







1

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

First version called Windows NT

- NT stands for “new technology”, to mark difference from old MS-DOS
- 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

Windows

4

Windows is a 64-bit Operating System

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
- 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

Windows

5

Access control lists (ACLs)

- implement Discretionary Access Control

Integrity levels

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

Several mitigations for exploits that include:

- file system and communications encryption
- 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

Security Principles

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

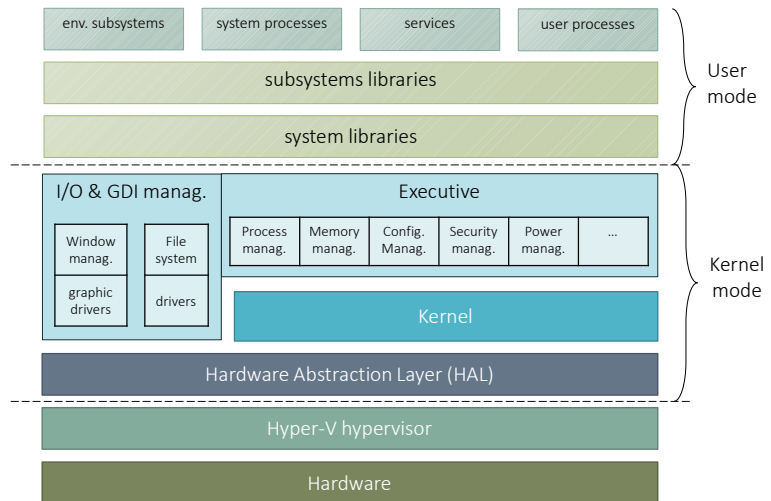
Reliability

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

Design Principles

8

Windows architecture



9

System components at each level operate under specific privilege layers

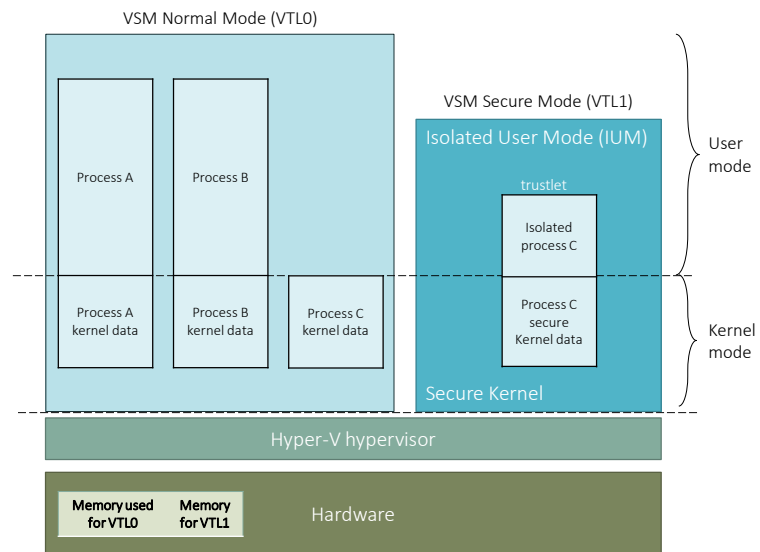
Beyond kernel and user mode, exploits virtualization (Hyper-V) to implement **virtual trust levels** (Win 10 feature):

- 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

10

Virtual trust levels (VTL)



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

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

To its purposes defines:

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.

Kernel

12

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

Process and Threads

13

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

Inter-process communications

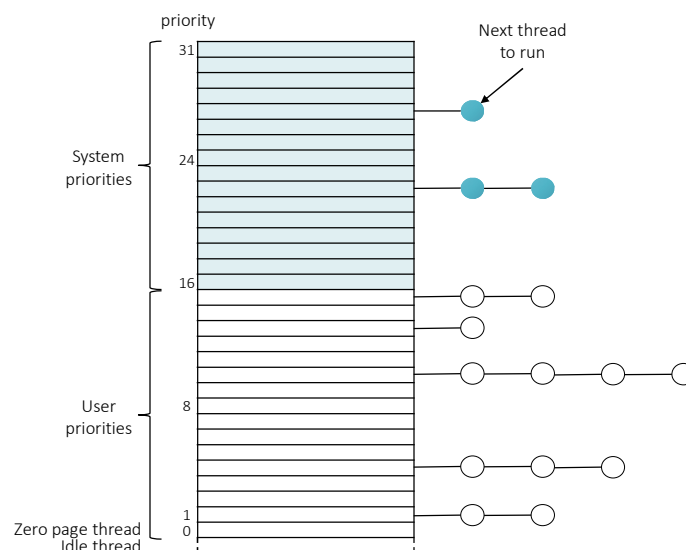
14

Pre-emptive, priority-based scheduling of threads:

- multi-level feedback queues with 32-levels of priority
- priorities from 16 to 31 for system threads
- 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-bound threads against CPU-bound threads

Threads scheduling

15



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)
- When a window goes in foreground the time share of its threads is enlarged

Threads scheduling

16

Windows assumes the underlying HW platform supports virtual memory with paging

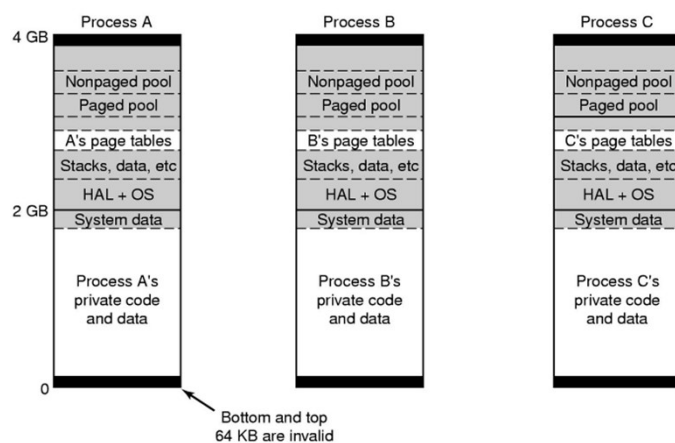
- page sizes defined by hardware (typically 4KB, can also have pages of 2MB or 1GB)
- multilevel page tables (number of levels depending on the addressing bit)
- working set page-replacement algorithm
- page states: valid, zeroed, free standby, modified and bad

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

Memory Management

17



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)

Virtual Memory

18

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

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

committed : if allocated to a region and in use

Memory Management

19

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

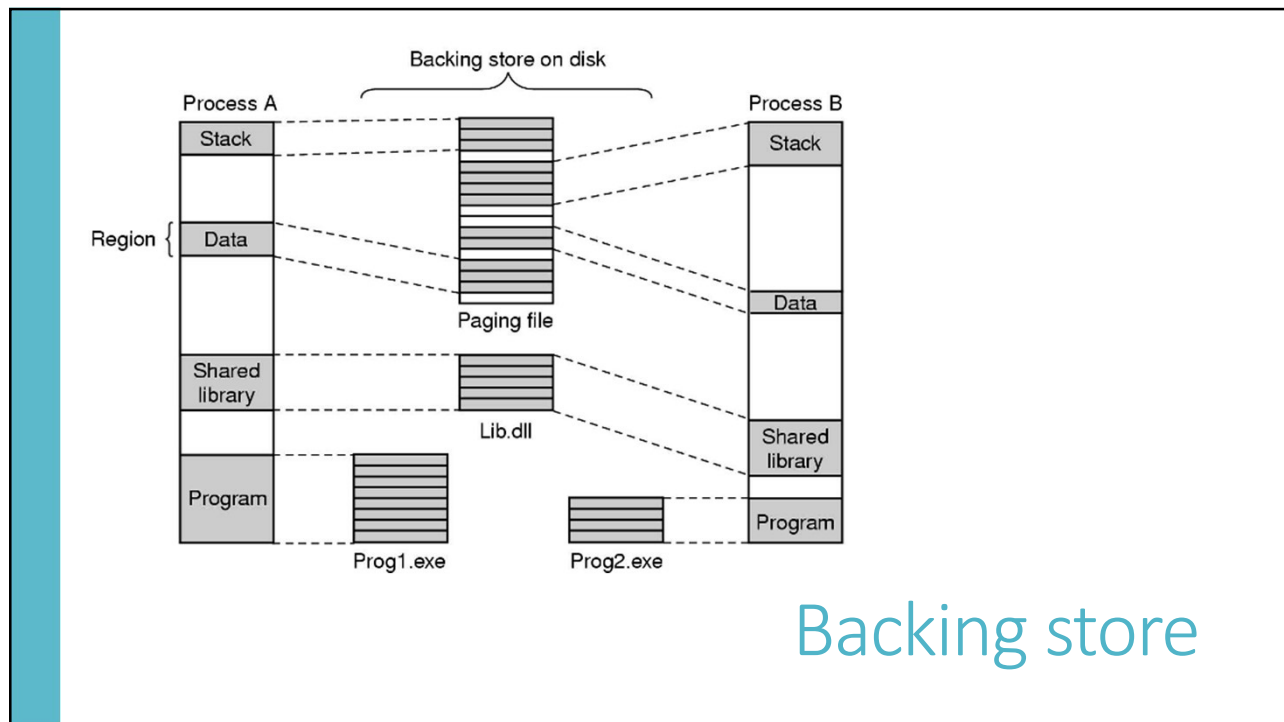
- two processes can share memory by mapping the same file into their respective virtual memories

The heap is a virtual memory region of reserved address space

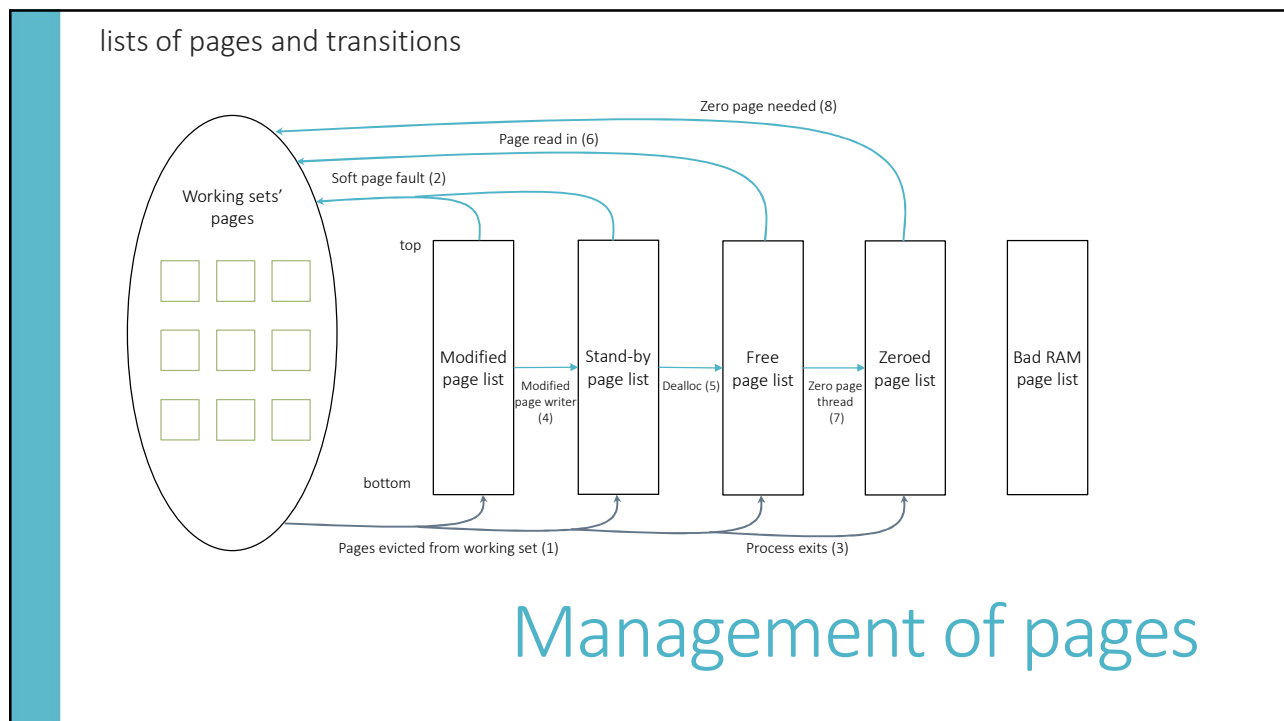
- by default a process starts with 1MB of heap
- heap shared by the process thread, hence its allocation is synchronized to prevent race conditions

Memory Management

20



21



22

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

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:

- 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 hibernation

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

System boot

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

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
...

Kernel objects

27

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
- 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

- a process specifies access rights when it creates an object
- ... and may change them later...

Kernel objects

28

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

Kernel objects

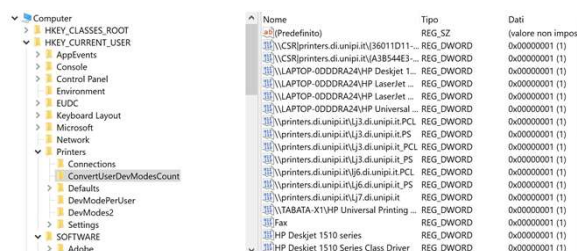
29

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

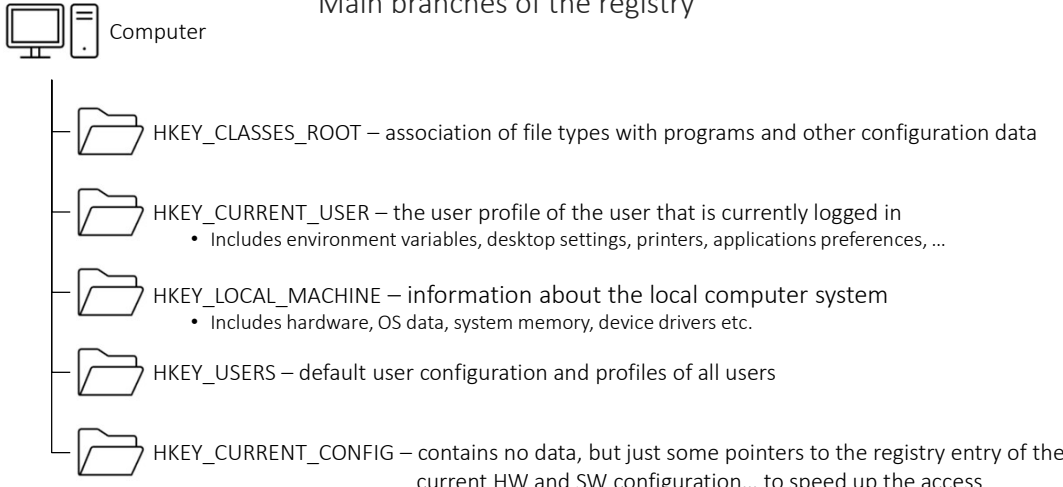


Nome	Tip	Dati
(Predefinito)	REG_SZ	(valore non impostato)
\System\CurrentControlSet\Control\Print\Printers	REG_DWORD	0x00000001 (1)
\System\CurrentControlSet\Control\Print\Printers\HP Deskjet 1510 series	REG_DWORD	0x00000001 (1)
\System\CurrentControlSet\Control\Print\Printers\HP Deskjet 1510 series\HP Deskjet 1510 Series Class Driver	REG_DWORD	0x00000001 (1)






The registry

30

Main branches of the registry



Computer

- 
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 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

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
- 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
 - *describe the FS configuration (e.g. cluster size, version, etc.)*
 - *all FS metadata in a regular file in the FS itself*

File system - NTFS

33

The Master File Table (MFT):

- 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

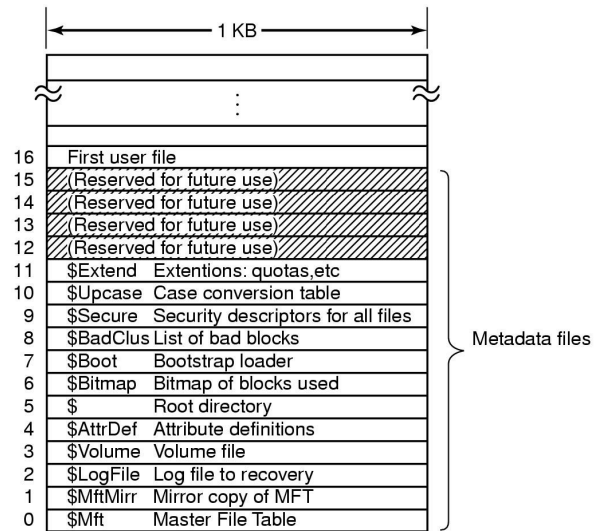
A file:

- 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*

File system - NTFS

34

MFT structure



35

- 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

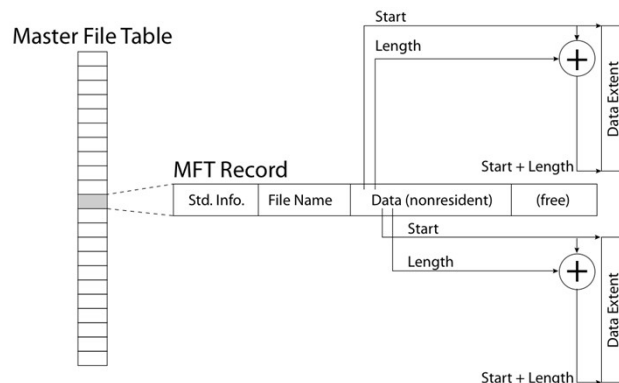
Master File Table



MFT record of a small-size file

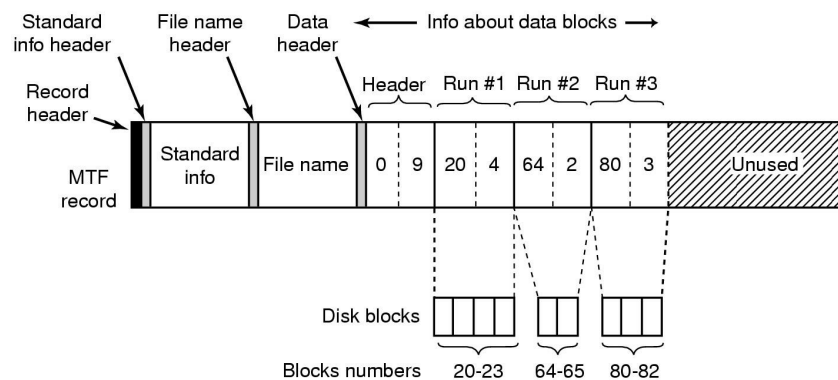
36

- 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



MFT record of a medium-size file

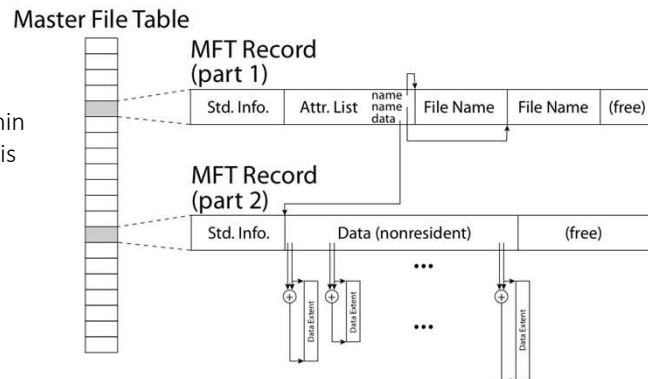
37



MFT record of a medium-size file

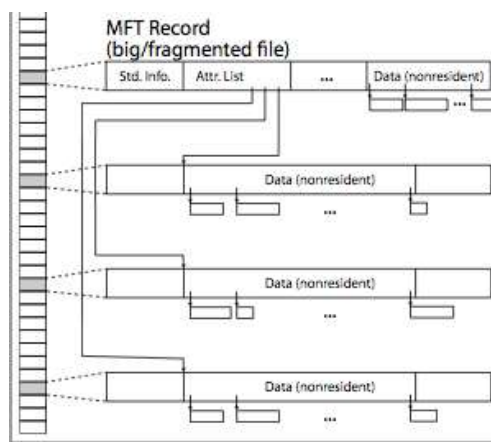
38

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

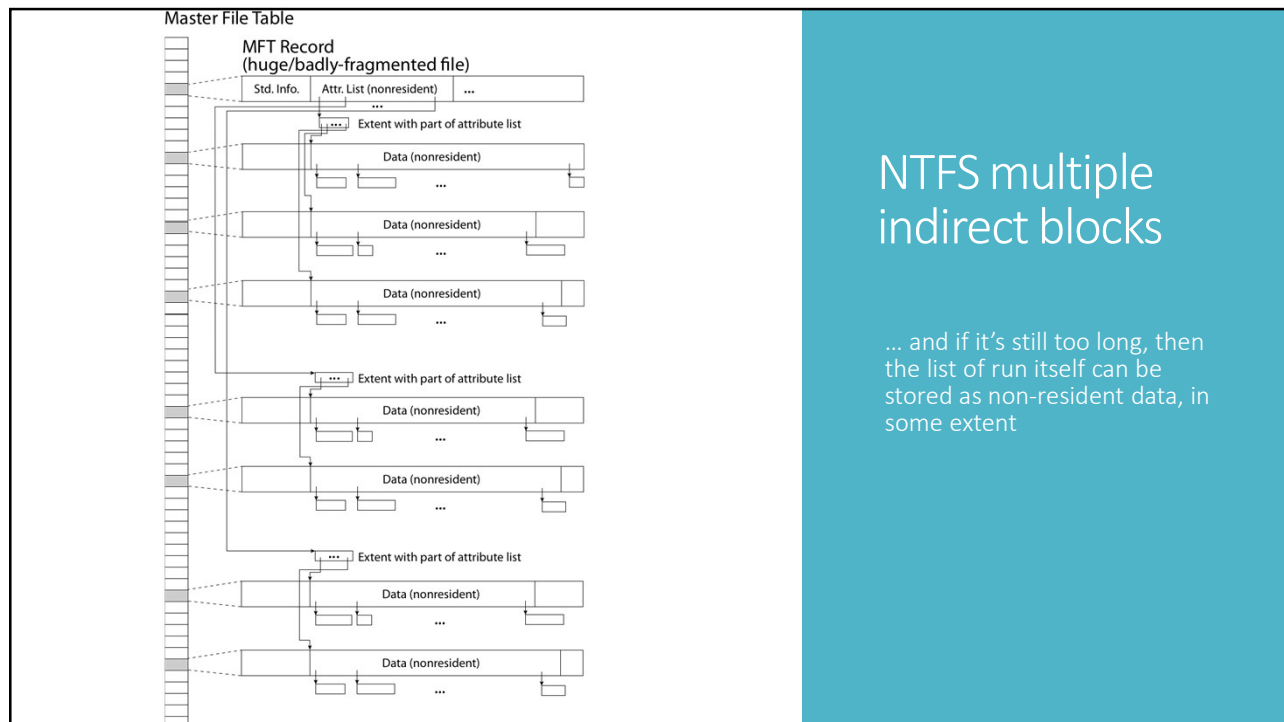
39



NTFS multiple indirect blocks

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

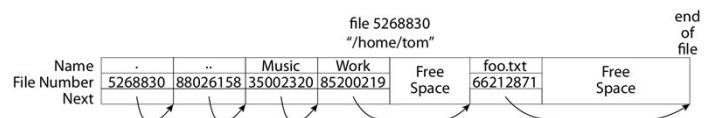
40



41

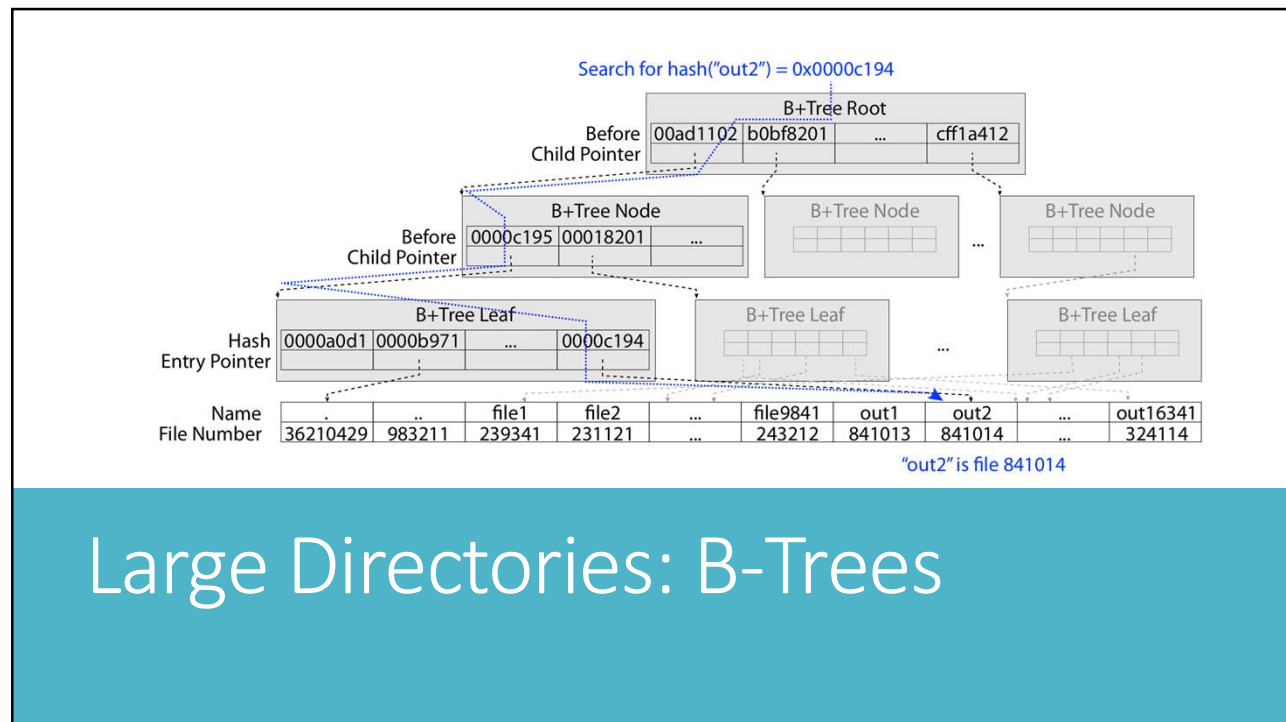
directories are files as well
map file name to file number
(#MFT record)

small directories organized as a
linked list



Directories

42



43

native support to **RAID** disks (level 0 and 1)

support for **FS compression**

journaling to improve the FS reliability:

- 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

NTFS – other features

44

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

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

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

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?

Why does in your opinion a key in the registry is a kernel kernel object?

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

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

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

Key elements of Windows security:

Security Reference Monitor (SRM)

Manages access control

Local Security Authority (LSA)

Manages security policies

Security Account Manager (SAM)

Database with users and groups information

Active Directory (AD)

User authentication in a domain

Plus:

Authentication packages

WinLogon and netLogon – handle logons at the keyboard and across the network, respectively

51

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

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

- 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

56

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
 - *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)*

57

```
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:

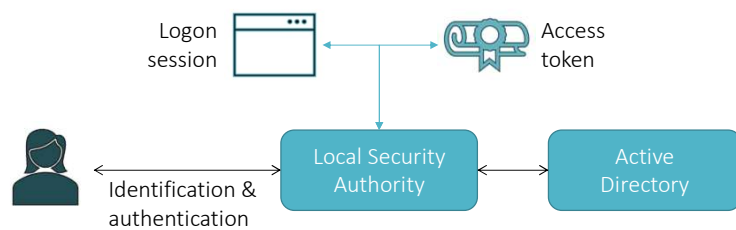
username & password
username & smartcard
biometrics

59

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

Login with SAM (workgroup)

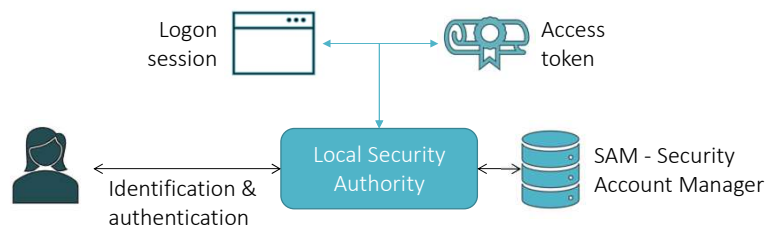
If the user has an account, it is associated to a Security ID (SID)

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

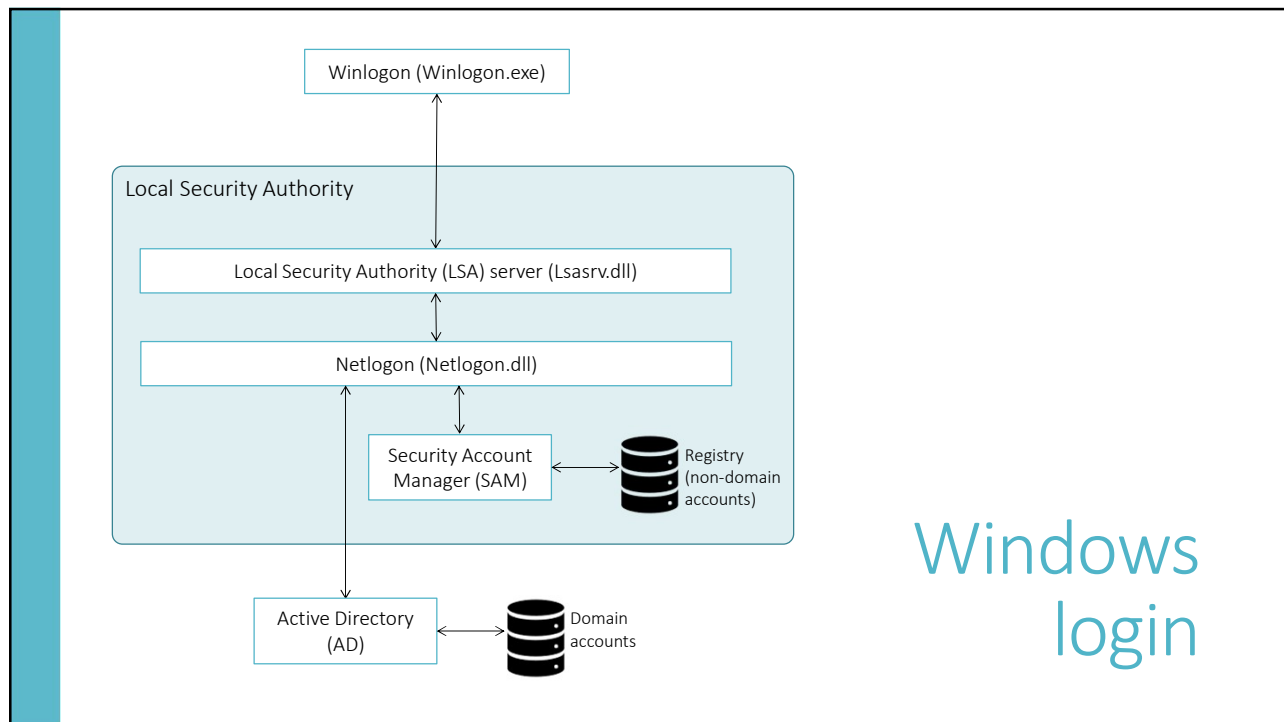
the token includes: SID, groups, privileges

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



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

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

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

Access Control Lists

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

Discretionary ACL (DACL)

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

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

69

Access Control Lists

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

each ACE includes a SID and an access mask:

- The SID specifies a user or a group
- The access mask could include ability to read, write, create, delete, modify, etc.
- access masks are object-type specific
 - *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
```

- 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

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

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

ACEs {

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

access mask, will be discussed later with the File System

Audit : // audit data for the SD from the system access control list (SACL).

Sddl : // it's the security descriptor in the SDDL syntax

Powershell Example

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

More on Security Descriptors & Access Control

- In powershell it is possible to set the DACL and the SACL by means of set-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

74

More on Security Descriptors & Access Control

Windows also supports “conditional ACEs”

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

For example, a conditional ACE may encapsulate the rule:

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

... 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

Object O

SD - DACL

ACE	Permissions
ACE1	Denied: Stefano read, write, execute
ACE2	Allow: Group A write
ACE3	Allow: Everyone read, execute

Thread A Access token: Stefano, Group A, Group B, Group C

Thread B Access token: Luisa, Group A

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

Object O

SD - DACL

ACE	Permissions
ACE1	Denied: Stefano read, write, execute
ACE2	Allow: Group A write
ACE3	Allow: Everyone read, execute

Thread A Access token: Stefano, Group A, Group B, Group C

Thread B Access token: Luisa, Group A

Access denied

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 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

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.

78

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

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 of a given integrity level can only change state of objects of equal or lower integrity levels

80

Mandatory Access Control

When a user launches an executable file:

- the new process is created with the **minimum** of the user integrity level and the file integrity level
- 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 uses integrity levels to run potentially hostile code from the Internet
- its process runs at low integrity level
- while the rest of the OS is marked medium or higher integrity

84

```
PS C:\Users\stefa> whoami /USER
USER INFORMATION
```

```
User Name SID
```

```
=====  
tabatayoga\stefa S-1-5-21-1144226474-1424528306-2619936039-1001
```

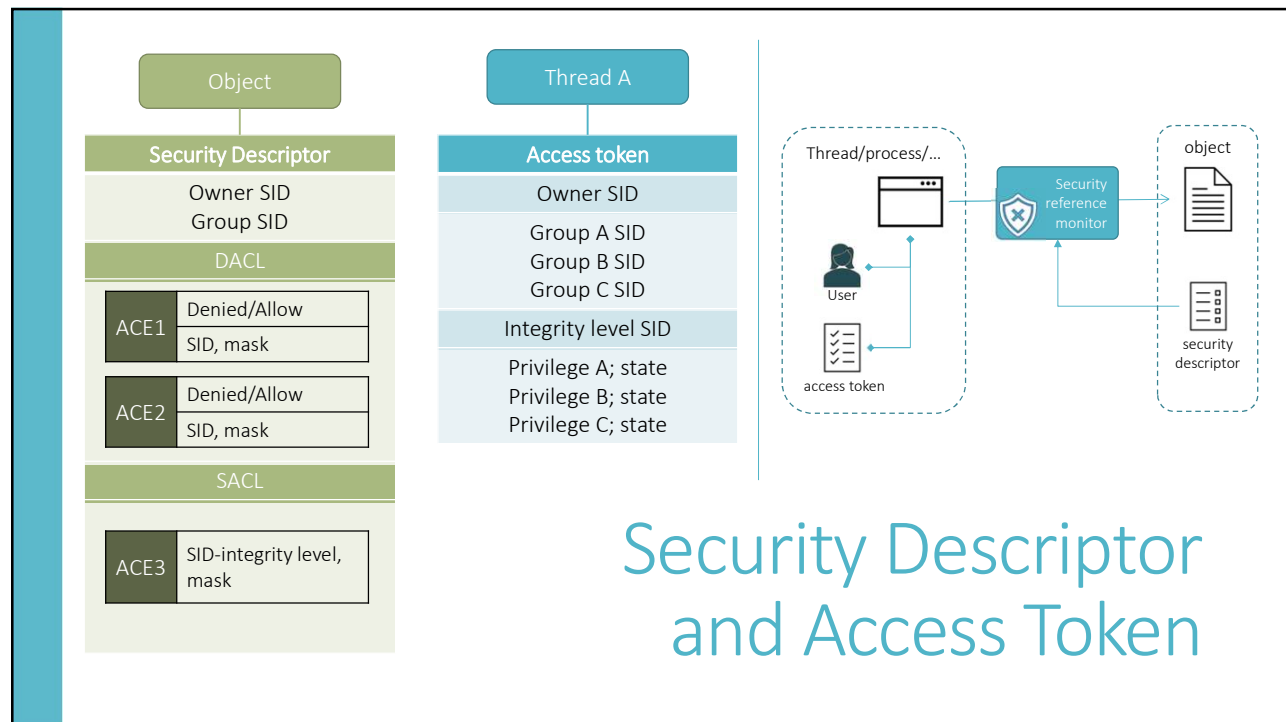
```
PS C:\Users\stefa> whoami /GROUPS
GROUP INFORMATION
```

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

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?

Does Windows use a protection matrix?

87

File system security



ACL in the FS

Privileges

auditing

88

File system security

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

- operations initiated by a driver bypass most security checks

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

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

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

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

Generic rights

Standard rights

Generic read	Generic write	Generic execute	Generic all	unused	Unused	Maximum Allowed Access System Security	Unused	Unused	Unused	Synchronize	Write owner	Write DAC	Read control	Delete	Specific rights												
--------------	---------------	-----------------	-------------	--------	--------	--	--------	--------	--------	-------------	-------------	-----------	--------------	--------	-----------------	--	--	--	--	--	--	--	--	--	--	--	--

93

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

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

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.*

100

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
- 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

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

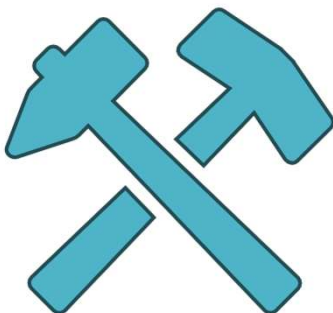
Windows Vulnerabilities...



Windows, like all OS's, has security bugs
and bugs have been exploited by attackers to
compromise customer operating systems

104

... and Hardening



Microsoft now uses process improvement called the
Security Development Lifecycle

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

105

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!

Windows Security Development Lifecycle

106

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

Windows Security Defenses

107

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

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

Windows System Hardening

108

some user accounts can have privileged SIDs

- E.g administrators

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

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

Account Defenses

109

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

Low Privilege Service Accounts

110

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

Low Privilege Service Accounts

111

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

Stripping Privileges

112

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

Network Defenses

113

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

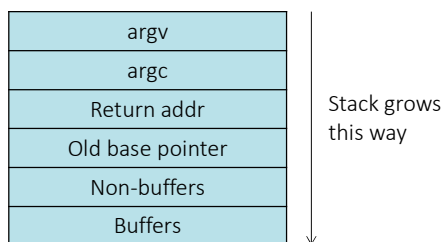
Memory Corruption Defenses

114

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

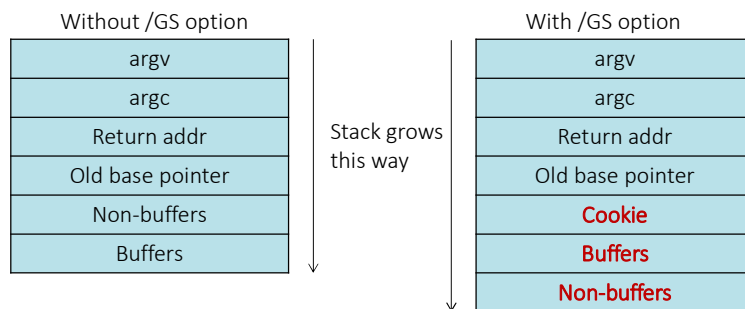


Stack-based Overrun Detection

115

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)



Stack-based Overrun Detection

116

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)

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)

Other Memory Corruption Defenses

117

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

- 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**

Other Defenses

118

web browser is a key point of attack

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

Browser Defenses

119

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

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

Encrypting file system

120

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

Data protection API (DPAPI)

121

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

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 answer

125

Summary



Windows architecture



Windows security architecture



vulnerabilities



security defenses

account, network,
buffer, browser



crypto services

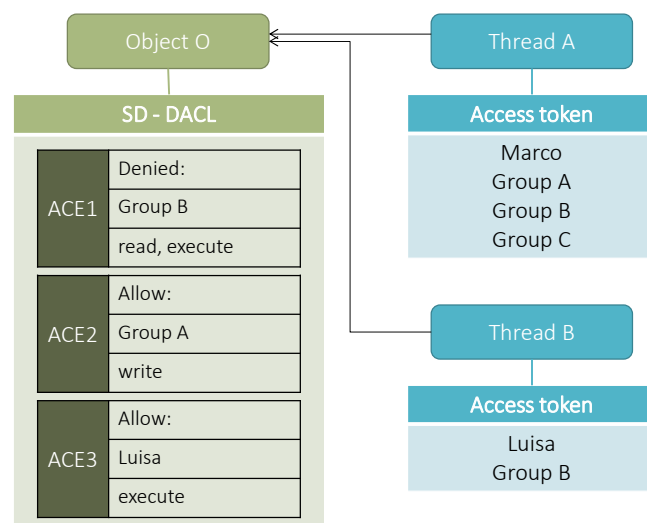
126

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



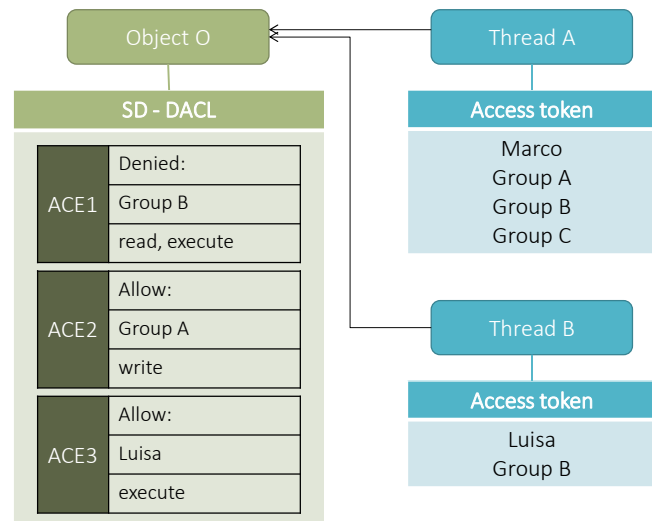
Solution

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



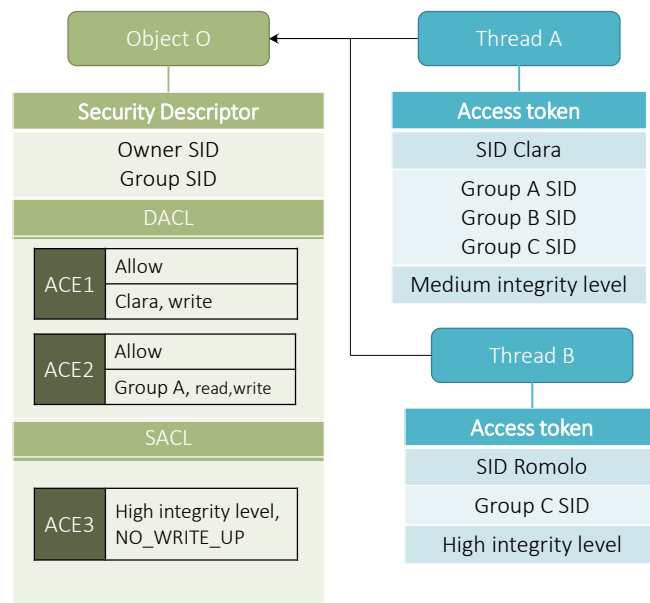
129

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.

