Secure Operating Systems

The fundamentals of OS design: Windows OS Analysis

Anonymous Student

School of Technology, Design, & Environment

Oxford Brookes University

Table of Contents

Abstract

A look at the security features provided through the Windows operating system file system and comparison to MacOS.

Introduction

Computers without software are of challenging use to humans; they are simply hardware: solid parts and data stored in a solid, electronic, or magnetic state. The software in computers allows data to be read in certain formats and processes to be performed. This software is split into two categories: **system programs** that manage the operation of the computer controlled by the **kernel**, and **application programs** which perform tasks for the human user. These will both use **system calls** to request service from the operating system's kernel. The kernel being the master software managing all other software within the OS.

The operating system (OS) is a middle-man between the hardware and the user software, allowing communication through one to the other in an abstracted way. The user need not know about all the minute details of the hardware.

Windows provides the security features including: filesystem, access control, encryption, process management, multiprogramming, and data management.

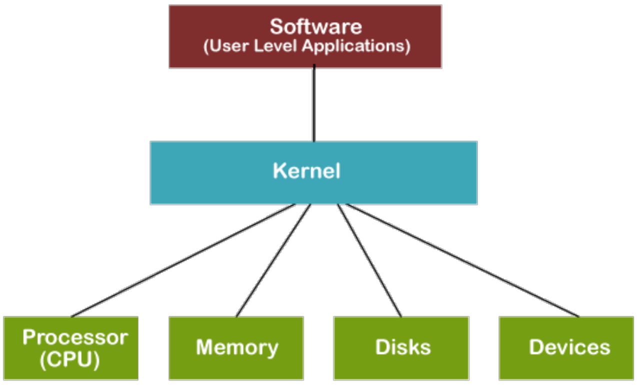


Figure 1: Kernel diagram [1]

How OS supports and provides security features, reliability, and protection

Filesystem

There are different types of file systems that can be implemented, including: **NTFS**, **FAT**, **CDFS**, **UDFS**, **TDBSS**, and **SMB**. Other operating systems may have different types of file systems also. The OS preceding Windows, MS-DOS, used FAT (file allocation table). NTFS later superceded FAT on Windows. As of all file systems: the purpose is to store, organise, and find files on hard disks efficiently.

NTFS provides a number of security features including security descriptors, encryption, disk quotas, and rich metadata.

Resilient File System (**ReFS**) was introduced in Windows Server 2012 as a proprietary feature with the intention of replacing NTFS. It is more **resilient** as it can detect and fix **corruption**. Files can become corrupted during the saving process as if the computer shuts down during this process the file addresses will not have been properly written to the disk.

Access control

Access control manages which users can access resources in the operating system. This includes **access tokens** that contain information about a user, and **security descriptors** which contain security information about a securable object.

Pipes can be used to send data one-way between files or processes. The pipe is a pseudofile that will connect file A to file B, acting as an input file for A and an output file for B. On windows named pipes are stored in a hidden partition called the **NPFS**.

Security descriptors

Windows includes **securable objects**. These include all named windows objects and some unamed ones including processes and threads. These objects can have security descriptors, these contain all the security information associated with the object **[1]**.

Files are protected with **9-bit binary protection code**. This has three 3-bit fields, one detailing the owner, one for other member's of the owner's group, and one for everyone else. Each bit has a bit for read access, write access, and execute access, these are known as **rwx bits [2]**. An example might be 'rwx--xr--' meaning current user can read, write, execute, user group can execute only, and everyone else can only read. When a file is opened a **file descriptor** integer is returned or an **error code** if access is prohibited.

Disk mounting will attach the disk's file system to a directory tree in the OS' filesystem.

Privileges and audits

Users can have **priviledges**, these allow or restrict access to certain parts of the operating system. Such as a guest may not have read/write permissions but an administrator is able to change resources across all users. Custom user groups can be made to predefine groups of priviledges. **Audits** are created when users attempt to access resources, these serve as a log of all who have attempted to access a resource and can flag any suspicious behaviour.

Encryption

Encryption for individual files is provided using a public-key system through Encrypting File System (**EFS**). Which was first introduced in version 3.0 of NTFS.

EFS is used to protect files storing data from being accessed by unauthorised users. Folders or files can be marked for encryption and once the author closes a file it is then encrypted.

EFS uses public key encryption (transparent encryption). When a user requests to encrypt a folder EFS first uses Advanced Encryption Standard (**AES**) cipher to create a key which is then encrypted through an **RSA algorithm** to create an **X.509 certificate**. The certificate stores a generated **private/public key pair**. The public key of which is used to encrypt and the private to decrypt. The private key is stored in the user profile and may only be used by the user whom owns the certificate. Although the public key which encrypts the folder can be seen by anyone.

I/O spooling and buffering

In early computers and OS there was the problem that whilst waiting for input/output the CPU would remain idle.

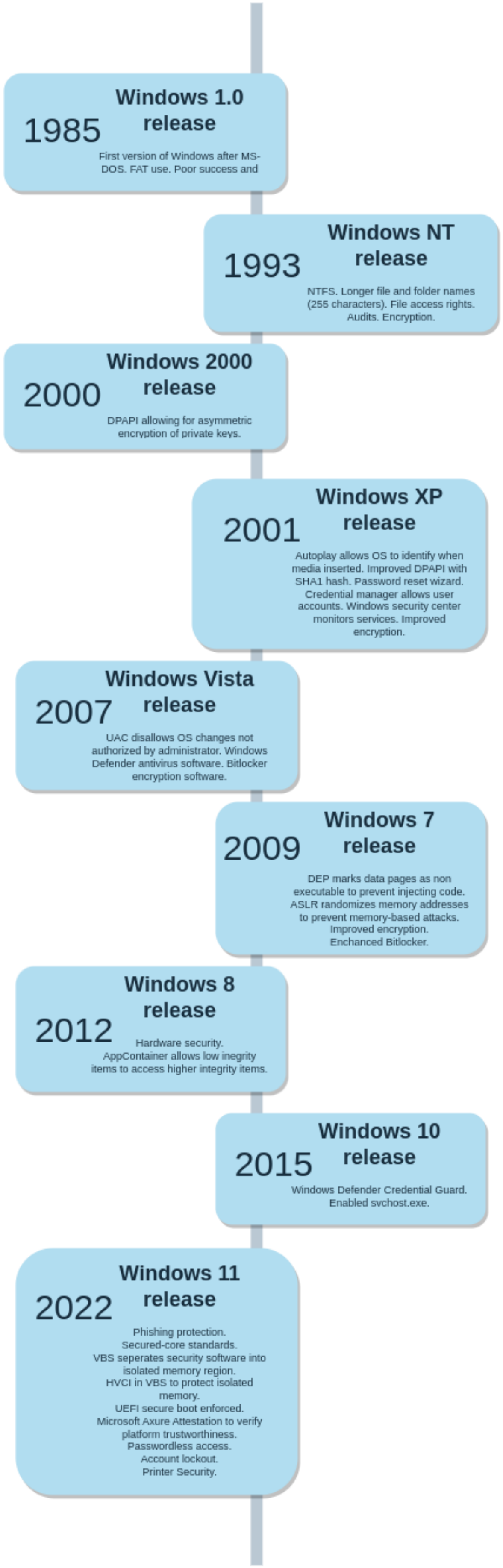
Spooling is a buffer management technique that allows input/output data to be temporarily stored in secondary memory until a process requests the data. This prevents the CPU idle time by allowing other processes to run whilst waiting.

Windows uses **virtual memory** to help memory management. This allows secondary (disk) memory to be used as additional main memory (RAM) for executable processes.

Buffering is used because the speed of a sender’s (machine or device) transmission may be slower than the reciever’s. In this case the reciever creates a buffer memory space in the main memory and accumulates the bytes sent. This may also happen in reverse.

Comparing Windows to MacOS

MacOS is known for being a secure OS. It uses **touchID** [3] to replace passwords and encrypt files. Windows has only introduced a similar feature in Windows 11 called **Hello Windows** which explores



1985 Windows 1.0 release
First version of Windows after MS-DOS. FAT use. Poor success and

1993 Windows NT release
NTFS. Longer file and folder names (255 characters). File access rights. Audits. Encryption.

2000 Windows 2000 release
DPAPI allowing for asymmetric encryption of private keys.

2001 Windows XP release
Autoplay allows OS to identify when media inserted. Improved DPAPI with SHA1 hash. Password reset wizard. Credential manager allows user accounts. Windows security center monitors services. Improved encryption.

2007 Windows Vista release
UAC disallows OS changes not authorized by administrator. Windows Defender antivirus software. Bitlocker encryption software.

2009 Windows 7 release
DEP marks data pages as non executable to prevent injecting code. ASLR randomizes memory addresses to prevent memory-based attacks. Improved encryption. Enchanced Bitlocker.

2012 Windows 8 release
Hardware security. AppContainer allows low inegrity items to access higher integrity items.

2015 Windows 10 release
Windows Defender Credential Guard. Enabled svchost.exe.

2022 Windows 11 release
Phishing protection. Secured-core standards. VBS seperates security software into isolated memory region. HVCI in VBS to protect isolated memory. UEFI secure boot enforced. Microsoft Axure Attestation to verify platform trustworthiness. Passwordless access. Account lockout. Printer Security.

Figure 2: Timeline of security features based on [4] [5]

passwordless technology such as fingerprint and face recognition. Although this can lock users out at times. Account lockout after failed attempts is also a security feature against remote desktop protocol (RDP) attacks. Windows 11 has specific hardware specifications; requiring at least 2 cores which may limit its availability to some users.

The hardware is also specialized in macOS and requires unique chips, and I/O ports and cables that are only produced by Apple. The chips include security features such as a Boot ROM for secure booting and AES engine space to manage encryption. MacOS also provides XD (execute disable), ASLR (address space layout randomization), and SIP (system integrity protection).

For the file system **APFS** is designed for SSD storage devices whereas NTFS was designed more for HDD. One major disadvantage of APFS over **NTFS** is that it does not have data compression. Windows 11 has also started to use the **ReFS** filesystem which offers more security.

For access control this is defined through the filesystem. Windows uses **securable objects** and Mac uses **keychain items** which contain access items which contain multiple ACLs. Mac’s keychain items are things such as passwords which contain an ACL of the apps they can access. Whereas Windows items are apps and contain an ACL that is either a **DACL** or a **SACL** (allows administrators to log access attempts) listing all of the ACEs of trustees (accounts) and their permissions of access.

For encryption macOS seems to be superior. It uses **FileVault** to encrypt the data on the startup disk. This will include a recovery key if the password is forgotten. Windows does not have such a feature unless it is used from a third party.

For spooling and buffering both OS’ are the same.

Overall it seems that macOS is more secure than Windows, notably because the security is built into the hardware through features such as the custom chips, and thus all aspects are controlled and secured.

Effects on application developers

Application developers can manipulate the functions and security of the operating system using the **programming interface** provided by Microsoft and **invoking system calls** to make the kernel perform lower-level functions. For instance, applications can call access control functions that manage access control for resources in the operating system.

Threads are processes that are split into multiple jobs. These jobs run concurrently and have their own stack, program counter, and register but share the same memory. This means threads can share data between themselves without pipes or signals. There are two types of threads: **user level**, and **kernel level**. User level threads allow developers to create threads for their programs.

Device drivers are software components that communicate between a device and the operating system to send data to user software. Software drivers can only simply communicate with the operating system to get core operating system data instead of data from a device. Developers can write custom drivers to get access to operating system data or to get data from devices.

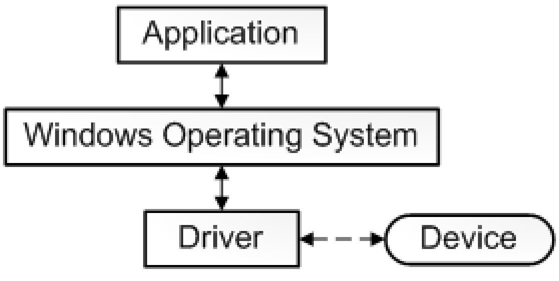


Figure 3: Driver diagram [4].

Conclusion

Recommendations for hardening the system and OS file systems

To harden the system the recommendations would be as follows. To enable **secure boot** to protect against malicious code being executed before the boot process. To use improved **antivirus monitoring software**. To encrypt the host drive using a **hardware TPM**. Use **biometrics** or **FIDO authentication** in addition to **secure passwords**. And to **limit the peripherals that can connect**.

Personal reflection

I think that I have completed this work relatively well. I have learnt a lot. I found it very helpful to refer to the recommended text books and documentation. Although it has been challenging to focus on the work because of my poor mental health. To improve I would have made my explanations more succinct, asked others for feedback, and dedicated more space to a conclusion.

Bibliography

1: Javatpoint, What is Kernel?, , https://www.javatpoint.com/what-is-kernel
2: Microsoft, Learn Microsoft Windows Security, 2023, https://learn.microsoft.com/en-us/windows/security
3: Andrew S. Tanenbaum, Albert S. Woodhull, Operating Systems Design and Implementation, 2006,
4: Greg Belding, Windows OS brief history, 2019, https://resources.infosecinstitute.com/topic/windows-os-security-brief-history/
5: Kolide, Windows 11 Security: What You Need to Know (22H2 Update) , , https://www.kolide.com/blog/windows-11-security-what-you-need-to-know-22h2-update
6: Apple, MacOS Documentation, , https://www.apple.com/macos/security/
7: Microsoft, What is a driver?, , https://learn.microsoft.com/en-us/windows-hardware/drivers/gettingstarted/what-is-a-driver-