Source: Bing Chat AI

GETTING STARTED WITH WINDOWS PERMISSIONS

DEPARTMENT OF COMPUTER SCIENCE. UNIVERSITY OF OVIEDO

Computer Security | 2023 – 2024 (V4.0 "*S-81 Isaac Peral*")

Content

Introduction

In this document we will try to explain how to protect the files (our virtual machines) that we may have to store on the disk of one of the computers in the laboratory, which are shared by many students.

The best way to work is for each student to have their own USB 3.0 SSD. It is the best option because you can create virtual machines there, work directly with them, both in class and at home. If this is your case, you can skip this guide.

But if you cannot afford (or do not want to) buy one of these disks, you can always use the D: disk (or the largest disk your machine has) from the lab computer to store your virtual machines. In this case, you will only be able to work in practice on that particular machine, and you must make sure that the files you leave on that machine will remain unchanged until the next time you go to the lab to work.

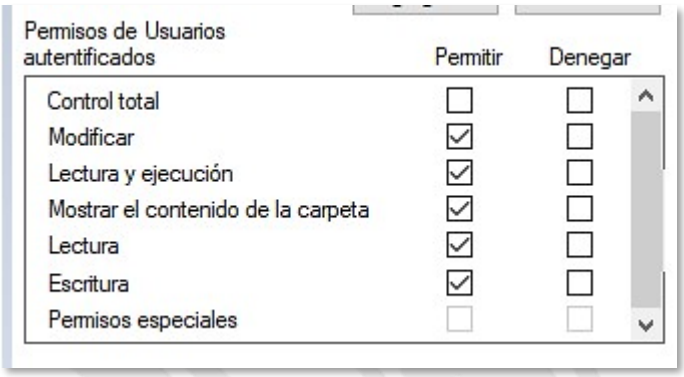
After a brief introduction on how the NTFS system works, a step-by-step guide is included to do so. Finally, it should be noted that this guide is basically **for people who work on laboratory machines** or at home but have doubts about the permissions of their *Windows*. If you are already familiar with them or work at home, it will not add much.

NTFS System Overview

The default file system used by *Windows* is called NTFS. This file system uses Access Control Lists to manage the permissions that a user (or group) has for each file on the disk.

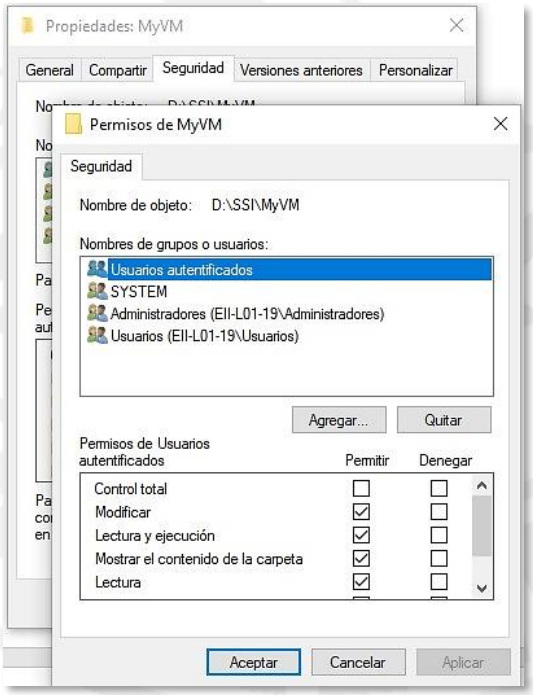
NTFS is based on the use of 13 different types of access, but they are typically grouped into 7 basic access types:

- Full control.
- Modify.
- Reading and execution.
- Display the contents of the folder.
- Read.
- To write.
- Special permits.



The meaning of each permission is quite clear from its name.

For each user/group, permissions can be given (*Allow*) or not (*Deny*):



In this example we can see that users who belong to "Authenticated Users" can modify, read, and execute, show the contents of the folder, and read the file in question. To view the permissions of another user or group in the list, simply select them from the list.

There are several rules that the operating system applies when a process wants to access a file to verify whether access is feasible or not. We can simplify them as follows:

- Any user/group that is not on the list cannot access the file.

- If more than one rule can be applied (for example, a user who is both in the "administrator" and "user" group), the Deny rules take precedence over the allow rules.
- File permissions take precedence over directory permissions.
- Assigned permissions take precedence over unassigned permissions.
- Explicit permissions (those indicated to the object itself) prevail over inherited permissions.

Permission Inheritance

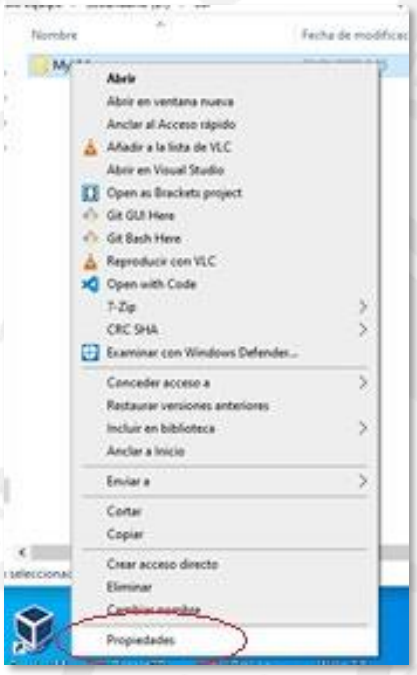
To simplify the assignment of permissions to files and directories, *Windows* includes a permissions inheritance mechanism. Thus, the permissions of a file or directory are, by default, inherited from those of the folder to which it belongs. And they are not just inherited, they are tied to them, so if you change the permissions of a folder, you also change the permissions of the objects in it. You cannot change an object's permissions if inheritance is enabled for it.

Therefore, if we want to change the permissions of a file or folder, the first thing we must do is disable inheritance for it.

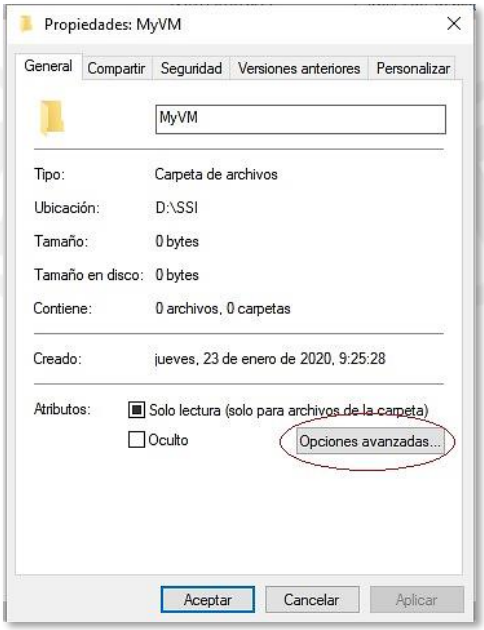
Securing our virtual machines

The directory structure that we are going to use to store our virtual machines on the local disk will be as follows:

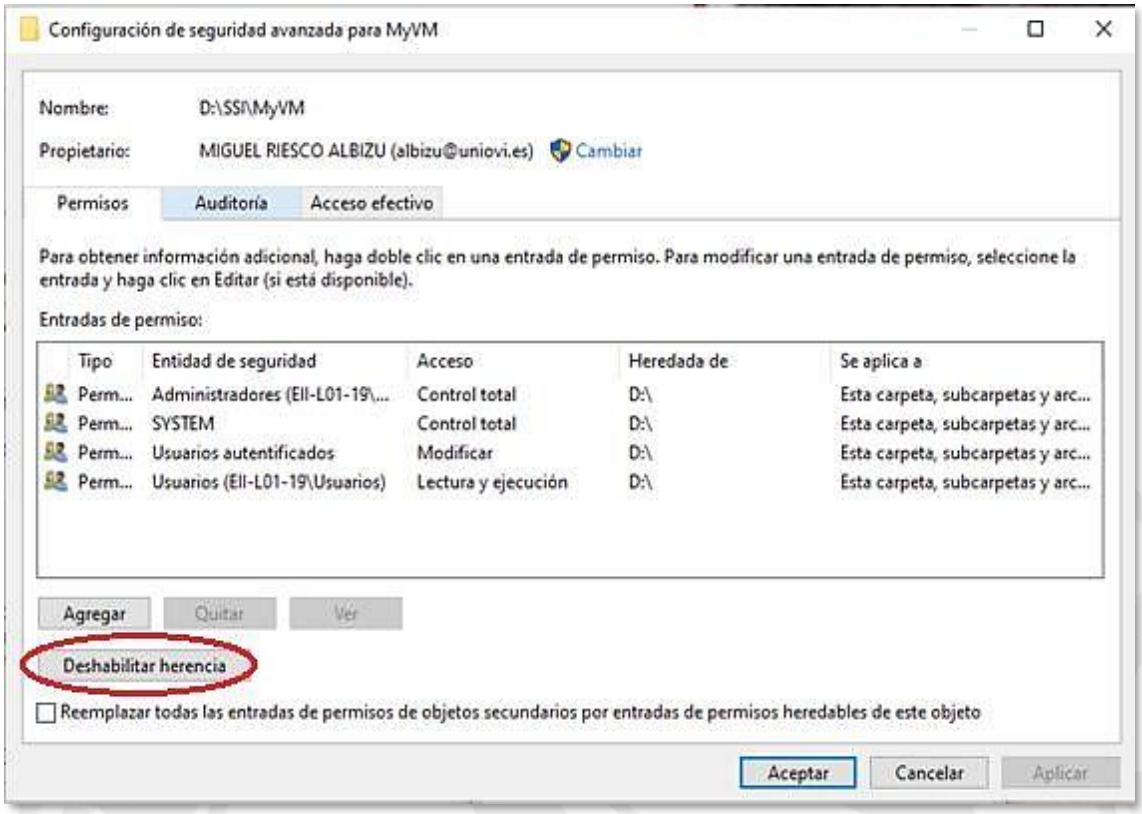
- On disk D: There must be a directory named SSI. If it is not, create it.
- Create a directory within SSI with your UO as the name. This will be where you will store your VMs, so it is the directory we are going to protect.
- To protect this directory (the **UOxxxx** directory) we are going to change its properties. After right-clicking on the directory, choose "Properties" from the drop-down menu:



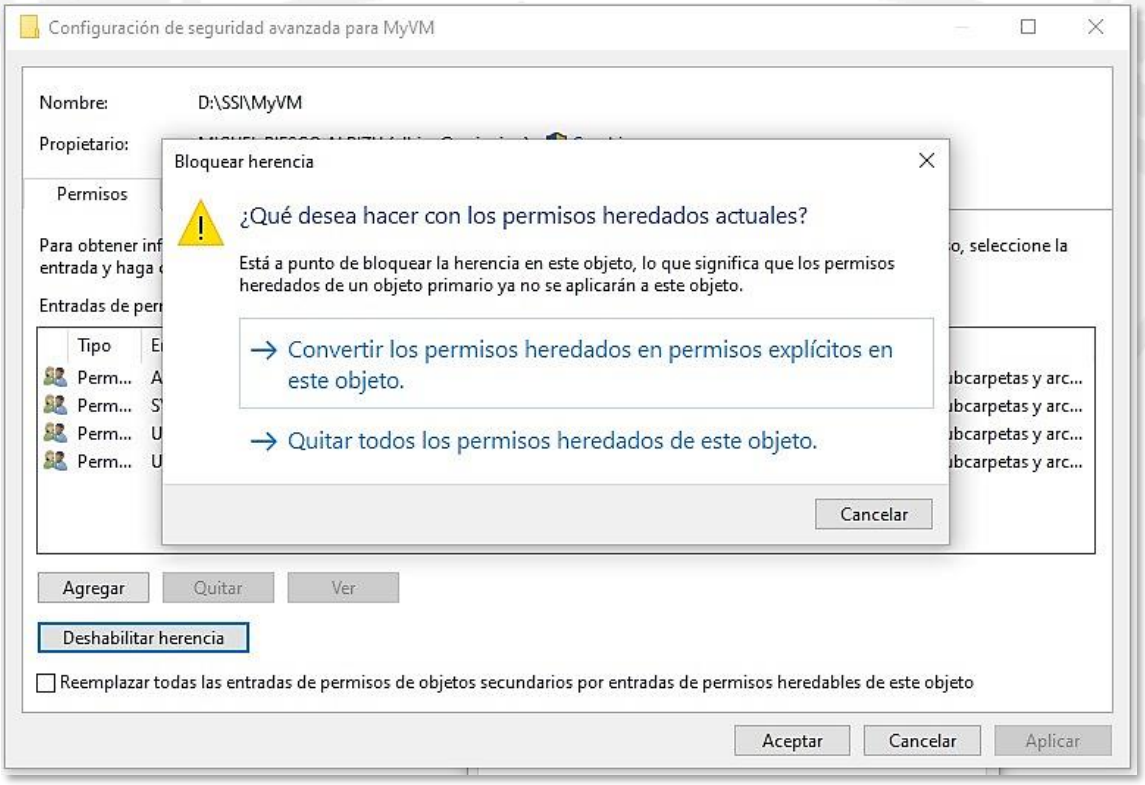
- The properties box appears. Since what we need to do first of all is to disable inheritance, we need to access "Advanced Options":



- There are several options in this window. We are interested in ***"Disable Inheritance***":



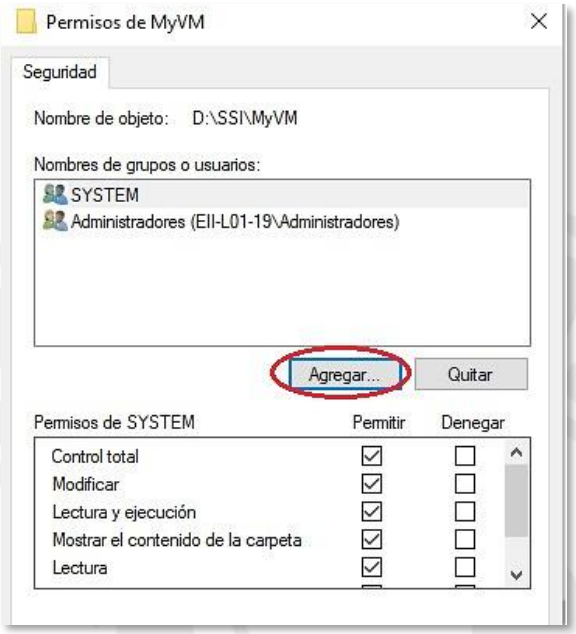
- There are now two options: ***"Make inherited permissions explicit permissions on this object***", which will leave the same permissions you have but making them explicit, or "***Remove all inherited permissions***". Both are possible, but it is almost better to choose the first option.



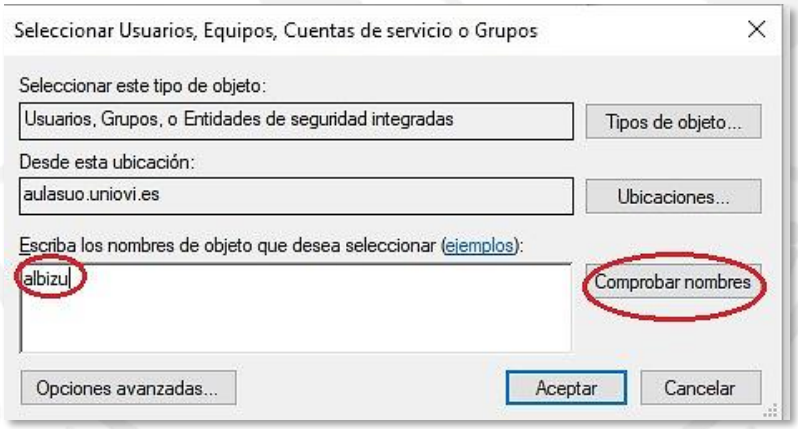
- After accepting in the previous window, we can now prevent other users from accessing our directory. To do this, we will remove the "***Authenticated Users" and "***Users*"* groups from the list of users and groups

that have some permission on the file. To do this, we just have to select one of those groups and press "*Remove*"

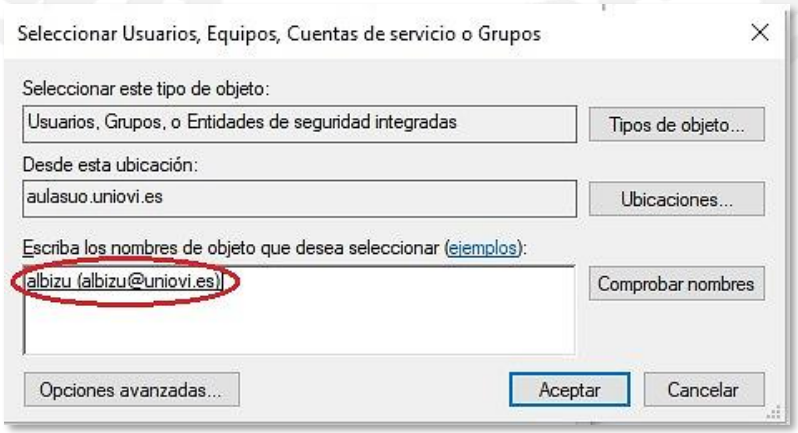
- The last step will be to ensure that our own user has access to the folder. To do this, by pressing Add, we will add a new element to the ACL. First we add the user:



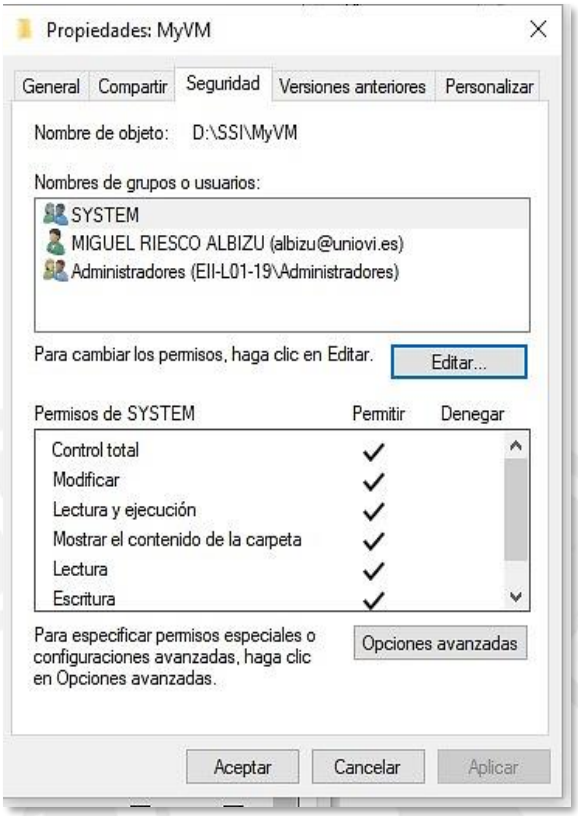
- This will take us to the following window, where we will enter our user identifier (OU):



- After pressing "Check Names", if the user exists, the system will fill in the name:



- After pressing "*OK*", we will assign the "***Full Access***" permission after pressing *"Edit*" in the next window. The end result should look something like this:



And that is it. This should be enough to allow us (and only us) to work with that folder. *System* and *Administrators* users can also log in, but they can do so in any case.