry of the current HW and SW configuration… to speed up the access

The registry

31

The registry

The registry content is stored into a set of files called *hives*

- SAM – contains information in the key HKLM\SAM of the Security Account Manager
- SECURITY – contains security info associated to key HKLM\SECURITY
- SOFTWARE – contains SW config of the key HKLM\SOFTWARE
- SYSTEM – contains system config of the key HKLM\SYSTEM
- DEFAULT – contains default system info of the key HKEY_USERS\.DEFAULT.

All these files are in c:\windows\system32\config

System and user processes and even the kernel store and retrieve data from the registry by means of standard WIN32 API calls

- these data are used to apply their default configurations
- the registry can be edited with regedit (be careful…)
- however, it is normally (and safely) edited by acting on the Windows interface

The registry

32

In UNIX, File Sys: disk is seen as array of blocks. Pointer of blocks in which file is stored is saved in I-Nodes. Stored in beginning of mem.
We still have descriptor of files, with diff. structure, but no concept of an area in which we have them. Can be everywhere.

From the point of view of the user, NTFS is a hierarchical structure of directories and files, hosted in a volume

- the volume is a logical disk partition that may even occupy the entire disk — File sys takes one volume and puts it into them — Volume divided in clusters.
- the actual content of files and directories is stored into disk blocks that are called ***clusters*** in the Windows nomenclature
- a cluster is typically of 4 KB (but a FS can be configured differently) and is identified by a logical cluster number
  - *at low-level the partition is an array of clusters, indexed by their logical number*
- The entire FS is an object with its own metadata from pov of WIN.
  - *describe the FS configuration (e.g. cluster size, version, etc.)*
  - *all FS metadata in a regular file in the FS itself*

VOLUME: logical division of storage in one disk. [eg. of a DRIVE]

File system - NTFS

33

CORE of FS is MFT!

The Master File Table (MFT): equivalent of I-list in Unix. Organized as a file itself. Organized in records of 1KB each. Every record is a file descriptor. MFT also contains record for itself as a descriptor. So to retrieve the MFT, you need to retrieve the first record of MFT (why for it). Why? Because it is good, like is flexible. First block of MFT is kept in the first block of the disk. The first block contains two file descriptors for the MFT for reliability.

- It's a table with fixed-size entries (1KB each)
- each entry contains file metadata and data
- is stored into a file (it is itself a file)
- the first two entries of the MFT are the descriptors of the MFT itself
- The location of the first block of the MFT is in the super block at the beginning of the volume

A file:

- has a unique 64-bits ID called *file reference*
- described by a Master File Table entry (MFT)
  - *that also contains the security descriptor of the file (owner & access control list)*
- Allocated in a set of extents:
  - *each extent is a contiguous runs of blocks, similar to EXT-4 extents*

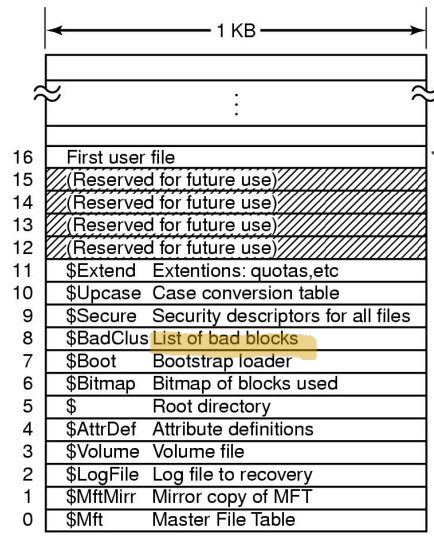
contains the pairs of <attribute, value> IN THE FD!!

File system - NTFS

Any file of FS is an object, formed as sequence of pairs of attribute and value.
File is a sequence of pairs.

34

MFT structure



Begins in record 16.

NOTE Both files defined by user or files of use or files of OS.

File Desc. of the MFT.

Every row contains records. The first 16 are reserved for OS

35

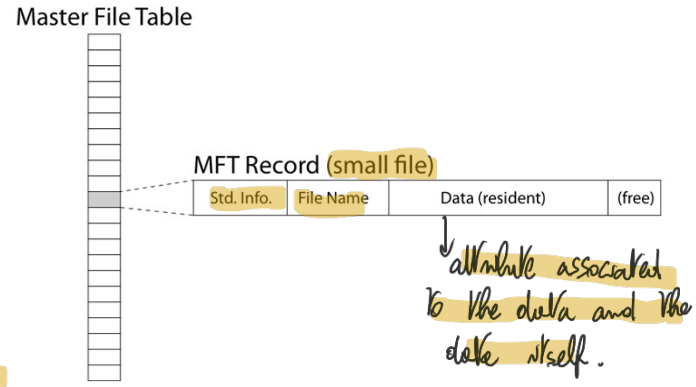
Retrievable from the 1st block of first.

Record 3: description of the entire volume (partition). By opening this file you can read at low level entire blocks.

Record 5 contains root dir. Root 6 bitmap of blocks in use, Record 8: not used.

Structure of MFT records. List of pairs, key-record.

- the MFT contains pairs «attribute,value»
- each key is associated to a metadata
- one of the keys is the file name
- even the content of the file is within a pair «key,value»
- hence, if the file is very small, its entire content can be stored into the MFT record directly



attribute associated to the data and the data itself.

Content of the file is one of the file attributes.

MFT record of a small-size file

36

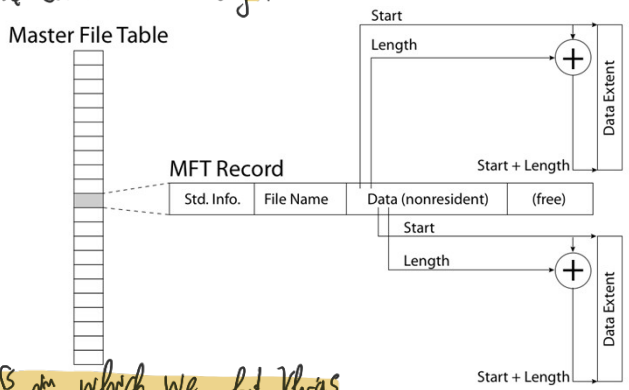
If file grows and goes beyond, you have to allocate blocks.

NOTE: Content of the file is an attribute of the file. In general

you do not store individual ptrs but initial blocks and length.

Key

- Pairs «attribute,value» that are too big to stay within the MFT record are non-resident
  - that is, they stored externally in data extents
- An extent is a sequence of contiguous data blocks (clusters):
  - identified by runs: «initial block, length»
  - also EXT4 in Linux adopts a similar model

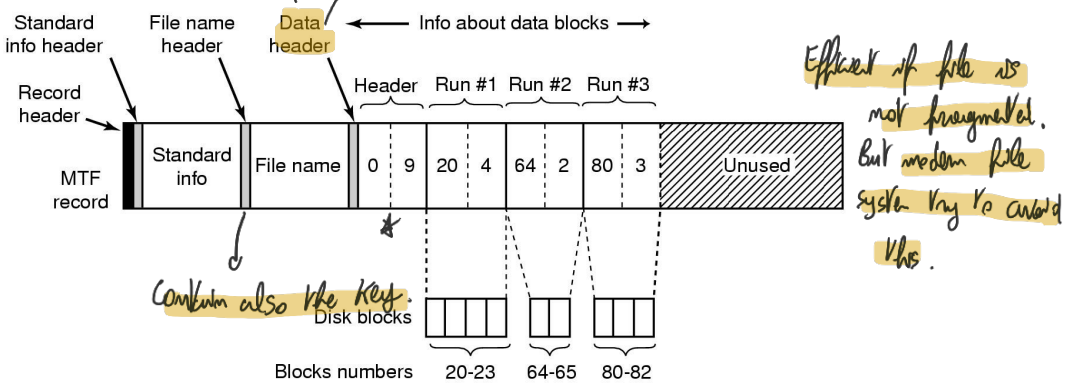


Key = data, values are blocks in which we find things.

MFT record of a medium-size file

37

Key that describe the key of data attributes.



Efficient if file is not fragmented. But modern file system try to avoid this.

Contains also the key.

MFT record of a medium-size file

38

It is possible to start writing a file in any arbitrary bytes. So you have empty regions and representing them is useless. The header is telling us where available data starts and stops.

What you do is to OPEN a file, then you need for ex.
When you open a file you initialise a ptr from the first byte of the file and work from there. In C is possible to do a seek operation on a file. This means you move your file ptr and then you write.

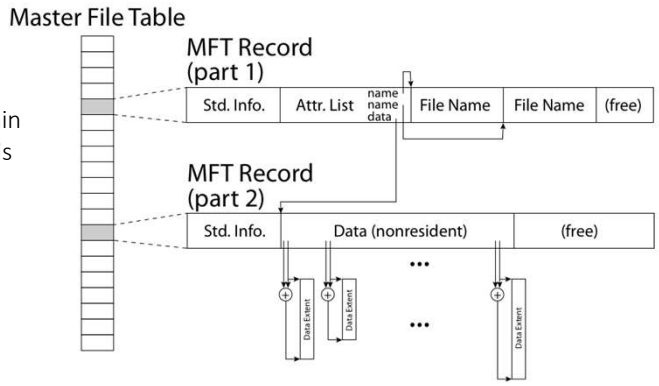
seek(f, 4096)
write(h, "hello"). The first 4kb have never been initialised. You expect to find a 0. The OS won't allocate a block for uninitialised blocks at the beginning.

Windows will say: the first logical block of the file begins at logical block 1 and ends at logical block 1.

Flag that tells
"This is the key. The value will be there: ----"

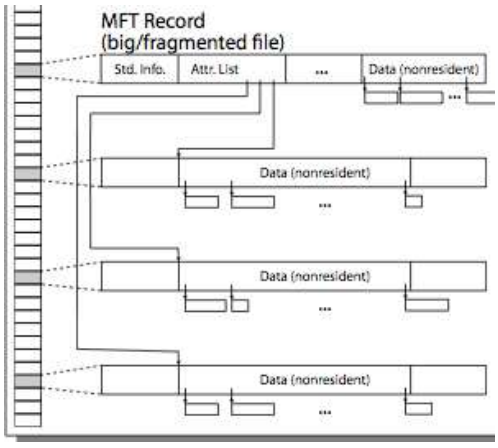
What if you don't have enough space for all the runs? Several possibilities: many runs, not too many. You use a 2nd MFT record that is an extension. In the attribute list you will have a ptr of another record.
What if not? You might use many extensions of MFT placing the parties of MFT records.
What if there are so many MFT records that do not fit the MFT record? They may be stored as external attribute. Becomes complex.

If the list of runs is too long to be kept within the MFT record, an additional MFT record is used.
It is linked as in the picture



NTFS indirect block

39

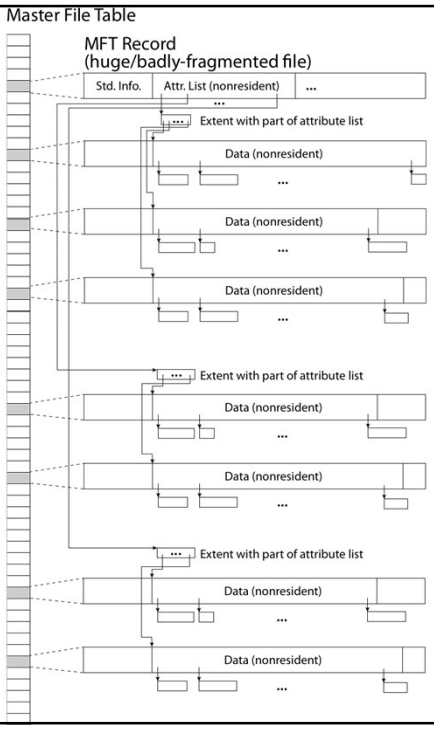


NTFS multiple indirect blocks

… if the list of runs is still too long then multiple indirect blocks can be used

40

File is in general extremely fragmented if disk is almost full



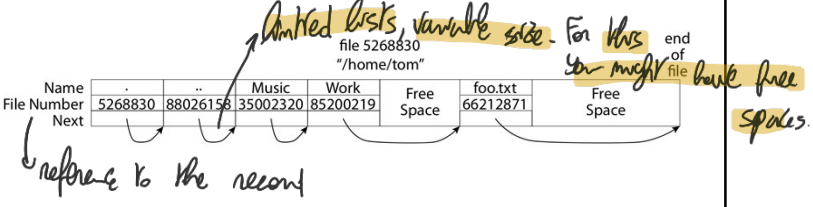
NTFS multiple indirect blocks

... and if it's still too long, then the list of run itself can be stored as non-resident data, in some extent

41

List of names associated to ptr to INODES (Unix)
↳ to MFT records (for Windows)

- directories are files as well
  - map file name to file number (#MFT record)
- small directories organized as a linked list



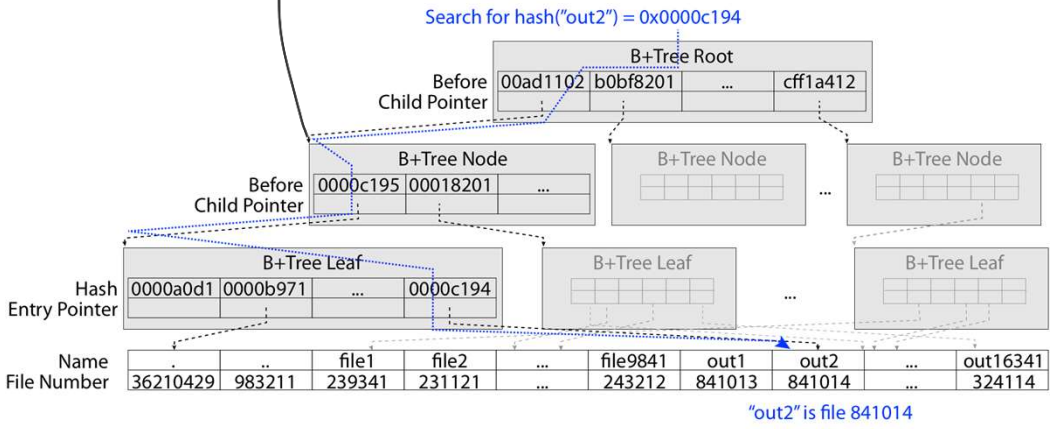
Linked lists, variable size. For this you might have free spaces.
end of file
reference to the record

If dir is large and has lots of files there's another representation.

Directories

42

Retrieval in logarithmic time for more efficiency.



Large Directories: B-Trees

43

organize in RAID0/1 with/o additional driver

native support to RAID disks (level 0 and 1)

support for FS compression

journaling to improve the FS reliability:

In cases of old sys, might happen that in case of crashes, op. not executed still on disk, there was a high risk for inconsistence of File System. For this reason, performing FS checks periodically was a must.

- all NTFS data structure updates are performed inside logged transactions (log file described by MFT record #2)
- active only on the NTFS metadata and structure, not on the actual data
- journaling method similar to that of EXT4 (we will see it…)
- aims at keeping the FS data structures consistent after a crash

When you have to apply a write op on the disk, in case of crash you can check the journal and resume a consistent form.

NTFS – other features

44

Each update to the file system first written in the journal in the form of a transaction

- Each transaction on the journal has a sequence number

Updates to the file system follow this procedure:

- First write a copy of the blocks to be written in the journal
- When data is committed in the journal then update the file system

Journaling — NTFS, EXT3 & EXT4

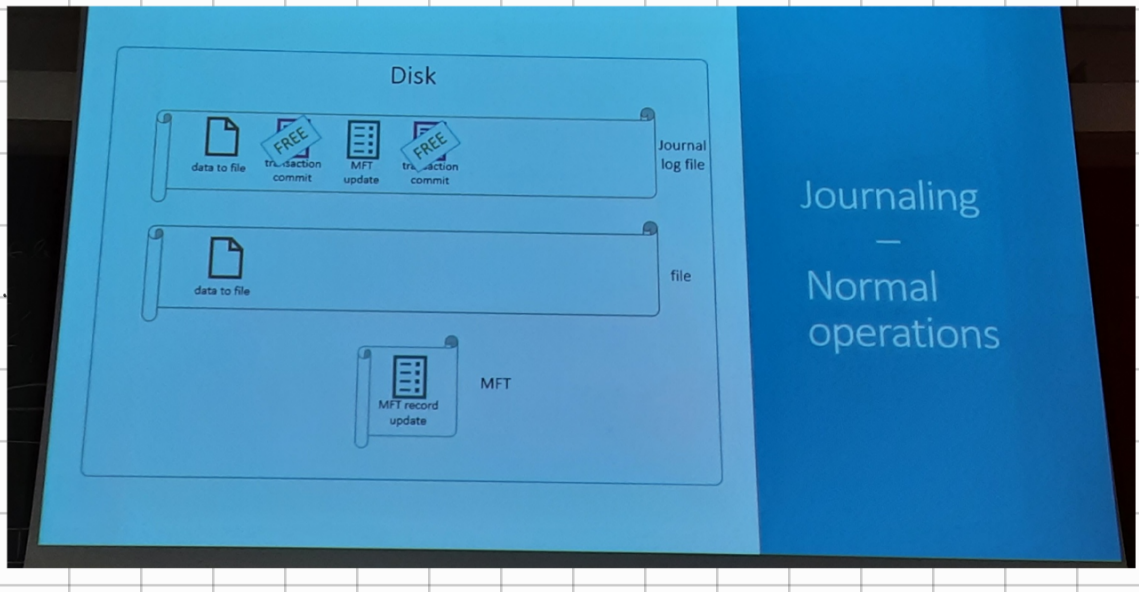
An update OP might call for update of several ops of updates for file system. For this, you might have inconsistencies.

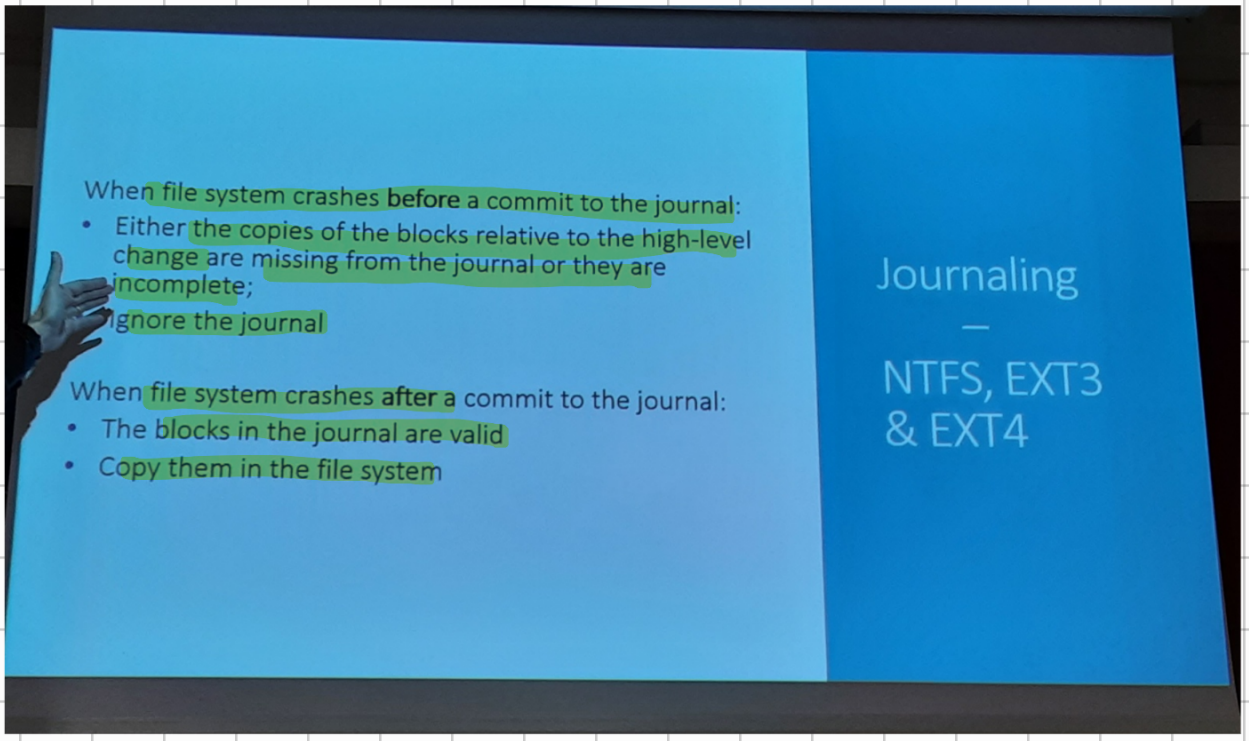
Let's say you have to append data to file. 1st, ya append data to the journal.

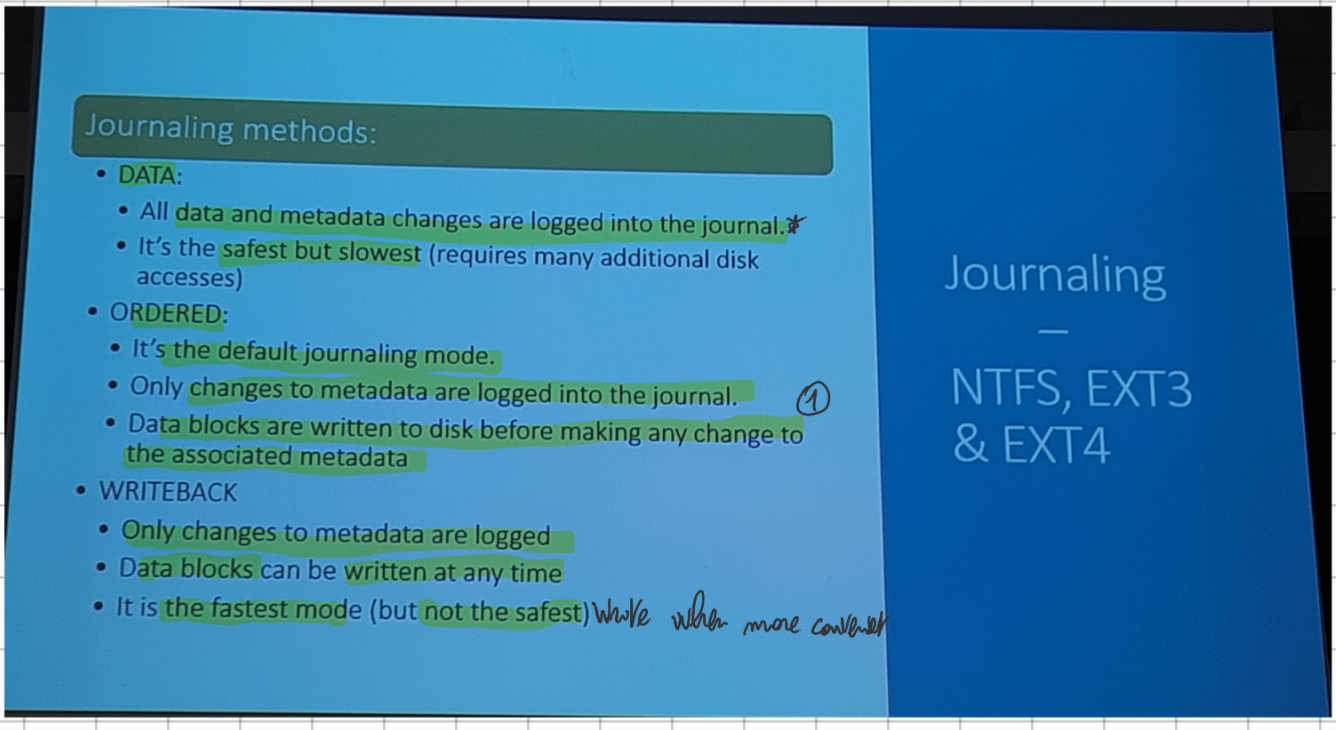
Then you write commit transaction knowing that this data has been committed to the journal.

After writing, you have to update MFT record (critical). Record your update of MFT, commit this on journal, then apply the update and then you free the commit. In case you find no unfreed committed transactions then ok.

Data + no commit = ignore.



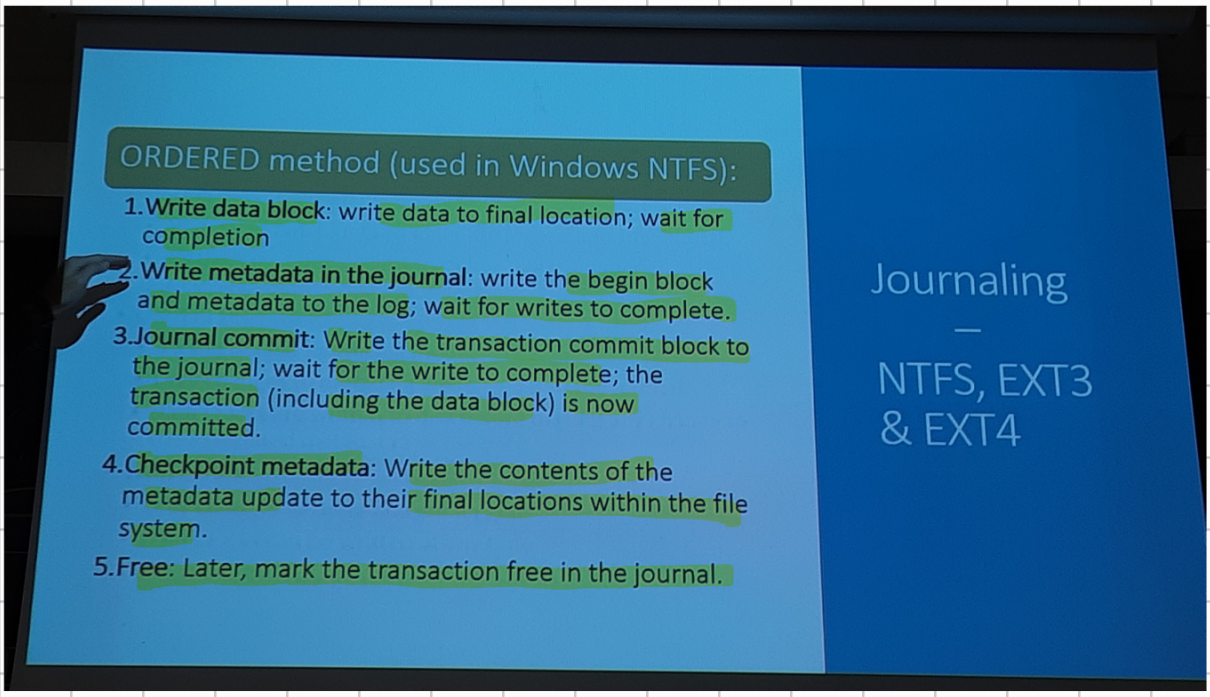




*Commit both changes in data and data structure (MFT record). So every data should be written twice! For this, not used.

① For metadata the sequence is the one we've seen. With ORDERED, data blocks where before making changes on the metadata.

Ordered is a good compromise. Risk of having written data on file that will not be recorded on the file. Only damage to file, not file system. Good compromise.


ORDERED method (used in Windows NTFS):
1. Write data block: write data to final location; wait for completion
2. Write metadata in the journal: write the begin block and metadata to the log; wait for writes to complete.
3. Journal commit: Write the transaction commit block to the journal; wait for the write to complete; the transaction (including the data block) is now committed.
4. Checkpoint metadata: Write the contents of the metadata update to their final locations within the file system.
5. Free: Later, mark the transaction free in the journal.
Journaling
—
NTFS, EXT3 & EXT4

Network protocols implemented as drivers, can be loaded (and unloaded) dynamically

supports a lot of network protocols and

Beyond TCP (v.4 and v.6), many other protocols are supported

- Server message block (SMB): a network file sharing protocol
- Network Basic Input/Output System (NetBIOS): a hardware abstraction interface for networks
  - *Establish logical names on the network*
  - *Establish logical connections of sessions between two logical names on the network*
  - *Support reliable data transfer for a session via NetBIOS requests or SMBs*
- Point-to-Point Tunneling Protocol (PPTP): create a secure communication among windows hosts (in practice implements a VPN)

Networking

45

specialised with Windows. Object oriented and output is object oriented too. Supports similar features than unix shells.

Power-Shell provides access to Windows computers, including security settings

- can be used to create tailored management tools
- it is based on .Net (hence whatever can be done in C# or VB.NET is also possible in powershell)
- supports many features of Unix shells, like piping a command to another, for example:
  - `Get-process chrome | Format-Custom` (shows only Chrome processes with customized formatting)
  - `Get-Service WebClient | Format-Custom` (shows the service WebClient with customized formatting)
- commands are called cmdlets, their output is object-based
  - this is different than Unix shells

Mostly for sys admins.

Manuals at https://docs.microsoft.com/it-it/powershell/scripting/getting-started/getting-started-with-windows-powershell?view=powershell-7

Powershell

47

Review question

What is the Windows registry?

↳ kernel object as something defined by kernel. It describes

Why does in your opinion a key in the registry is a kernel kernel object? configuration of critical parameters. For this reason they must be protected. Because objects come with security descriptors we can implement access control.

What kind of data structure is the MFT? What is its purpose?

↳ Table, array of records

What is the smallest space a file may occupy in the file system?

48

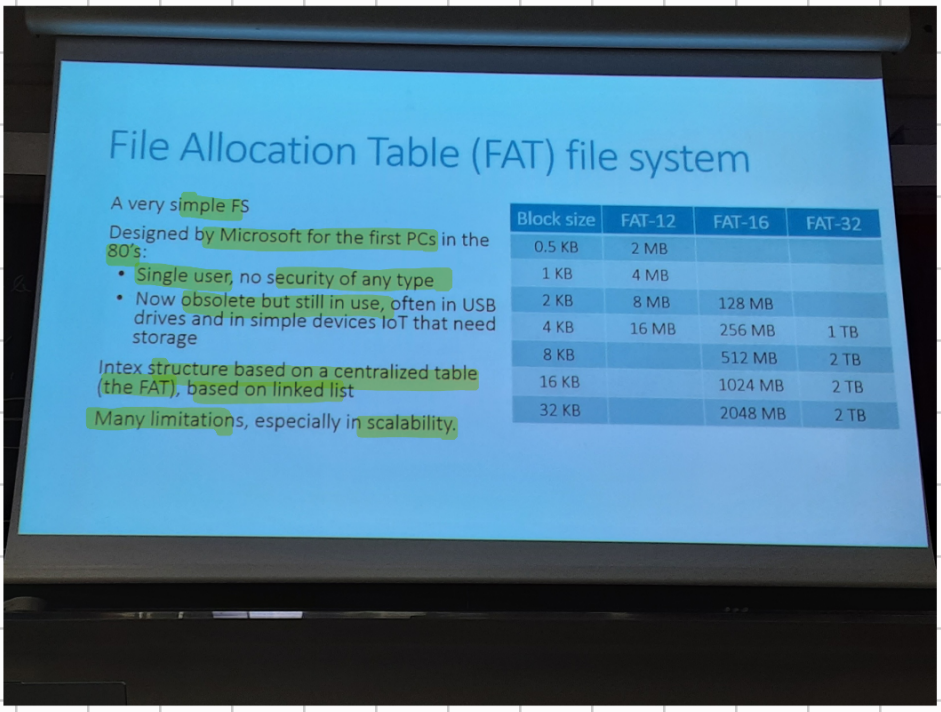
Windows security

Computer Security – Principles and Practice, W. Stallings, L. Brown

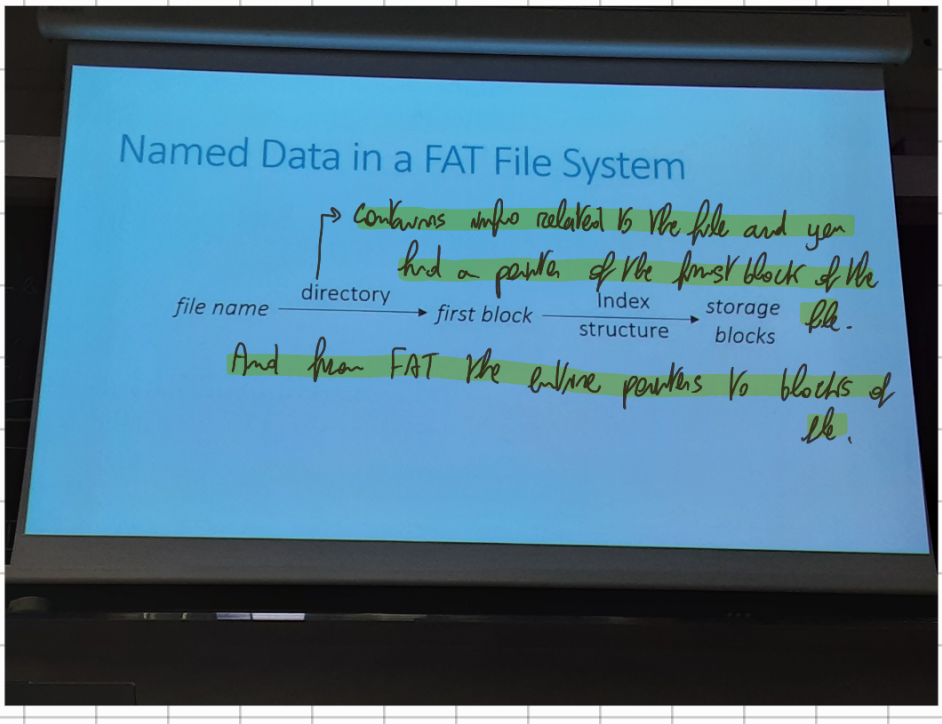
… but also Windows documentation: https://docs.microsoft.com/en-us/windows/

49

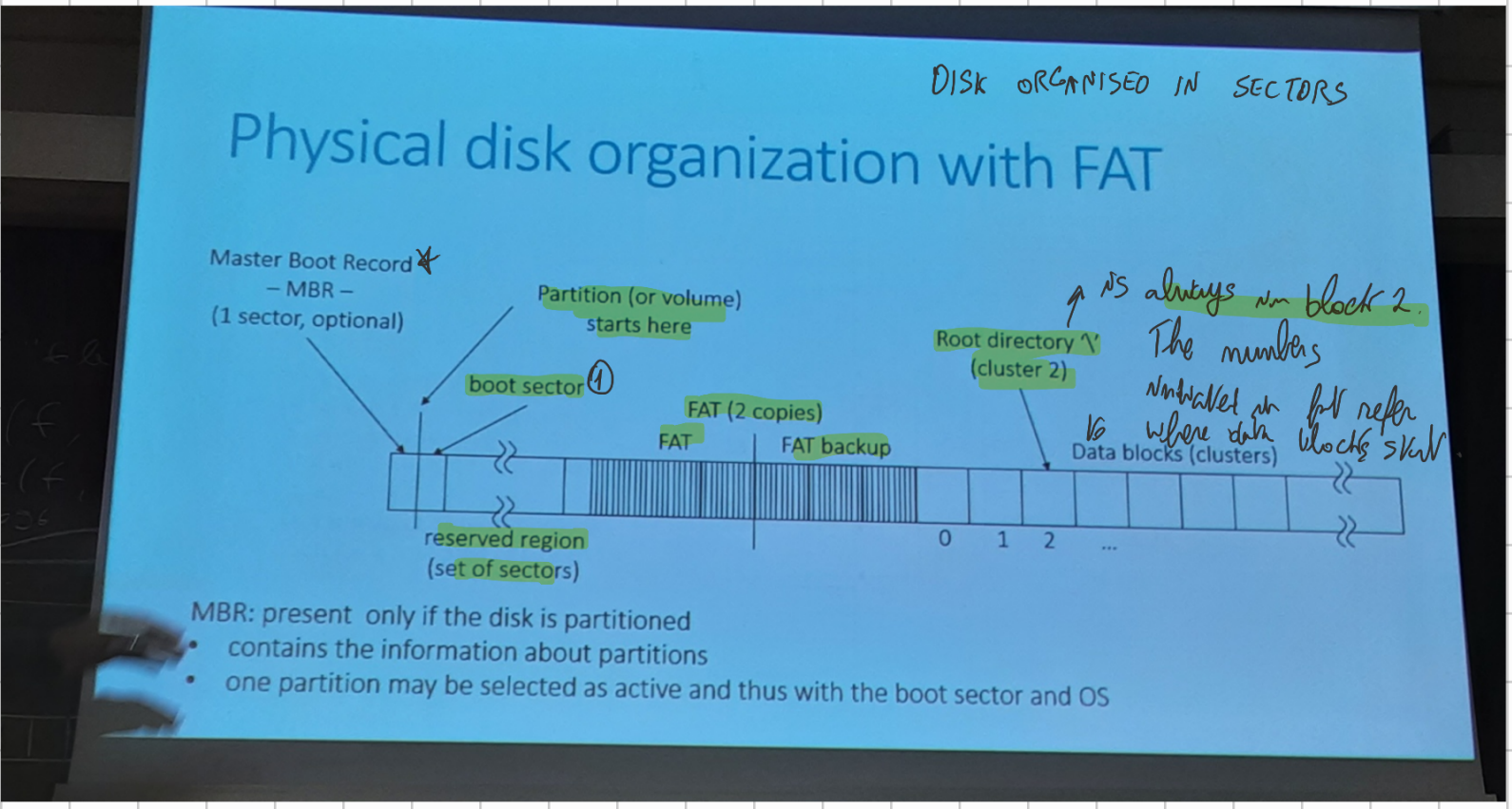
Nowadays on different systems you find good compatibility drivers for NTFS, so not much need for FAT anymore, which was used for compatibility.



2 reasons: 1. No security information kept by FS. 2. Leverages on a unique data structure to define all the files and you cannot separate files. One big structure that describes everything, so to operate with FS you need to keep in memory all the FAT.



FAT contains pointers to data blocks. Imagine you have the first block of the file, in that position of the FAT, you will find the next block of the file or a null or termination.



* 1st block: descr. of partition of the disk. In every partition you have one FS.

① Contains code loaded when you start the OS. Plus you also have table of pointers for FS.

Physical disk organization with FAT 32

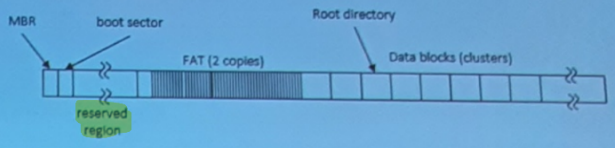
The reserved region, contains two copies of reserved sectors, in particular:

- Boot sector
- FSInfo sector:
  - *Number of free clusters (blocks)*
  - *First free cluster*
  - ...

...the actual format depends on the version of the FAT

Boot sector in FAT 32: ↱ Physical sector

- **Number of bytes per sector** (2 bytes, offset 0X0B)
- **Number of sectors per cluster** (1 byte, offset 0X0D)
- **Num. of reserved sectors** (2 bytes, offset 0X0E) (equals the first sector of the FAT)
- **Number of FAT** (1 byte, offset 0X10), usually =2
- **Number of sectors per FAT** (4 bytes, offset 0x24)
- **Number of the first cluster of root directory** (4 bytes, offset 0x2C) – usually =2
- Total number of data clusters
- ...
- and boot code, of course



Boot sector in sector 2 actually.

Structure of the Directory

A sequence of fixed-size entries

Each entry:

- Short filename: 11 bytes (8 bytes filename, 3 bytes extension) – offset 0x00
- Attribute byte: 1 byte – offset 0x0B (read only / hidden / system / directory / ...)
- Time&date of: creation, access, last write
- File cluster high: 2 bytes – offset 0x14
- File cluster low: 2 bytes – offset 0x1A
- File size: 4 bytes – offset 0x1C
- Extended filename
- ...

Windows Security

Windows is the world's most popular OSs:
- strength is that security enhancements can protect millions of nontechnical users
- weakness is that vulnerabilities in Windows can also affect millions of users

next points:
- overall security architecture of Windows
- vulnerabilities
- security defenses

50

Windows Security Architecture

Main elements:

Key elements of Windows security:

Security Reference Monitor (SRM) Most important el. Component that implements Access control.
- Manages access control

Local Security Authority (LSA)
- Manages security policies in particular responsible for authentic. and authorization of users. *1

Security Account Manager (SAM)
- Database with users and groups information
  - ↳ only for local users

Active Directory (AD)
- User authentication in a domain*
  - ↳ Authenticate users over the network

Plus:
- Authentication packages
- WinLogon and netLogon – handle logons at the keyboard and across the network, respectively

51 *1 The LSA returns a security token. DS that contains security info about you. When you are recognized

*Manages remote request. So you authenticate with external services.
You connect to a service and receive a token and you can use this token to access things.

Security Reference Monitor

Security Reference Monitor (SRM) – a kernel-mode component that:

- performs access control – when a process opens a handle to an object:
  - checks the process's security token
  - checks the object's access control list
  - verify whether the process has the necessary rights
- generates audit log entries,
- manipulates user rights (privileges)

Small component that can be easily verified and made vulnerability-proof

A similar component included in most modern OS

52

Local Security Authority

Local Security Authority (LSA) – responsible for enforcing local security policy that manages:

- password policy, such as complexity rules and expiration times
- auditing policy, specifying which operations on what objects to audit
- privilege settings, specifying which accounts on a computer can perform privileged operations

It also issues security tokens to accounts as they log on to the system.

It runs in a user-mode process named lsass.exe (although in Isolated User Mode – VLT 1)

53

Security Account Manager

Security Account Manager (SAM) – a database that stores user accounts and local users and groups security information:

- *Local*: only user and groups information for a specific machine, different than *domain* accounts (which are managed centrally for an entire organization by the Active Directory)
- local logins perform lookup against SAM database
- In old Windows passwords were stored using MD4, now uses password-based key derivation function (PBKCS)

resides in the `\Windows\System32\Config` directory (equivalent to the `/etc/passwd` of Unix)

NOTE: SAM does not perform logon, that's matter of the Local Security Authority (LSA)

54

Active Directory

Active Directory (AD)

- It's the Microsoft's LDAP directory
  - *LDAP (Lightweight Directory Access Protocol) is a standard protocol for managing directory services …*
  - *… that are centralized managers of information and resources in a computer network (with its respective access control)*
- all Windows clients can use AD to perform security operations including account logon
- authenticate using AD when the user logs on using a domain rather than local account
- user's credential information is sent securely across the network to be verified by AD
  - *credentials and not just passwords…*
  - *… they can take other forms…*
  - *…refer to user authentication classes*

WinLogon (local) and NetLogon (net) handle login requests

55

Local vs Domain Accounts

Only local auth. against SAM.

A networked Windows computer can be either:

In a domain

- users can login with domain (by means of AD)
- local accounts are also possible but do not grant accesses to domain resources (printers, mail servers, etc…)
- centrally managed and much more secure:
  - *account management, security policies all centralized in AD*
  - *more secure and saves time to administrators*

in a workgroup ↳ early simple means of networking

- a collection of computers connected together
- only local accounts in SAM can be used
  - *hence only local authentication of users*
- no infrastructure to support AD domain

Use local auth that has validity on the workgroup.

Not recommended.

56

No user id, token given to the user described in terms of security identifiers.

Accounts in the Active Directory

domain administrator adds user's account info to the system (name, account, password, groups, privileges)

- groups and privileges are optional

account is represented by a Security ID (SID)

- unique to each account within a domain
  - *note: if you delete an account and recreate a new one with the same name the new one will be actually different.*
- of form: S-1–5–21-AAA-BBB-CCC-RRR (version of SID)
  - *S stands for SID, 1 is the SID version*
  - *AAA-BBB-CCC is the unique number representing the domain*
  - *RRR is a unique number within the domain (this is what makes each account unique)*

Numbers tell you context of SID.

S, 21 identify the security authority and sub-authority for the account.

Windows NT domain, 21 = sub-authority, specific

57

CONTAINED IN THE TOKEN

```
PS C:\Users\stefa> whoami /USER
INFORMAZIONI UTENTE
----------------
Nome utente     SID
=============== ==============================================
tabatayoga\stefa S-1-5-21-1144226474-1424528306-2619936039-1001


PS C:\Users\stefa> whoami /GROUPS
GROUP INFORMATION
-----------------
Group Name                                 Type             SID                                            Attributes
========================================================================================================================
Mandatory label\Medium Integrity level     Label            S-1-16-8192
Everyone                                   Known group      S-1-1-0                                        Mandatory, enabled, predefined,…
…
BUILTIN\Administrators                     Alias            S-1-5-32-544                                   Only for negotiation
BUILTIN\Users                              Alias            S-1-5-32-545                                   Mandatory, enabled, predefined,…
…
NT AUTHORITY\Auth. Users                   Known group      S-1-5-11                                       Mandatory, enabled, predefined,…
MicrosoftAccount\ste@outlook.it            User             S-1-11-96-3623454-…-2…5863                     Mandatory, enabled, predefined,…
NT AUTHORITY\Local account                 Known group      S-1-5-113                                      Mandatory, enabled, predefined,…
LOCAL                                      Known group      S-1-2-0                                        Mandatory, enabled, predefined,…
```

Powershell Example

58

Login with Active Directory

username in one of two forms:

- SAM format: DOMAIN\Username (legacy format)
- User Principal Name (UPN): username@domain.company.com

if at login the user enters only the username, it is attached the domain of the machine

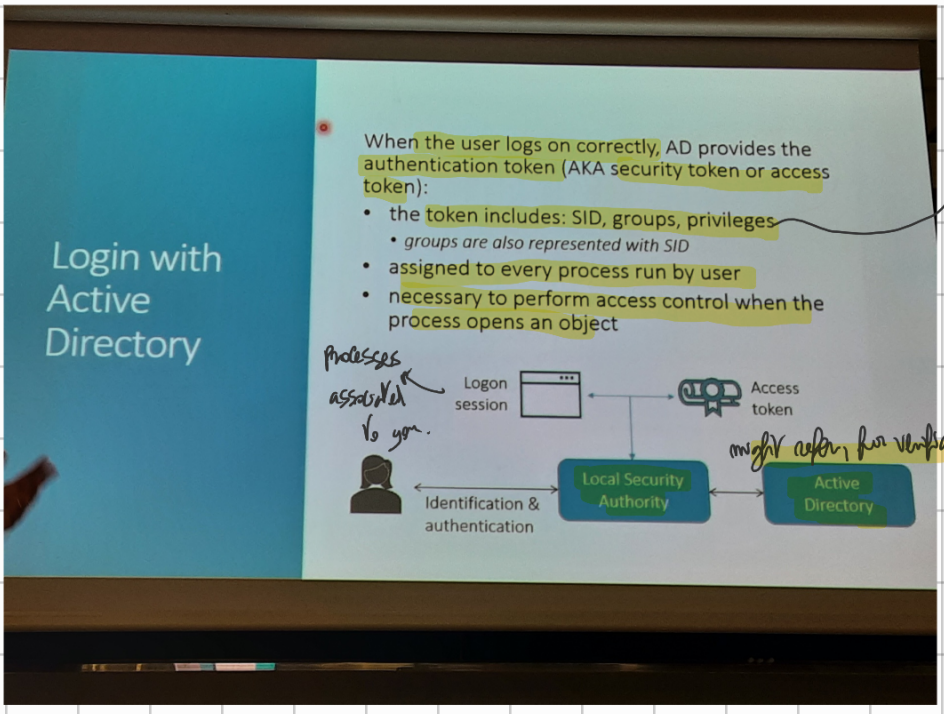
logins support several modalities:

username & password

username & smartcard

biometries

59



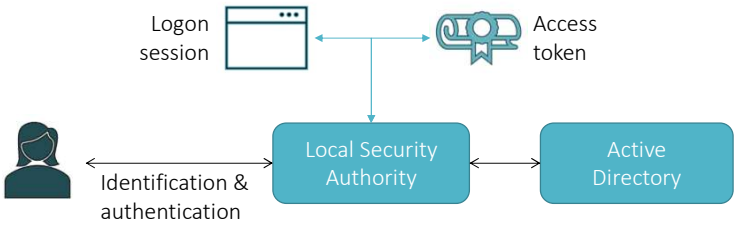
not access control lists.

The ACL on an object will tell if my SID can do what you're trying to do. There is not the SID.

Login with Active Directory

If the user logs on correctly AD provides an authentication token:

- the token includes: SID, groups, privileges
  - *groups are also represented with SID*
- assigned to every process run by user
- necessary to perform access control when the process opens objects



60

Note: in windows with local account you can have non-admins without PW.

Login with SAM (workgroup)

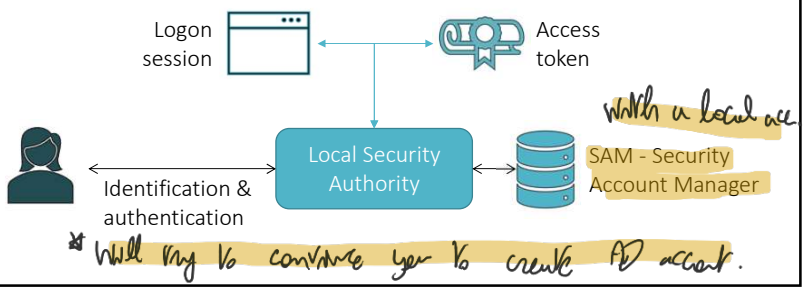
If the user has an account, it is associated to a Security ID (SID)

When the user enters username and passwd LSA generates the authentication token:

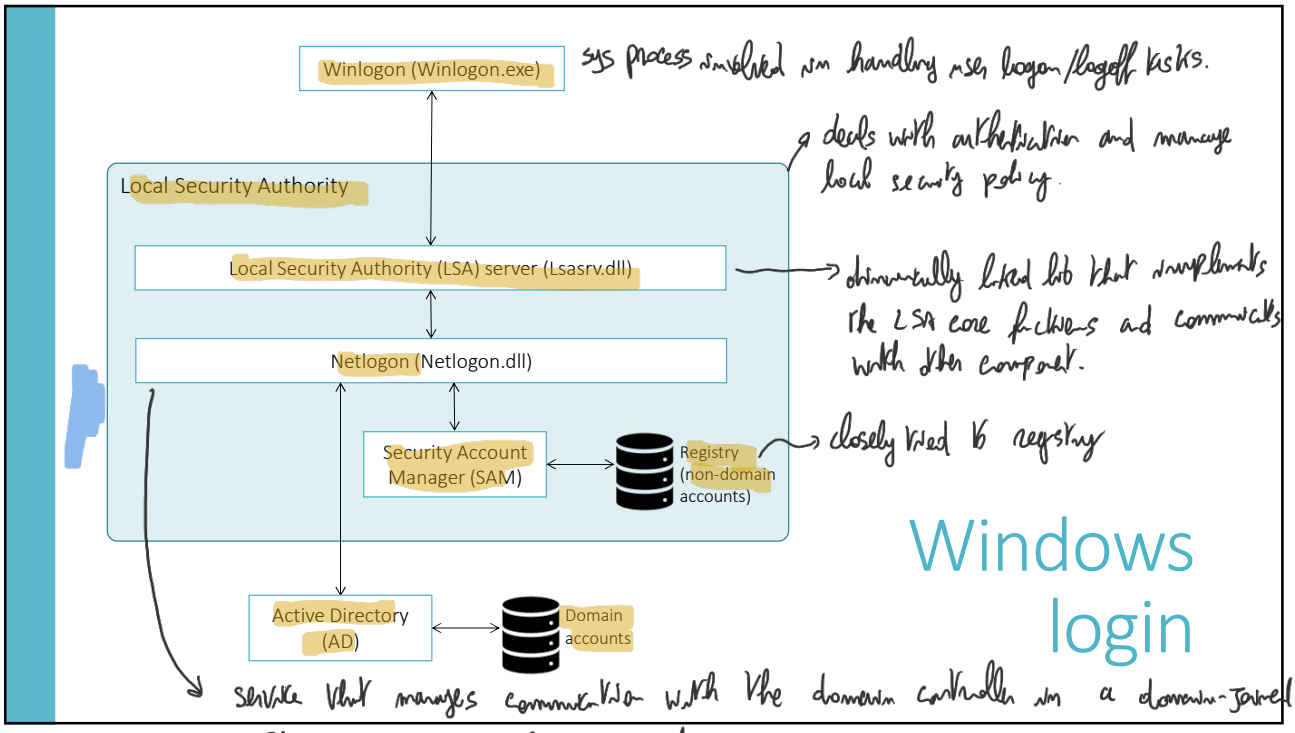
the token includes: SID, groups, privileges

Note: the user must already have a (local) account and an (optional) password

- optional passwd because in some settings user wants to avoid it...
- ... a potential security issue.
- no remote access without passwd anyway, and admin must have passwd
- also, the password is actively encouraged at setup
- domain accounts must always have a password

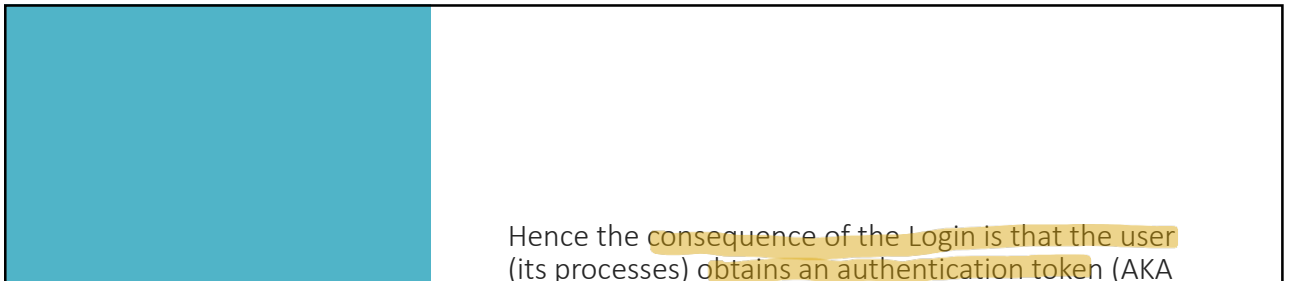


61



Windows login

62



Windows Login

Hence the consequence of the Login is that the user (its processes) obtains an authentication token (AKA security token):

- it represents the "security context" of the user: privileges and permissions that a user has
- it identifies the user (and his processes) in all subsequent interactions with securable objects…
- … and thus it is used to implement access control

63

Rights and privileges…

privileges,

security descriptors and access control lists,

mandatory access control and integrity levels

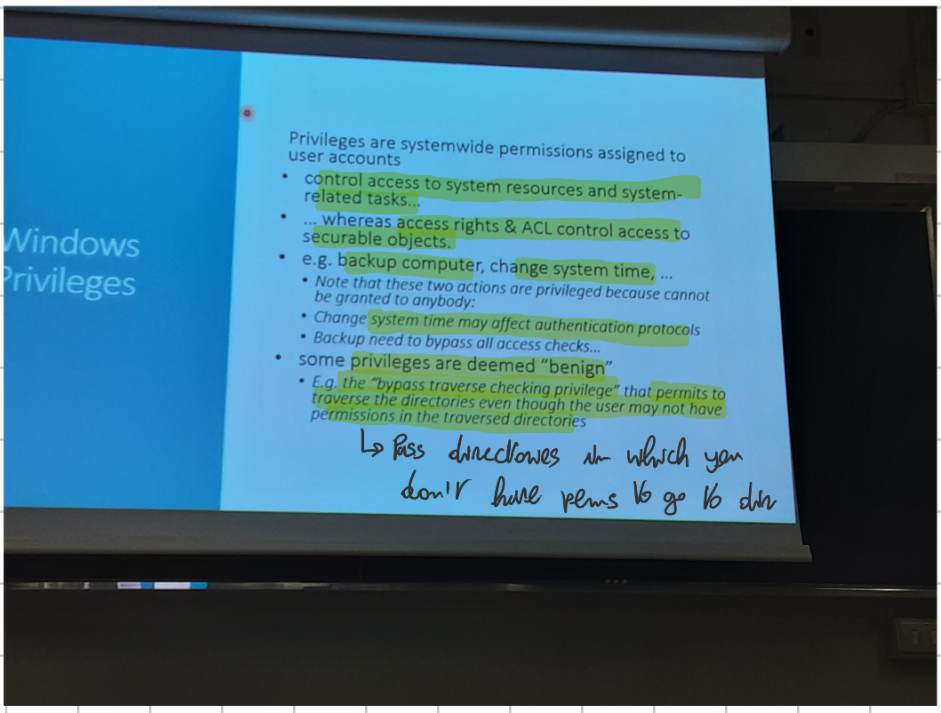
64

Windows Privileges

Privileges are systemwide permissions assigned to user accounts

- e.g. backup computer, change system time, …
  - *Note that these two actions are privileged because cannot be granted to anybody:*
  - *Change system time may affect authentication protocols*
  - *Backup need to bypass all access checks…*
- some privileges are deemed "benign"
  - *E.g. the "bypass traverse checking privilege" that permits to traverse the directories even though the user may not have permissions in the traversed directories*

65


Windows Privileges
Privileges are systemwide permissions assigned to user accounts
control access to system resources and system-related tasks...
... whereas access rights & ACL control access to securable objects.
e.g. backup computer, change system time, ...
Note that these two actions are privileged because cannot be granted to anybody:
Change system time may affect authentication protocols
Backup need to bypass all access checks...
some privileges are deemed "benign"
E.g. the "bypass traverse checking privilege" that permits to traverse the directories even though the user may not have permissions in the traversed directories
↳ Pass directories in which you don't have perms to go to dir in which you have them.

Windows Privileges

some privileges are deemed "dangerous" such as:

- act as part of operating system privilege
  - *AKA Trusted Computing Base (TCB) privilege*
  - *Grants the privilege to run as the most secure part of the system (the security code itself)*
  - *The most dangerous in Windows*
  - *Granted only to the Local System account (administrators do not have it)*
- debug programs privilege
  - *Allows to debug any program in Windows*
  - *Normally not needed by users*
  - *It implies the ability to run any code in any running processes…*
- backup files and directories privilege
  - *need to access the entire file system bypassing access controls*
  - *also to restore files and directories need to bypass access control and it is dangerous*

66

PS C:\Users\stefa> whoami /priv

PRIVILEGES INFORMATION

Privilege name	Description	State
SeShutdownPrivilege	System shutdown	Disabled
SeChangeNotifyPrivilege	Ignore cross-checking	Enabled
SeUndockPrivilege	Removing your computer from the housing	Disabled
SeIncreaseWorkingSetPrivilege	Increase a process working set	Disabled
SeTimeZonePrivilege	Changing the time zone	Disabled

strive: safety measure. If you need to use them you need to enable them to avoid making mistakes.

if a process requires more memory, it will need to request to enlarge working set

Powershell Example

67

Review question

SRM, LSA, SAM are all security components, but what are they in practice… kernel modules, system drivers, processes, threads, data structures, …?

Concerning the «SeIncreaseWorkingSetPrivilege», is there any vulnerability concern associated with this privilege? (you may look on the web…)

What does the first number in the SID mean?

68

ACL takes two forms: 1 is the usual one. For every user tells ops permitted
② in addition description of the security policy: info about auditing and
logging. Entries there specify whether access will need to be recorded.

Access Control Lists

Windows has two forms of access control list (ACL):

Discretionary ACL (DACL) — Defines ACL for every sys resource

- grants or denies access to protected resources (objects) such as files, shared memory, named pipes etc.

② System ACL (SACL)

- used for auditing – enables the log of attempts to access an object. An entry in SACL:
  - *specifies the types of access attempts that generate audit reports in the security event log.*
  - *identifies a trustee, a set of access rights, and a set of flags*
  - *flags: generate audit records when an access attempt fails, when it succeeds, or both.*
- also used to enforce mandatory integrity policy

Contains entries related to mandatory AC.

69

In every object that contains ACL.

Access Control Lists

ACL organized as list of ⓔ. Element of the list.

objects needing protection are assigned a DACL (and possibly a SACL) that includes e list of access control entries (ACEs)

each ACE includes a SID and an access mask:

- The SID specifies a user or a group (identify an active entity)
- The access mask could include ability to read, write, create, delete, modify, etc. Richer than the unix basic one.
- access masks are object-type specific (can also be specialised to the object)
  - *e.g. service abilities are create, enumerate*

70

Security Descriptor (SD)

- The Security Descriptor (SD) is a data structure that contains object owner, group, DACL, & SACL (if present)
- each "securable object" has its own SD
  - a securable object is any system resource (file, directory, registry entry, process, thread, pipes, etc…) that need to be protected

Example of an SD:

```
Owner: CORP\Blake
Group: CORP\Clerks
ACE[0]: Allow CORP\Blake Full_Control
ACE[1]: Allow CORP\Paige Full_Control
ACE[2]: Allow Administrators Full_Control
ACE[3]: Allow CORP\Cheryl Read, Write, Delete
```

group to which object belongs (associated to SID)

- This gives full control to users: Blake (who is the owner), Paige and Administrators
  - In new versions of Windows it is possible to limit full control of the owner, and owner too should be included in the DACL
- There is no implied access, if there is no ACE for a user, then the access to the object by processes of that user is denied
- Processes must request correct type of access
  - if just request "all access" when need less (e.g. read) and when not all is not allowed, access will be denied

71 When analyzing an ACL, operation is permitted only when permission is specifically included.

Can inspect ACL with this command on powershell

To obtain an ACL of an object: **get-acl**

output in a pipe to Format-list program

```
PS C:\Users\stefa> get-acl c:\Windows | Format-List

Path   : Microsoft.PowerShell.Core\FileSystem::C:\Windows
Owner  : NT SERVICE\TrustedInstaller
Group  : NT SERVICE\TrustedInstaller
Access : CREATOR OWNER                  Allow  268435456
         NT AUTHORITY\SYSTEM            Allow  268435456
         NT AUTHORITY\SYSTEM            Allow  Modify, Synchronize
         BUILTIN\Administrators         Allow  268435456
         BUILTIN\Administrators         Allow  Modify, Synchronize
         BUILTIN\Users                  Allow  -1610612736
         BUILTIN\Users                  Allow  ReadAndExecute, Synchronize
         NT SERVICE\TrustedInstaller    Allow  268435456
         NT SERVICE\TrustedInstaller    Allow  FullControl
         ...
Audit  :                                       // audit data for the SD from the system access control list (SACL).
Sddl   :                                       // it's the security descriptor in the SDDL syntax
```

access mask, will be discussed later with the File System

not admin, it is the one that installs the Windows OS

ACEs

flags describing privileges. If few allowed, we explicitly tells them. Otherwise we have a mask

System control list

Powershell Example

Audit info + Sddl: compact form to represent an ACL with a syntax specific to windows to program ACL. Useful when creating an object.

72

More on Security Descriptors & Access Control

- each ACE in the DACL determines access:
  - either *allow* or *deny*
- Windows evaluates each ACE in the ACL until access is granted or explicitly denied
  - hence deny ACEs come before allow ACEs
  - order by default if set using GUI
  - … but the order is up to programmer if set by program
- when user attempts to access a protected object, the OS performs an access check
  - comparing user/group info with ACE's in ACL
  - access granted if all requested operations are granted; else access is denied

73 One thing specific to Windows: ACL can specifically deny operation. Redundant. DAC! So owner or other users can modify security descriptor. So it is possible that eventually you allow one specific op. and then you deny it. This has two entries in conflict. That can be direct (either user entry, or user allowed, group denied). For Win, earliest has priority. We scan ACL until we can make a decision. Access control Entries evaluated. So positions are important. This is managed

automatically by a GUI, but also change ACL by code. So if you modify things through ACL, the GUI will automatically order the selection for you. In case of programming, order is not guaranteed.

More on Security Descriptors & Access Control

- In powershell it is possible to set the DACL and the SACL by means of set-acl
  - specify entry of ACL.
- It can also use the SDDL syntax to express the SD:
  - is just a text representation of a SD into a single string
  - can be converted into binary format to be used to set the SD to another object

↓ you can use also a binary format to be faster.

74

More on Security Descriptors & Access Control

↳ something similar to ABAC only by programming.

Windows also supports "conditional ACEs"

- allow application-level access conditions to be evaluated when an object is accessed
- Conditions on user/group attributes

↳ only for extremely specialized needs

For example, a conditional ACE may incapsulate the rule:

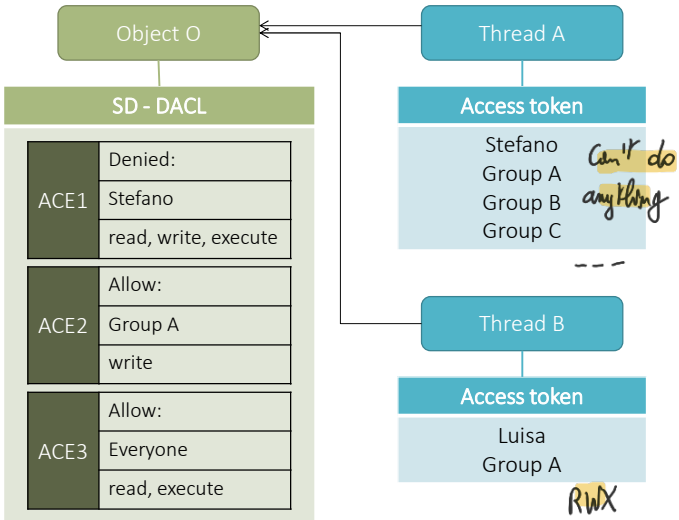
```
(Title=="Manager" && (Division=="Sales" ||
Division== "Marketing"))
```

↳ Attributes of objects of entity and user

… that expresses the fact that a user is a Manager in Sales or Marketing

Conditional ACEs cannot be set by GUI, can only be set by programs using SDDL

75



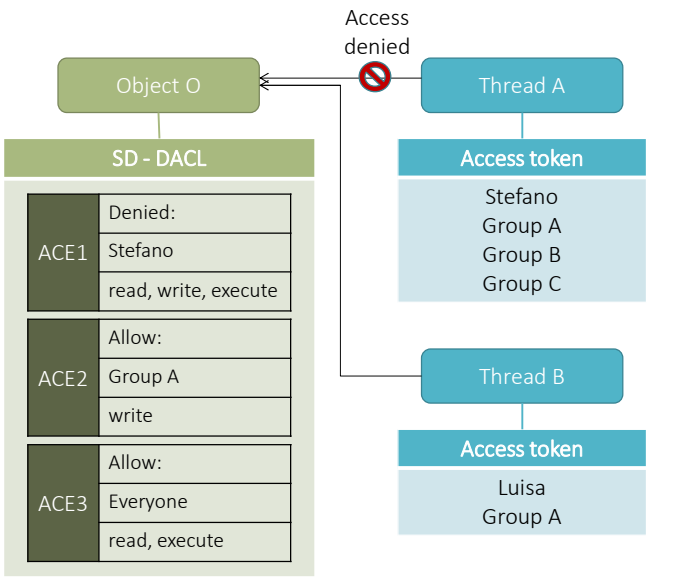
Question

Consider threads A, B and Object O.

What operations are permitted to thread A on object O?

What operations are permitted to thread B on object O?

76



Solution

Thread A:
- Access is denied
- The thread has the token of user Stefano, for which ACE 1 denies access in rwx
- The evaluation of the ACEs ends immediately at the ACE 1

Thread B:
- Access granted
- The thread has the token of user Luisa and of Group A
- ACE 2 grants w
- ACE 3 grants r,x

77

Initializing a SD is complicated. There are lots of parameters to setup. You don't do it by hands, mostly automated.

Initializing a security descriptor for a new object

Security Descriptor assigned and initialized at the creation of the object

Several API functions to build and initialize an SD from scratch

If the creator does not specify a SD, the object takes one inherited or the default one Many objects have parents.

SD Inheritance:

- many objects (directory service objects, files, directories, registry keys, etc.) have a parent object
- the system checks for inheritable ACEs in the security descriptor of the parent object…
- … and typically merges any inheritable ACEs into the ACLs of the new object's security descriptor.
- inheritance of DACL or SACL can be prevented by setting the SE_DACL_PROTECTED or SE_SACL_PROTECTED bits in the security descriptor's control bits.

But it is possible to specify, for one object if you want to inherit or not.

78

A user might ask for a service from the OS to do something. Impersonation is a solution. We want to allow the system to operate on the user resources. This is a specific privilege a thread should have. Thread can authenticate itself as if it was the user and work on their behalf.

Impersonation

Windows processes are multithreads

- common for both clients and servers
- each process runs as a specific account
- In case of servers however, it may be useful to let a thread to run as a different account
  - *to serve requests specific of that user*

Impersonation allows a server to serve a user, using his specific access privileges:

- for example, *ImpersonateNamedPipeClient* function sets user's token on the current thread to manage a named pipe as that user
- then access controls for that thread are performed against this token not server's…
- i.e. with user's access rights

To use impersonation a process must have the "Impersonate a client after authentication" privilege

- by default administrators and services accounts have this privilege

79

Js in form of integrity levels. On Windows, they wanted to safeguard integrity more than ever. In addition to ACL, a special focus was placed on integrity by adding special layer of AC.
It operates at gross grain: granularity of MAC, consider classes of obj and classes of users in a very broad sense. To enable generic integrity protection for parts of system. For example protect OS from external users. So even in case of vulnerability in ACL, additional layer.

Mandatory Access Control

ACL allow fine-grained control, but…

in addition Windows also have Mandatory Access Control called Integrity Control

- this limits operations changing an object's state
- each object and principal (user) is assigned an integrity level (stored in the SACL)
- there are 4 integrity levels in Windows
- a ~~process~~ thread of a given integrity level can only change state of objects of equal or lower integrity levels

MAC has a priority: first MAC check, then DAC check. ①
② prevents user messing up the data of another user.
User messing up with the sys prevented by both.

80

Mandatory Access Control

sys exes still with low integrity level.

When a user launches an executable file: → process will run threads

- the new process is created with the **minimum** of the user integrity level and the file integrity level ⇒ keep on the safe side.
  - i.e. the new process will never execute with higher integrity than the executable file.
  - i.e. If the administrator user executes a low integrity program, the token for the new process functions with the low integrity level.
  - this helps protect a user who launches untrustworthy code from malicious acts performed by that code:
    the user data, which is at the typical user integrity level, is write-protected against this new process.

81

Mandatory Access Control

objects and users are labeled as:

- Low integrity (S-1-16-4096)
- Medium integrity (S-1-16-8192)
- High integrity (S-1-16-12288)
- System integrity (S-1-16-16384)

Note the SID associated to the integrity levels, that's how Windows implements them:

- a high-integrity process will include the S-1-16-12288 SID in the process token
- processes or objects that do not have an integrity label are deemed at medium integrity

82

Mandatory Access Control

The SACL contains a specific ACE to keep the integrity SID of the object (if present)

- It's called SYSTEM_MANDATORY_LABEL_ACE
- It's just for Mandatory Access Control of securable objects
- Its access mask specifies the access that users with integrity levels lower than the object are granted.
- The values defined for this access mask are:
  SYSTEM_MANDATORY_LABEL_NO_WRITE_UP,
  SYSTEM_MANDATORY_LABEL_NO_READ_UP,
  SYSTEM_MANDATORY_LABEL_NO_EXECUTE_UP.
- by default, the system creates every object with an access mask of SYSTEM_MANDATORY_LABEL_NO_WRITE_UP.

83

Mandatory Access Control

when a write operation (a change of an object state) occurs:

- Windows first checks whether the subject's integrity level dominates object's integrity level…
- …if lower checks if the operation is permitted anyway by the integrity level mask
- If integrity check succeeds, and the normal DACL check also succeeds, then the write operation is granted

Note: much of OS marked medium or higher integrity

Example: Integrity levels to create a sandbox:

- Explorer (Edge) uses integrity levels to run potentially hostile code from the Internet Runs what it downloads from net at low integrity level.
- its process runs at low integrity level
- while the rest of the OS is marked medium or higher integrity

1) Download executable on the net
2) Execute that thread is lowest integrity level. But thread will still be able to interact with objects that don't have NO_READ_UP.

84

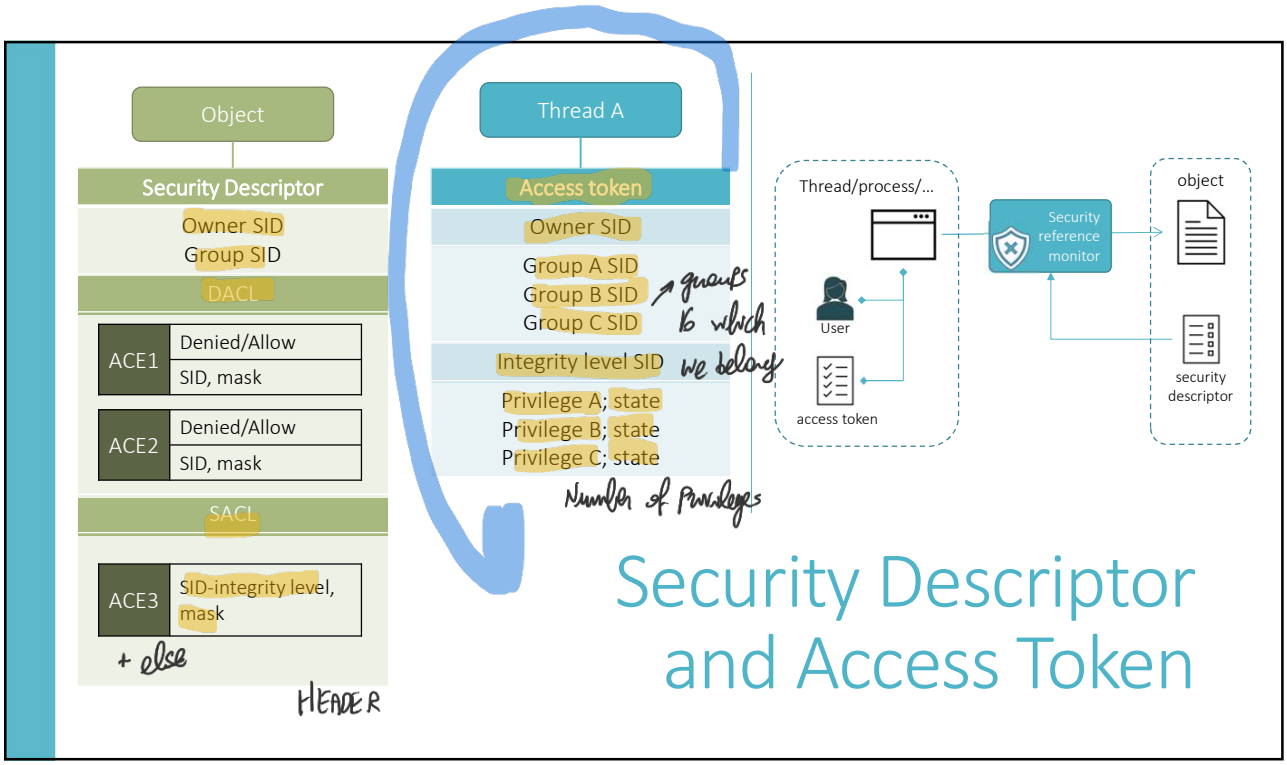
```
PS C:\Users\stefa> whoami /USER
USER INFORMATION
----------------
User Name   SID
=============== ============================================
tabatayoga\stefa S-1-5-21-1144226474-1424528306-2619936039-1001


PS C:\Users\stefa> whoami /GROUPS
GROUP INFORMATION
-----------------
Group Name                                 Type             SID                                            Attributes
========================================== ================ ============================================== ==================================================
Mandatory label\Medium Integrity level     Label            S-1-16-8192
Everyone                                   Known group      S-1-1-0                                        Mandatory, enabled, predefined,…
…
BUILTIN\Administrators                     Alias            S-1-5-32-544                                   Only for negotiation
BUILTIN\Users                              Alias            S-1-5-32-545                                   Mandatory, enabled, predefined,…
…
NT AUTHORITY\Auth. Users                   Known group      S-1-5-11                                       Mandatory, enabled, predefined,…
MicrosoftAccount\ste@outlook.it            User             S-1-11-96-3623454-…-2…5863                     Mandatory, enabled, predefined,…
NT AUTHORITY\Local account                 Known group      S-1-5-113                                      Mandatory, enabled, predefined,…
LOCAL                                      Known group      S-1-2-0                                        Mandatory, enabled, predefined,…
```

The user has a medium integrity level

it's similar to the concept of groups

Powershell Example

85



86

Review question

If a user with integrity level S-1-16-8192 requests an access to an object of integrity level S-1-16-4096, will the access be granted?

Consider a user U with integrity level S-1-16-8192 that has a privilege granting access to object O. If the object has integrity level higher than that of U, will U be allowed to access O?

↳ Depends on the mask.

Does Windows uses a protection matrix?

↳ One possible representation of the Access Control. But windows is using lists, so not uses the matrix as data structure. ACEntries represented in a list, not a matrix.

87

File system security : in FS we have also very specific obs.

ACL in the FS

Privileges

auditing

88

File system security

→ modern OSes can manage several FS. So FS itself is implemented within a driver. FS is implemented in supervisor mode. So

The file system itself is a driver, hence all considerations for driver security holds:

- operations initiated by a driver bypass most security checks

It's a part of the OS.

However, unlike most other types of drivers, file systems are intimately involved in normal security processing.

- this is because of the nature of security and its implementation within Windows.

89

You may deal with a lot of FS. Security policy might depend on the FS itself.
For ex. FAT doesn't have security. Driver implements security.
NTFS is native of Windows. Still implemented with a driver but coordinated with Windows.

File system security

The specific granularity of security control is entirely up to the file system

- In NTFS files and directories are objects
- hence all the considerations concerning DACL and SACL also holds for NTFS
- In particular, it supports a per-file (or directory) security descriptor model.

That's not true for all FS supported in Windows:

- For example, FAT, CDFS, UDFS do not support security descriptors.

Here we focus on NTFS

90

File system security – security descriptor

The files (and directories) security descriptor is one of the file's attribute in its MFT record

identical to many other objects managed by OS.

The security descriptor contains the usual information:

- SID of the file (or directory) owner
- SID of the group of the object
- DACL
- SACL

Note: an object's owner always has the ability to reset the security on the object.

- this allows to remove all access to an object
- even if owners remove their ability to perform all operations, this inherent right allows them to restore their security rights on the object.

91

File system security – access control list

auditing policy is particularly important with files

NTFS access control lists provide a discretionary access control environment

- hence the owner of an object is allowed to grant access to the object

DACL contains a list of Access Control Entries (ACE) that describes the access policy of the security descriptor (discretionary access control policy)

SACL contains a list of ACE that describe the auditing policy of the security descriptor

But Mandatory Access Control implemented with the integrity levels also present in NTFS

92

File system security – access control entries

each ACE defines describes the access rights associated with a particular SID

access rights in a compact form represented by means of a 32-bit access mask

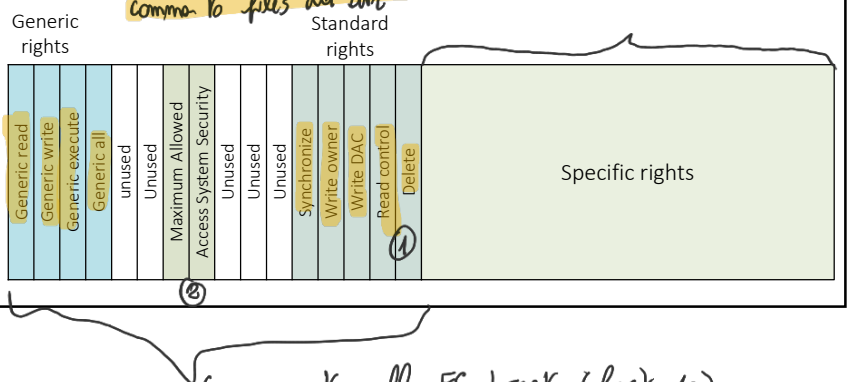
the mask takes different meanings depending on the object it is associated

For FS objects:

- generic rights (4 bits)
- standard rights (5 bits)
- specific rights (16 bits)
- right to access SACL (1 bit)
- other bits reserved or not used

32 bits mask for rights

common to files and dirs



93

common to all FS objects (first 16)

① Do not concern simple method of access to object, but to access to description of object: synchronize locks. ② One for Access Sys Sec. Allows thread to access the object ...

File system security – access control entries

Generic rights (4 bits):

- GENERIC_READ – the right to read the information in the object
- GENERIC_WRITE – the right to write the information in the object
- GENERIC_EXECUTE – the right to execute the object
- GENERIC_ALL – read, write and execute together
- can be combined together, same as rwx in Unix

94

File system security – access control entries

Standard rights (5 bits)

- DELETE—the right to delete the particular object.
- READ_CONTROL—the right to read the control (security) information for the object.
- WRITE_DAC—the right to modify the control (security) information for the object.
- WRITE_OWNER—the right to modify the owner SID of the object. Recall that owners always have the right to modify the object.
- SYNCHRONIZE—the right to wait on the given object (assuming that this is a valid concept for the object)

95

File system security – access control entries

Specific rights for files:

- FILE_READ_DATA—the right to read data from the given file.
- FILE_WRITE_DATA—the right to write data to the given file (within the existing range of the file).
- FILE_APPEND_DATA—the right to extend the given file.
- FILE_READ_EA—the right to read the extended attributes of the file.
- FILE_WRITE_EA—the right to modify the extended attributes of the file.
- FILE_EXECUTE—the right to locally execute the given file. Executing a file stored on a remote share requires read permission, since the file is read from the server, but executed on the client.
- FILE_READ_ATTRIBUTES—the right to read the file's attribute information.
- FILE_WRITE_ATTRIBUTES—the right to modify the file's attribute information.

96

File system security – access control entries

Specific rights for directories:

- FILE_LIST_DIRECTORY – list the contents of the directory.
- FILE_ADD_FILE – create a new file within the directory.
- FILE_ADD_SUBDIRECTORY – create a subdirectory within the directory.
- FILE_READ_EA – read the extended attributes of the given directory.
- FILE_WRITE_EA – write the extended attributes of the given directory.
- FILE_TRAVERSE – the right to access objects within the directory.
- FILE_DELETE_CHILD – delete a file or directory within the current directory.
- FILE_READ_ATTRIBUTES – read a directory's attribute information.
- FILE_WRITE_ATTRIBUTES – modify a directory's attribute information.

97

File system security – privileges

FS also support privileges: the cross privilege is an example.

Privilege is a separate mechanism wrt ACL and integrity levels

Each privilege is associated to particular operations that may be performed if the privilege is held and enabled by the caller.

note the two conditions here:

the privilege must be held by the caller.

the privilege must also be enabled.

The privilege must be enabled prior to its use

… rather than simply assumed

98

File system security – privileges

enable this only when you need it

Restoring a backup is critical

Example: the SeRestorePrivilege privilege:

- allows a user to bypass the usual checks for write access to a file.
- an administrator may not wish to actually override the normal security checks when copying a file…
- but would wish to do so when restoring that same file using a backup/restore utility

Normally the administrator operates without this privilege.

It enables this privilege only when it needs it

Minimizes the chance a user might inadvertently perform an operation they did not intend

99

File system security – privileges

Several privileges are associated to the file system. The main are:

- **SeBackupPrivilege –** allows file content retrieval
  - *even if the security descriptor on the file might not grant such access*
  - *A caller with this privilege enabled obviates any ACL-based security check*
- **SeRestorePrivilege –** allows file content modification
  - *even if the security descriptor on the file might not grant such access*
  - *this function can also be used to change the owner and protection*
- **SeChangeNotifyPrivilege –** allows traverse right.
  - *it is an important optimization in Windows,*
  - *the cost of performing a security check on every single directory in a path is obviated by holding this privilege.*
- **SeManageVolumePrivilege –** allows specific volume-level management operations
  - *such as lock volume, defragmenting, volume dismount etc.*

operate at low level with disk.

↳ critical: at low level, making files configurations to speed up access to them.

100 Windows supports auditing, and

File system security – auditing

The auditing system provides a mechanism for tracking specific security events

- the resulting logs can be analyzed off-line to perform post-mortem analysis of a damaged or compromised system.
- auditing intimately involves the file system because it maintains the persistent storage of system data.
- when security needs are low, auditing can be disabled. Some FS (like FAT) do not implement auditing

NTFS implements auditing

- Several tools to analyze audit logs
- In Windows Event Visualizer (Eventvwr.exe)

101

Services security

Services in Windows correspond to daemons in Unix

The Service Control Manager:

- is a component of the executive that keeps track of services
- keeps a database on the installed services and their configurations
- the registry key of the DB is:
  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services

Each service runs in the security context of a user account:

- it's a user account specific for the service
- when it starts it logs on with the credential of the user account…
- … and it thus obtain the corresponding security token

for ex. services concerning network run with SID of user.

102

Review question

Where is the security descriptor of a file stored?

What is the purpose of privileges?

What is a tool that you can use to inspect audit logs?

103

Windows Vulnerabilities…

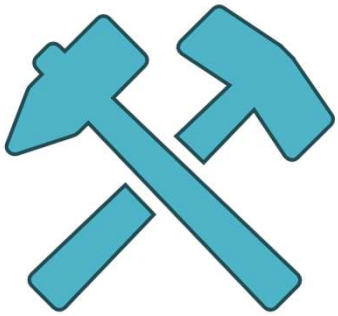
Windows, like all OS's, has security bugs

and bugs have been exploited by attackers to compromise customer operating systems

OS is one of the most complex pieces of SW ever written.

104

… and Hardening Policy



Microsoft now uses process improvement called the Security Development Lifecycle Software Engineering

net effect approx 50% reduction in bugs

Windows Vista was the first to use security development lifecycle start to finish

IIS v6 (in Windows Server 2003) had only 3 vulnerabilities in 4 years, none of them critical

= OS that comes to cover a wide spectrum of use cases. Achieved by improving development technology with the purpose of reducing bugs and vulnerabilities.

105

Windows Security Development Lifecycle

The core Security Development Lifecycle are as follows:

- Mandatory security education
- Secure design requirements
- Threat modeling
- Attack surface analysis and reduction
- Secure coding requirements and tools
- Secure testing requirements and tools
- Security push
- Final security review
- Security response

Note: this does not mean bug free!

106

Windows Security Defenses

nowadays attackers are criminals and are highly motivated by money

Windows defenses are grouped into four categories:

- account defenses
- network defenses
- buffer overrun defenses.
- browser defenses

107

Key points: reduce attack surfaces, especially because of the scope of the OS. Hardening is difficult also because users can install whatever.

Windows System Hardening

process of shoring up defenses, reducing exposed functionality, disabling features

- known as attack surface reduction
- use 80/20 rule on features: if not used by 80% population it should be disabled…
- … but it's not always achievable, may result in a system not usable for non-technical users
- e.g. requiring RPC authentication in XP SP2
- e.g. strip mobile code support on servers

Hardening

= hardening example. You won't need support for mob. apps like browsers.

servers easier to harden:

1. are used for very specific and controlled purposes
2. server users are generally administrators with better computer configuration skills than typical users

108

Users come with SIDs for DAC. But there was an evolution. Early versions of Windows NT

Account Defenses

some user accounts can have privileged SIDs

- E.g administrators

least privilege dictates that users operate with just enough privilege for tasks

↳ weakened security because old sw couldn't run without admin priv.

in Windows XP users normally operate as local Administrators

- for application compatibility reasons, most apps for previous Windows would not work otherwise
- Although XP introduced "Secondary Logon" to run apps with other user privileges (option "run as…")
- … and it also introduced restricted tokens to limit per-thread privilege

From Windows Vista this all reversed with User Control Account (UAC)

- by default, all user accounts are users and not administrators
- when a user wants to perform a privileged operation it is prompted to introduce admin credentials
- … unless it is an administrator, in which case it is notified to give consent to the operation

109

Low Privilege Service Accounts

Windows services are long-lived processes started after booting

- many ran with elevated privileges
- but many do not need elevated requirements

Windows XP introduced Local Service and Network service accounts

- allow a service local or network access
- but with a very low privilege level

110

Low Privilege Service Accounts

Another example of least privilege principle is the RPC service:

- it used to run with high privilege (with System identity)
- just to let DCOM built on top of it to run on a remote computer correctly
- but RPC itself did not need high privileges

From Windows XP it is split in two (RPCSS and DCOM Server Process)

- RPCSS runs with Network service account with low privileges
- DCOM Server Process runs as System

Apache, OpenSSH and others also use this model:

- small amount of code with elevated privileges
- most of the code with low privileges

111

Stripping Privileges

another defense is to strip privileges from an account soon after an application starts

- e.g. Index server process runs as system to access all disk volumes
- but then sheds any unneeded privileges as soon as possible
- using AdjustTokenPrivileges

Windows can define privileges required by a service by using ChangeServiceConfig2 function

112

Network Defenses

Windows have IPSec and IPv6 with authenticated network packets enabled by default

IPv4 also enabled by default, expect less use

Windows have built-in software firewall

block inbound connections on specific ports

Vista can allow local net access only

optionally block outbound connections

default was off (XP) but then default since Vista

113

Memory Corruption Defenses

They do not provide support from buffer OF, so we need defenses for these.

Most OS code and many software is written in C/C++

- as already discussed, C was designed as a high-level assembly
- gives direct memory access to the programmer
- for example:

```
Char password[32];
Char *p=password;
```

- with this flexibility come risks: the ability to corrupt memory

Rewriting the OS in Java or C# is not an option of course

- … and it would not solve the real problem, that programmers have too much trust on the data they receive

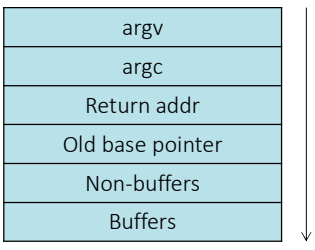
Hence many OS introduce defenses against memory corruption. Windows is not an exception

114

Stack-based Overrun Detection

The figure shows a conventional structure of a stack (only the portion corresponding to the invocation of a function

- Non buffers are like variables, that may also contain pointers to data structures
- Buffers may be subject to buffer overrun attacks



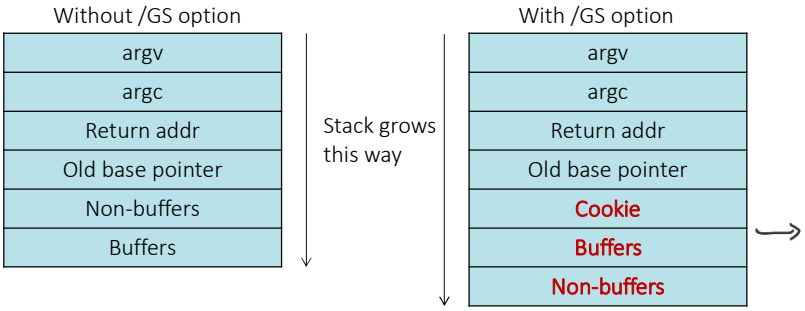
Same order as presented, but this is not always.

115

Stack-based Overrun Detection

Windows compiler (Visual C++) offers the /GS option when you compile code, which does two things:

1. Inserts a random number (Cookie AKA random canary) between all variables allocated in the function
   this number is checked at the end of the function: if it is changed: buffer overrun, abort process
2. Reverses the placement of non-buffer variables and buffers
   prevents buffer overrun from overwriting non-buffers, which are sensible variables (like pointers to data structures)



this way you won't destroy no buffer vowables.
+ Random Cookie.

116

Other Memory Corruption Defenses

Prevents code executing in data segments
- it's a control introduced in modern CPUs (AMD, Intel,…)
- … and exploited in Windows since XP and Vista
- … as commonly used by buffer overrun exploits

Stack Randomization (since Vista) Randomize stack position
- randomizes thread stack base addresses
- makes impossible for the attacker to predict where the stack will be and thus set its shellcode appropriately

Heap-based buffer overrun defenses:
- add and check random values (cookies) on each heap block and checks heap integrity (since XP)
- also introduce heap randomization: places the start of the heap at a random offset (0-4MB) (since Vista)

117

OS loaded in virtual memory of every process. This position can be randomised.

↑ Randomization

Other Defenses

OS Image Randomization

- OS boots in one of 256 configurations
- i.e. the entire OS is shifted up or down in memory when it is booted
- makes OS less predictable for attackers

Service Restart Policy — To make sys useful for most users

- services can be configured to restart if fail
- great for reliability but lousy for security
- Since Vista, some critical services so can only restart twice, then manual restart is needed
- **gives attacker only two attempts**

With privilege escalation, service would be discharged.

118

Browser Defenses

web browser is a key point of attack — Protection: running browser at low integrity level

- via script code, graphics, helper objects
  - *runs ActiveX controls, Flash, Java applets, .NET apps*
  - *renders various multimedia objects, mp3/4, JPEG, BMP,…*
  - *Invokes helper objects (MIME) to manipulate data formats (Windows Media Player, Quicktime, etc…)*

Microsoft added many defenses to IE7

- ActiveX opt-in
  - *unloads ActiveX controls by default*
  - *when any then first run prompts user to confirm*
- protected mode
  - *IE runs at low integrity level (see earlier)*
  - *so more difficult for malware to manipulate OS*

119

Encrypting file system

… a number of low-level crypto functionalities for encryption, hashing, signing…

① Encrypting File System (EFS)

allows files / directories to be encrypted / decrypted transparently for authorized users

the administrator just set the encryption property for a directory

- from that point on any file in the directory is encrypted

generates random File Encryption Key (FEK)

- the key is protected by DPAPI (see next slide)

to grant access to an encrypted file to another user:

- the FEK itself is encrypted with the user encryption key and stored along with other files' metadata in the MFT

EFS also support recovery if the FEK key is lost

Files can be encrypted. Key is generated automatically. Good practice to copy it and take it in a safe space.

120

Data protection API (DPAPI)

Data Protection API (DPAPI)

- Allow users to encrypt and decrypt data transparently
- The management of encryption keys (maintaining, protecting,…) is removed from the users and given to the OS
- Keys generated automatically by the OS and derived in part from user's password
- Developers need only to call *CryptProtectData* to encrypt and *CryptUnprotectData* to decrypt

121

In modern systems encryption is available at Hardware level.
↳ at SW you don't care. Key is kept in hardware on a chip. Copy of this key is kept by microsoft. So for investigation can give you this possibility.

Trusted Platform Module (TPM)

It's a hardware solution to enhance security, from a specification of the Trusted Computing Group

moves many sensitive cryptographic operations into hardware.

Windows uses TPM to validate that Windows itself had not been tampered with

- this is known as trusted boot, or secure startup
- as the OS boots, critical portions are hashed and the hashes verified.

Another use of TPM is to encrypt entire File System (next slide)

Use of Trusted Platform Module (TPM)

122

BitLocker Drive Encryption

especially useful to protect data disclosure on stolen laptops

It is a policy that can be set locally or on the Active Directory

encrypts an entire volume with AES and almost no performance degradation

key either on USB or on a chip in the motherboard (the Trusted Platform Module – TPM) or in the Active Directory

- BitLocker also supports key recovery

When booting a system the key must be available

- either the USB drive with the key must be connected
- or the key must be available in the TPM or AD

Bitlocker different than EFS:

- EFS need explicit management, for each single file/directory
- Bitlocker is "set and forget" and operates on an entire volume

BitLocker Drive Encryption

123

MANDATORY ACCESS CONTROL IS GROSS GRAINED SECURITY and ORIENTED AT INTEGRITY

You can check if disk is encrypted with BitLocker:

Example with powershell (run as administrator):

```
PS C:\WINDOWS\system32> manage-bde -status
Crittografia unità BitLocker: strumento di configurazione versione 10.0.19041
Copyright (C) 2013 Microsoft Corporation. Tutti i diritti sono riservati.

Volumi del disco che possono essere protetti con
Crittografia unità BitLocker:
Volume C: [Windows]
[Volume del sistema operativo]

    Dimensioni:                          952,62 GB
    Versione BitLocker:                  2.0
    Stato conversione:                   Crittografia del solo spazio utilizzato
    Percentuale completamento crittografia: 100,0%
    Metodo crittografia:                 XTS-AES 256
    Stato protezione:                    Protezione attivata
    Stato blocco:                        Sbloccato
    Campo identificazione:               Sconosciuto
    Protezioni con chiave:
        TPM
        Password numerica
```

BitLocker Drive Encryption

124

Review question

Can you say that encrypting a disk is a method of system hardening?

Motivate your anwser

125

Summary

- Windows architecture
- Windows security architecture
- vulnerabilities
- security defenses — account, network, buffer, browser
- crypto services

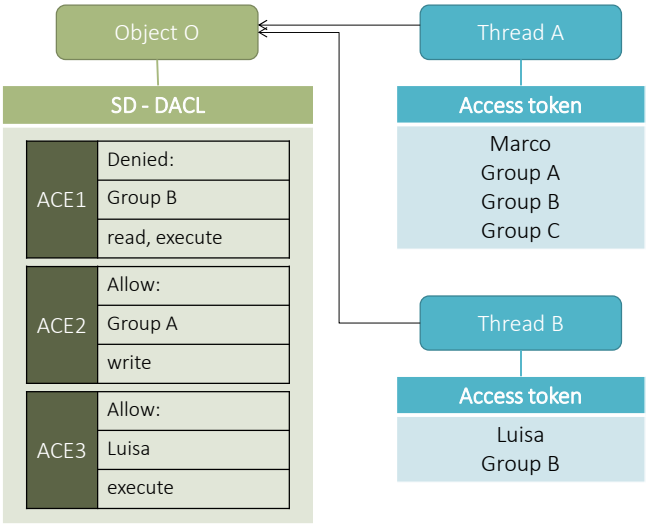
126

Exercise

Consider threads A, B and Object O in Windows.

What operations are permitted to thread A on object O?

What operations are permitted to thread B on object O?

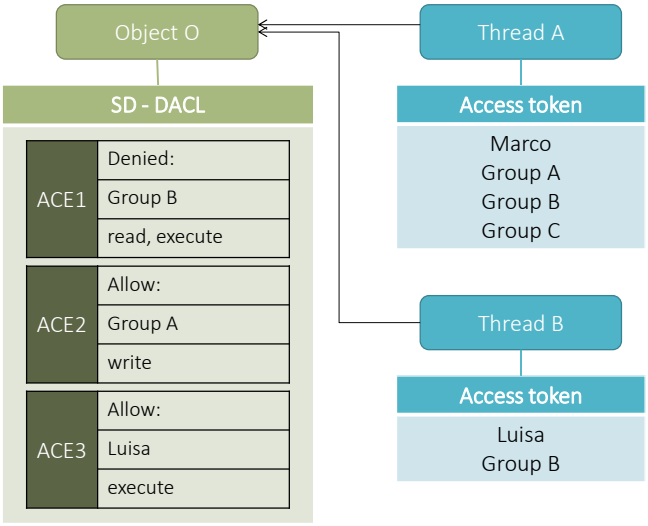


128

Solution

Thread A:
- Access is denied read and exec
- The thread has the token of Group B, for which ACE 1 denies access in rx
- It is permitted write, as the thread belongs to group A

Thread B:
- Access is denied read and exec by means of ACE1 (thread of group B)
- ACE3 does not have effect in this case
- … hence no access is possible



129

A: read

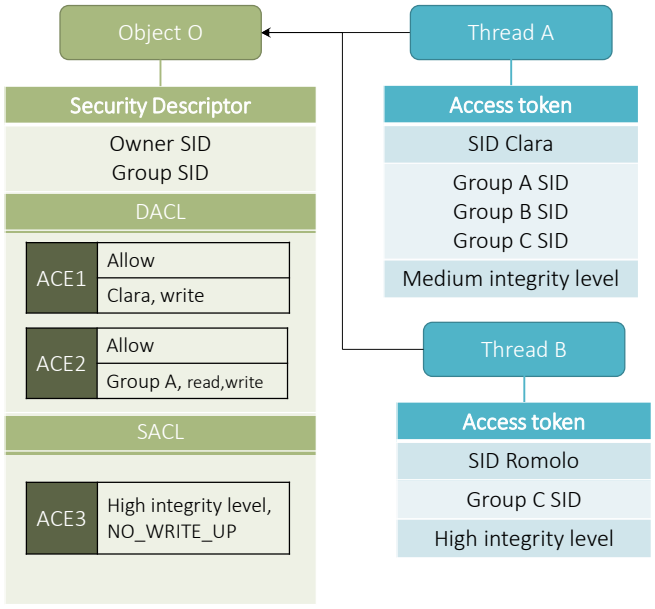
B: No op

Exercise

Consider threads A, B and Object O in Windows.

What operations are permitted to thread A on object O?

What operations are permitted to thread B on object O?



130

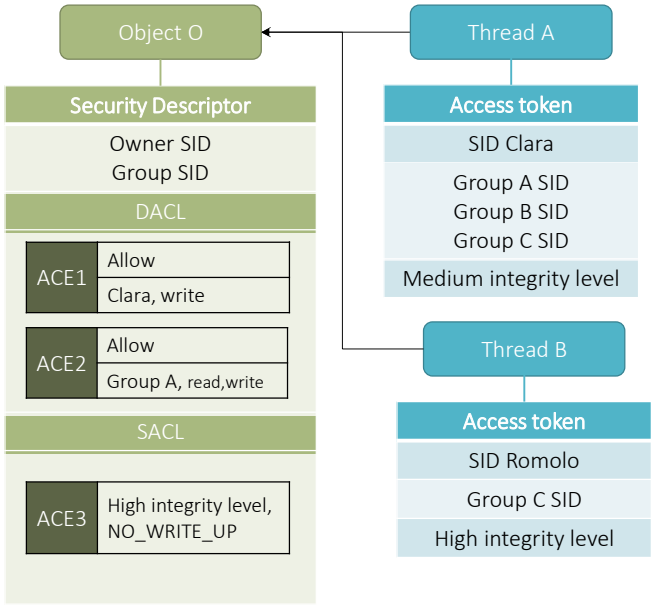
Solution

Thread A:

- SYSTEM_MANDATORY_LABEL_ACE (ACE3) in the SACL denies write access
- The thread has medium integrity level and the token of Group A, for which ACE 2 gives access in RW
- Hence it is only permitted to read

Thread B:

- Romolo has the same integrity level, but the DAC (ACE1 and ACE2) do not allow any operation to Romolo.



131