ITAS141 Lab 4

By Ethan Holmes

Table of Contents

Objective:

The objective for this lab is to work with configuring storage solutions as well as File Systems across our Virtual Machines in the class. To show this, we have various steps throughout Chapter 4 of our textbook to show what we need to accomplish. In this lab we will be documenting the final RAID 5 configuration in activity4-3, Our mounted virtual drive in activity 4-4, our Powershell command text in activity 4-5, A screenshot of our open files screen in activity 4-9 and ending with a step by step guide to completing step 4-12.

Activity 4-3: Screenshot of RAID configuration

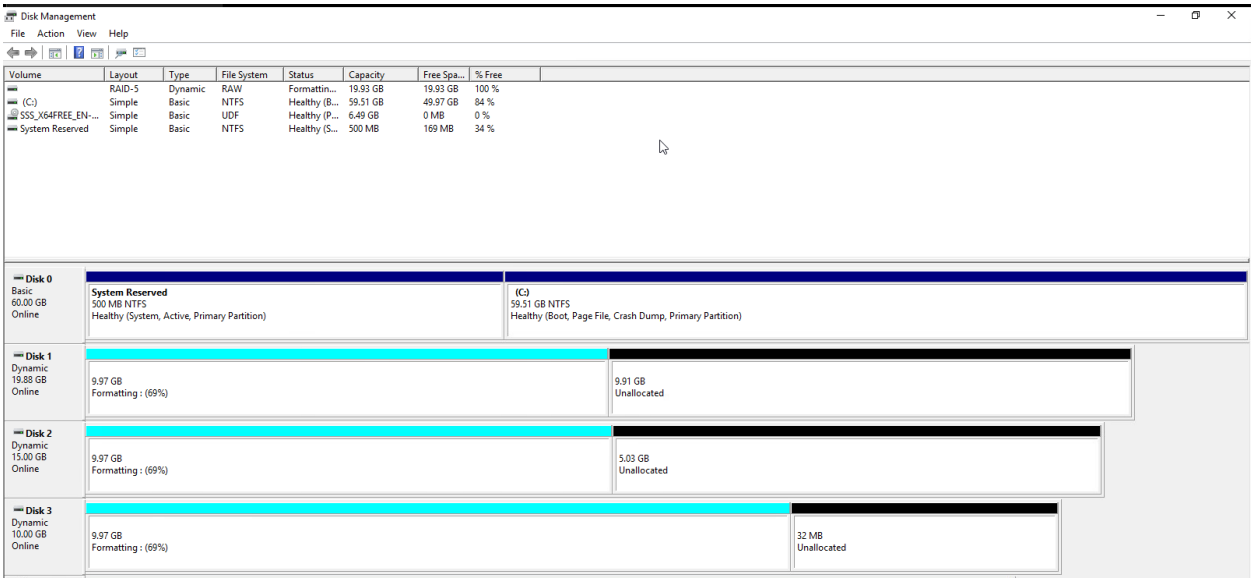

Figure 1: A completed RAID 5 format.

Here is the completion of my RAID volume setup, Disk 1,2 and 3 have all been setup in a RAID 5 setup.

Activity 4-4: Working with Virtual Disks in Disk Management

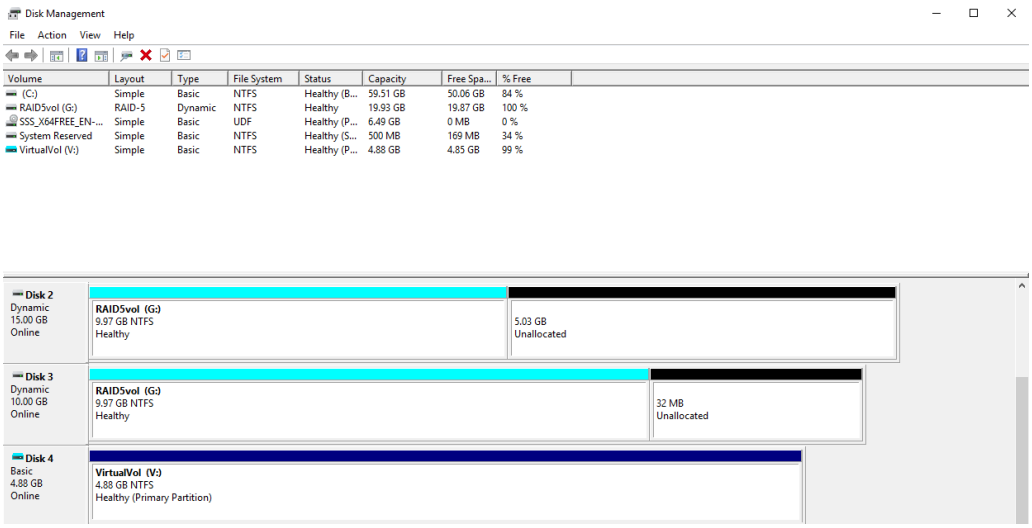

Figure 2: A created VDH being mounted

In the figure above, we can see the creation of the virtual disk, and the virtual disk being mounted.

Activity 4-5: Working with Virtual Disks in PowerShell

In activity 4-5, we use PowerShell to create and mount virtual disks

```
PS C:\Users\Administrator> cd \
PS C:\> New-VHD Virtual1.vhdx -SizeBytes 5GB


ComputerName            : WIN-VC8KMVFSA7H
Path                    : C:\Virtual1.vhdx
VhdFormat               : VHDX
VhdType                 : Dynamic
FileSize                : 4194304
Size                    : 5368709120
MinimumSize             :
LogicalSectorSize       : 512
PhysicalSectorSize      : 4096
BlockSize               : 33554432
ParentPath              :
DiskIdentifier          : 6CD61394-B1DE-4F2A-A2D1-1F4B22434374
FragmentationPercentage : 0
Alignment               : 1
Attached                : False
DiskNumber              :
Number                  :
```

Figure 3: New-VHD command to create VHDX

```
PS C:\> New-VHD Virtual2.vhd -SizeBytes 5GB


ComputerName            : WIN-VC8KMVFSA7H
Path                    : C:\Virtual2.vhd
VhdFormat               : VHD
VhdType                 : Dynamic
FileSize                : 16384
Size                    : 5368709120
MinimumSize             :
LogicalSectorSize       : 512
PhysicalSectorSize      : 512
BlockSize               : 2097152
ParentPath              :
DiskIdentifier          : EF930937-449E-4FBD-8226-145A647B3317
FragmentationPercentage : 0
Alignment               : 1
Attached                : False
DiskNumber              :
Number                  :
```

Figure 4: New-VHD to create a VHD

```
PS C:\> Mount-VHD Virtual1.vhdx
PS C:\> Get-Disk

Number Friendly Name Serial Number                 HealthStatus         OperationalStatus      Total Size Partition
                                                                                                          Style
------ ------------- -------------                 ------------         -----------------      ---------- ----------
1      Msft Virtu...                               Healthy              Online                      5 GB RAW
0      VMware Vir... 6000c290cc99cdb169de0b755860cb19 Healthy           Online                    100 GB MBR
```

Figure 5: Mount-VHD and Get-Disk to show the disk

```
PS C:\> Set-Disk -Number 1 -IsOffline $false
PS C:\> Initialize-Disk -Number 1
PS C:\> Get-Disk

Number Friendly Name Serial Number                    HealthStatus         OperationalStatus      Total Size Partition
                                                                                                  Style
------ ------------- -------------                    ------------         -----------------      ---------- ----------
1      Msft Virtu...                                   Healthy              Online                      5 GB GPT
0      VMware Vir... 6000c290cc99cdb169de0b755860cb19 Healthy              Online                    100 GB MBR
```

Figure 6: Initializing the Disk, giving it a partition

```
PS C:\> New-Partition -DiskNumber 1 -Size 4.9GB -DriveLetter V

   DiskPath: \\?\scsi#disk&ven_msft&prod_virtual_disk#2&1f4adffe&0&000001#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
2                V           34603008                                 4.9 GB Basic

PS C:\> Format-Volume V -FileSystem NTFS -NewFileSystemLabel VirtualVol

DriveLetter FileSystemLabel FileSystem DriveType HealthStatus OperationalStatus SizeRemaining   Size
----------- --------------- ---------- --------- ------------ ----------------- -------------   ----
V           VirtualVol      NTFS       Fixed     Healthy      OK                      4.87 GB 4.9 GB

PS C:\>
```

Figure 7: Formatting the Disk, giving it a size and drive letter

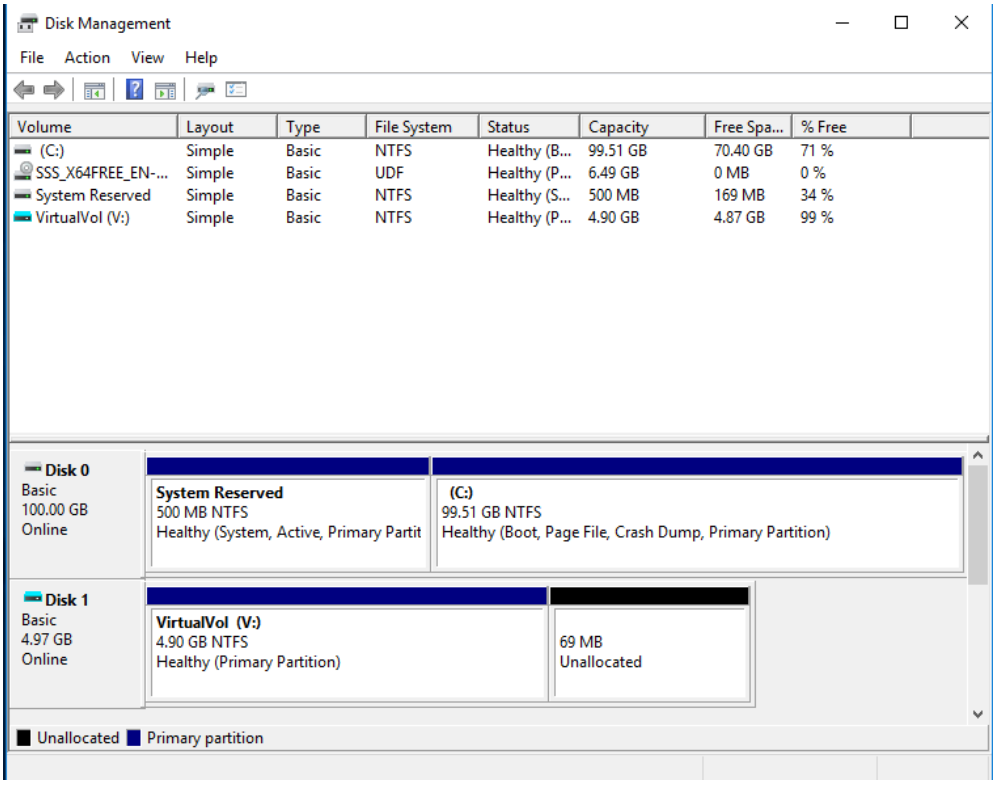

Figure 8: Showing the PowerShell commands worked in Disk Management

```
PS C:\> Dismount-VHD Virtual1.vhdx
PS C:\> del virtual*
PS C:\>
```

Figure 9: Demounting the Disk through PowerShell

Activity 4-9: Creating a Hidden Share and Monitoring Access

In this activity, we had to create a hidden folder and locate it though Network, we also had to see if we could find client connections through Computer Management.

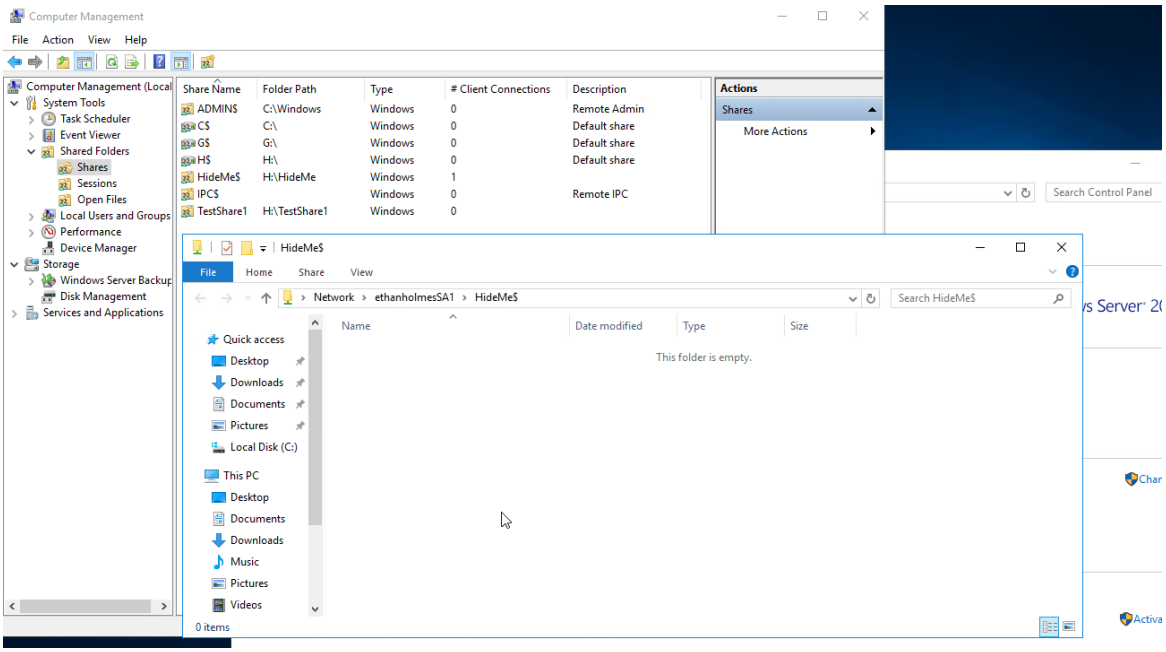

Figure 10: Monitoring and showing the Hidden folder

Activity 4-12: Experimenting with File and Folder Permissions

In this Activity, we will experiment with a variety of File and folder permissions by changing what we can do with them, to start this lab, navigating to our “TestPerm” folder that we created in an earlier activity, we will first turn on file name extensions from the View tab in the toolbox.

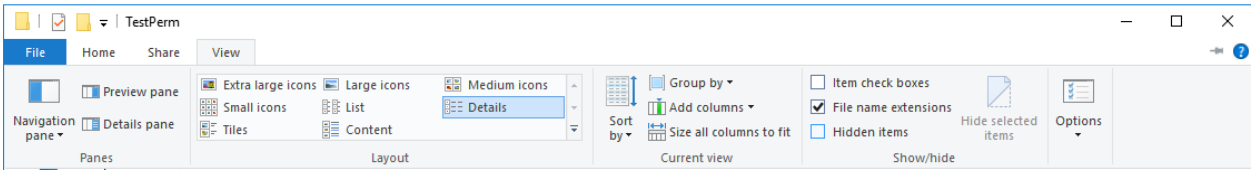

Figure 11: Enabling file extensions

After this we will create a text file called “TestBatch.bat” and when it gives us a warning about changing the file extension, we will accept the warning. After this, we will edit the file we created and add the following lines of text as seen in the figure below.

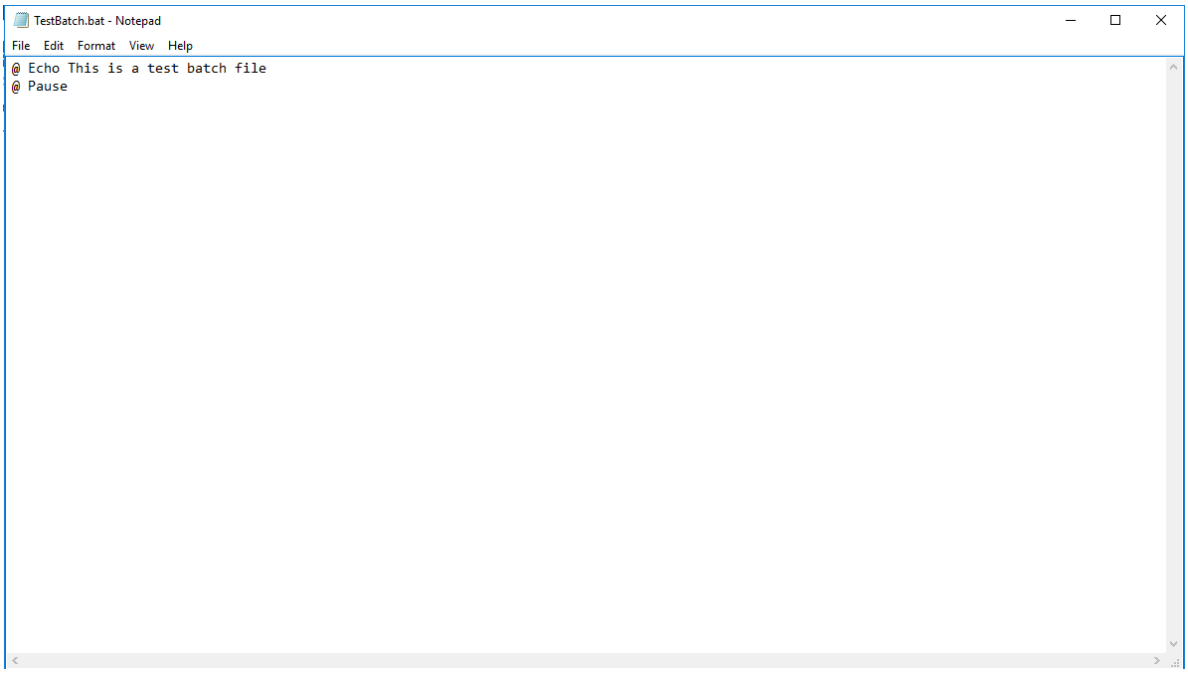

Figure 12: Creating our .bat file

When we run the file afterwards, we see this

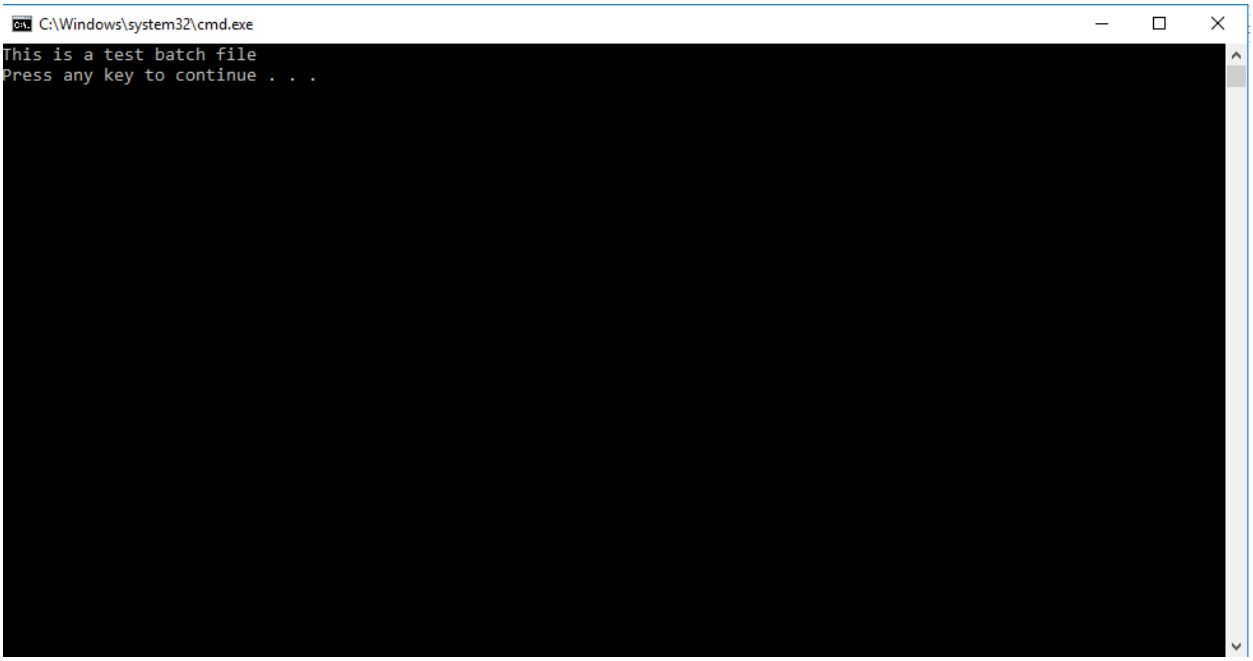

Figure 13: Testing the .bat file

After closing the command prompt, we will right click and go to properties, then security and disable inheritance on the TestBatch file that we created earlier.

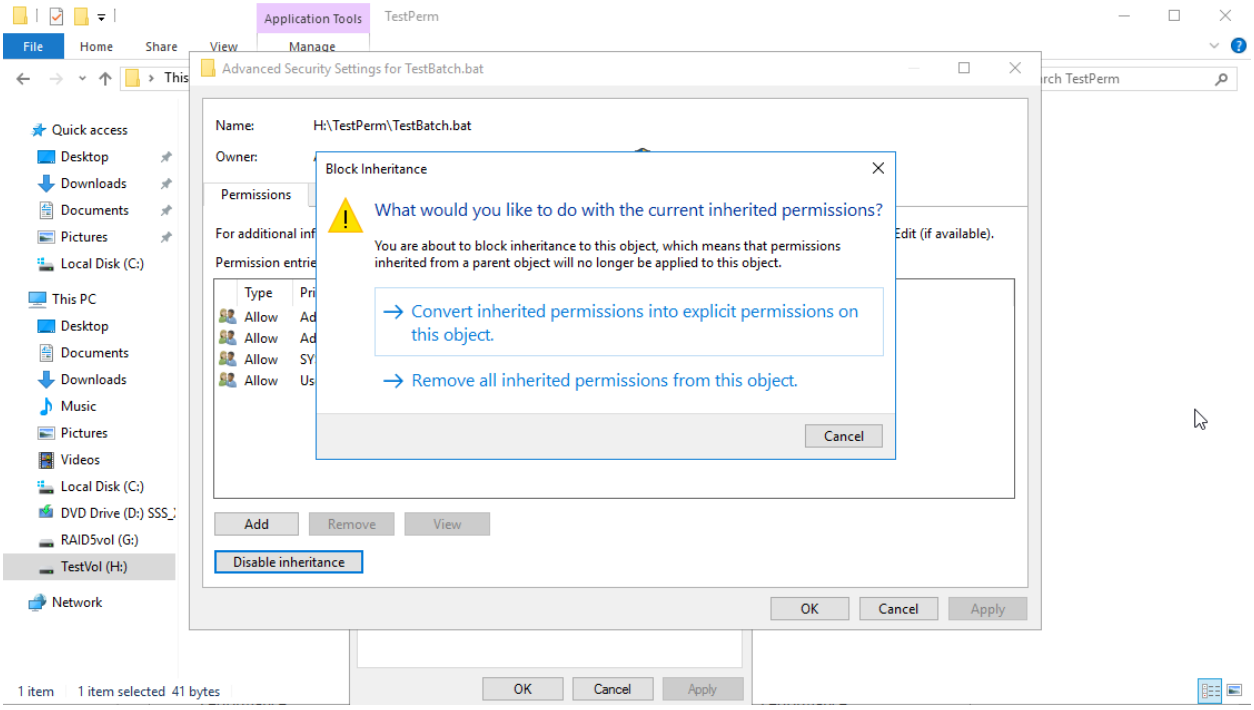

Figure 14: Disabling the Inheritance, converting the permissions

From here, we will select “Convert inherited permissions into explicit permissions on this object” then click “ok” and navigate back to our security tab in TestBatch.bat

Once there, we will click on the “Advanced” tab and click “Users” once it opens.

Here, the only permission that we will keep is the “Read” permission, removing the Read & Execute permission that is checked.

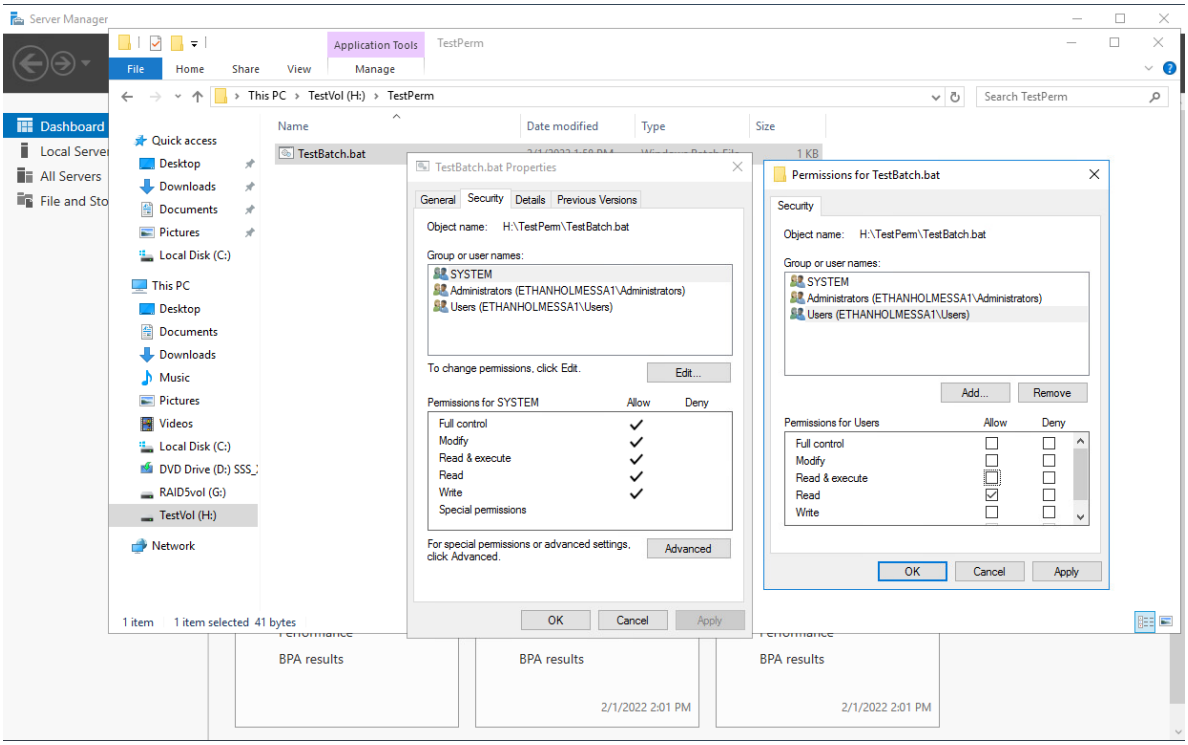

Figure 15: Removing read & write execute

After changing the permission settings, we will then switch user accounts to a test account that we created earlier, from this account we will try accessing the “TestBatch” file that we created on the Administrator account.

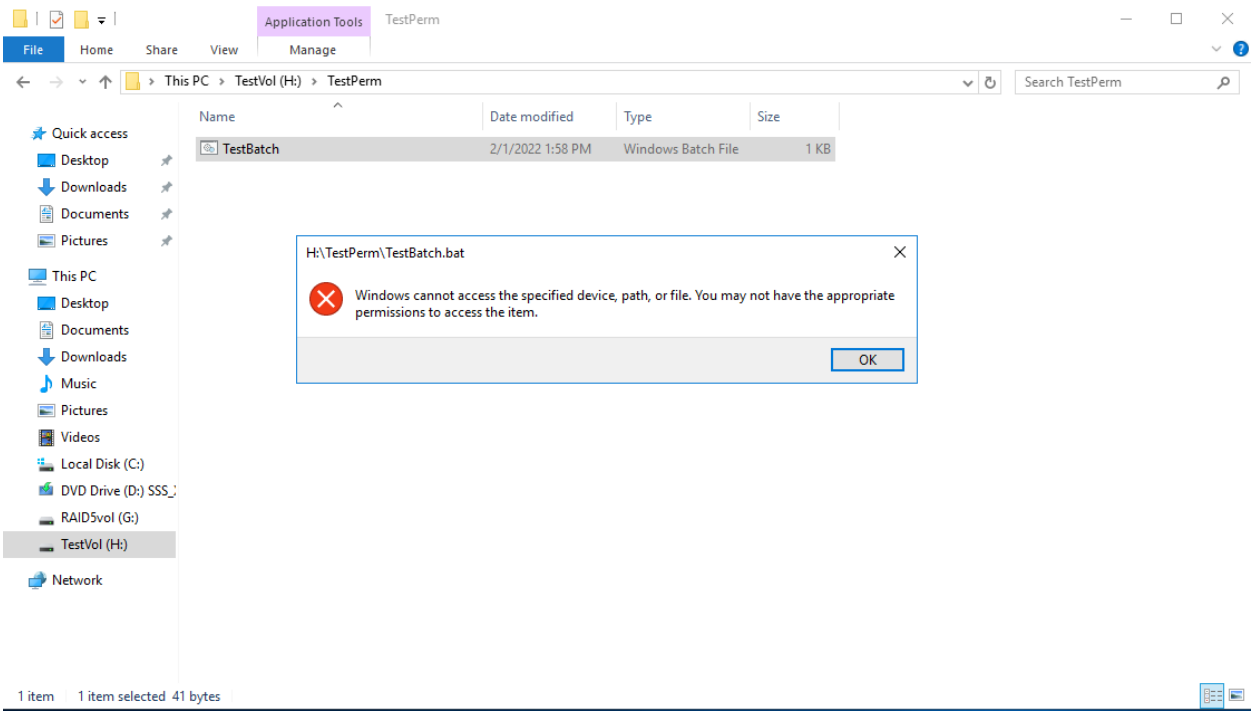

Figure 16: Error from loss of a permissions

Instead, we are given an error message detailing that we do not have appropriate permissions to view this file, we can still edit the file thanks to the Read permissions that we gave the user. However, because of our lack of Write permissions, if we try to save the file, we are given another error message.

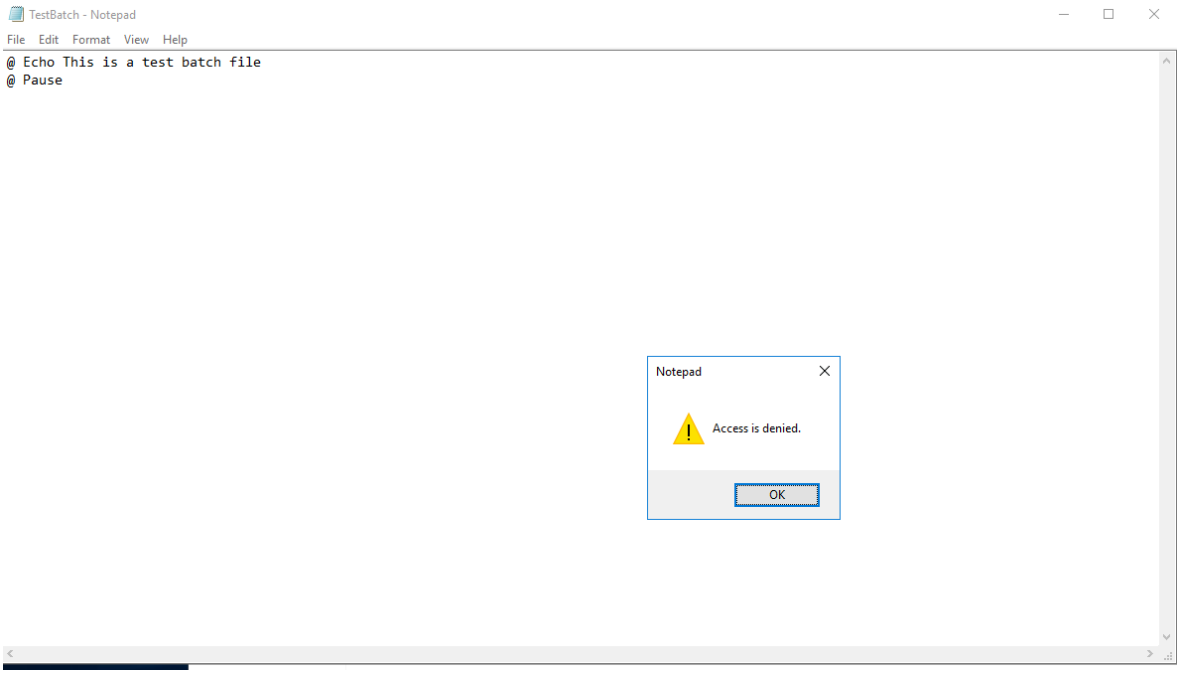

Figure 17: Error when trying to write

However, if we open this “TestBatch” again, and save as and change the name of the file to “NewBatch” we can do that.

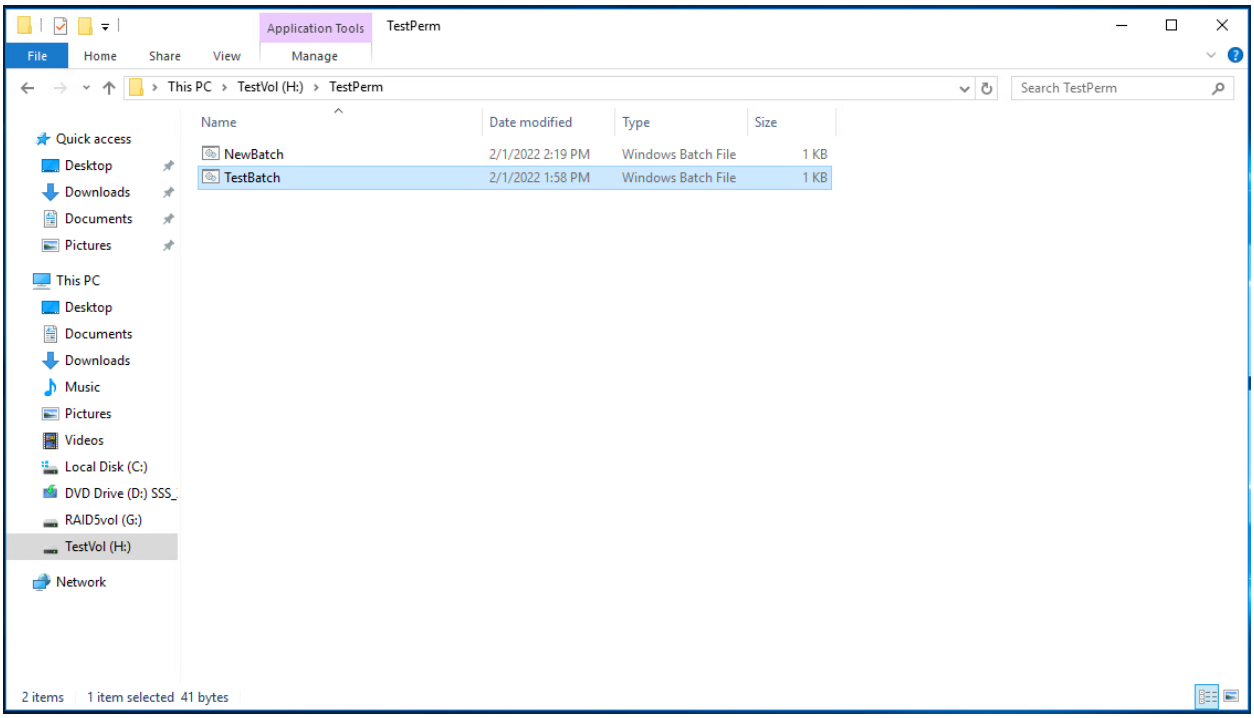

Figure 18: Creating a new .bat file

If we inspect the security properties of the file, we can see that the Test account that we made this “NewBatch” on is given full control of what is effectively the same file. This is because of the “Creator Owner” permissions that are on the parent folder that we created this file in.

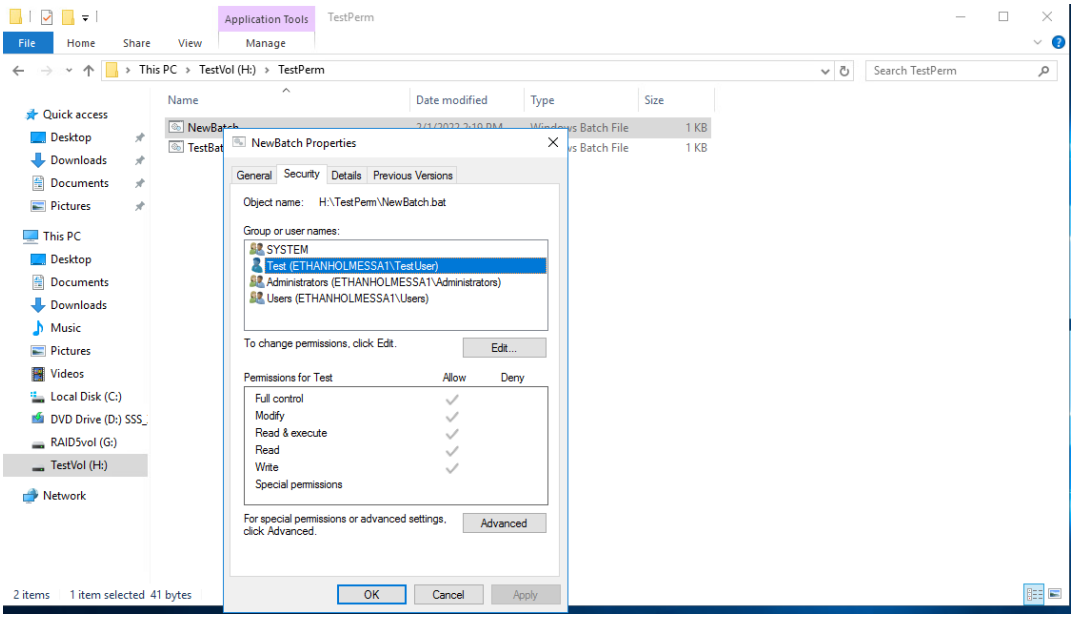

Figure 19: The groups with permissions

After disabling inheritance on this file as well, we will then remove the users' group as well as the Test group by clicking edit, follow by remove from the security tab leaving System and Administrators.

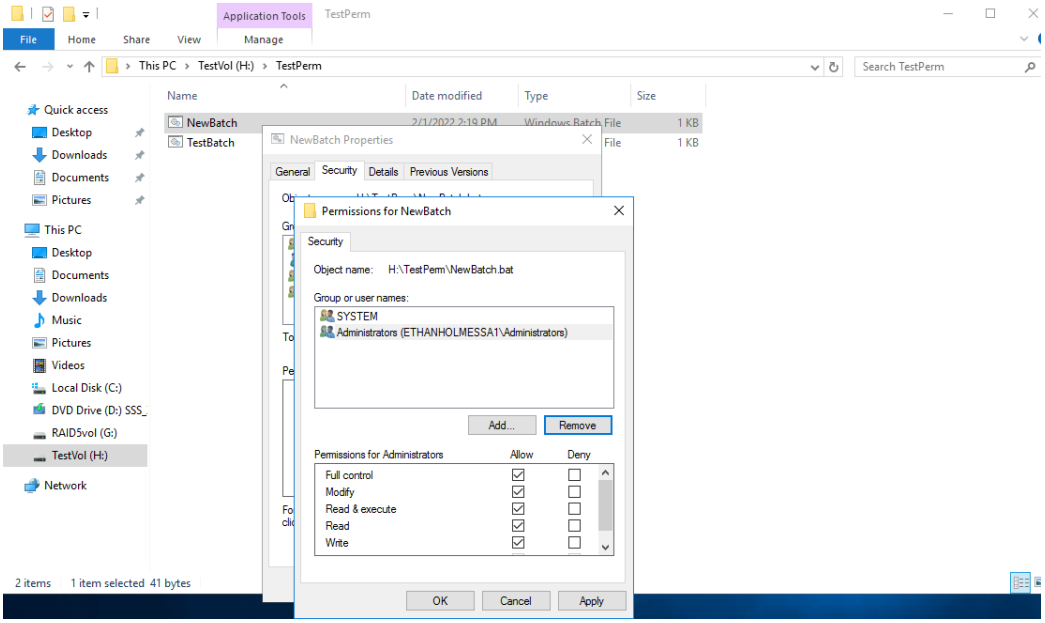

Figure 20: Deleting groups permissions

Now when we try to access the "NewBatch" file, you can see that we no loner have permission as we have removed the test permissions to the file.

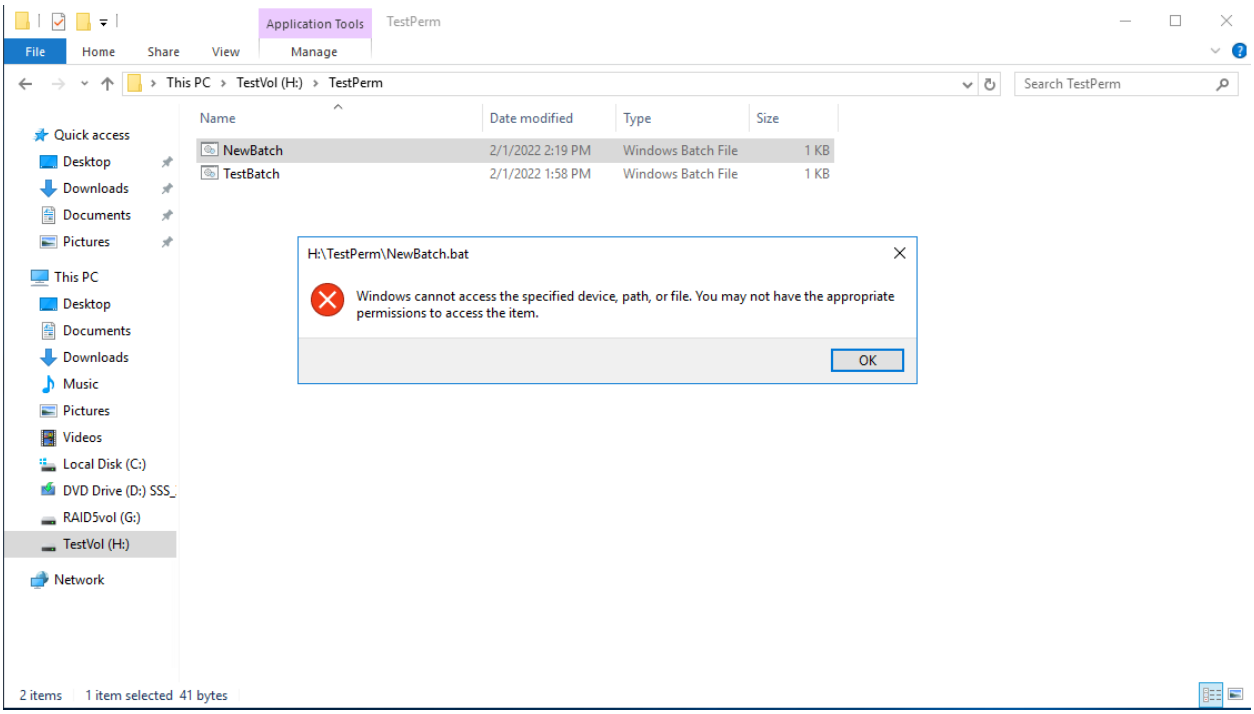

Figure 21: Losing access after deleting permissions

However, because we are the owner of the “NewBatch” file, we can just give ourselves permission to access the file again by adding ourselves back to the permissions

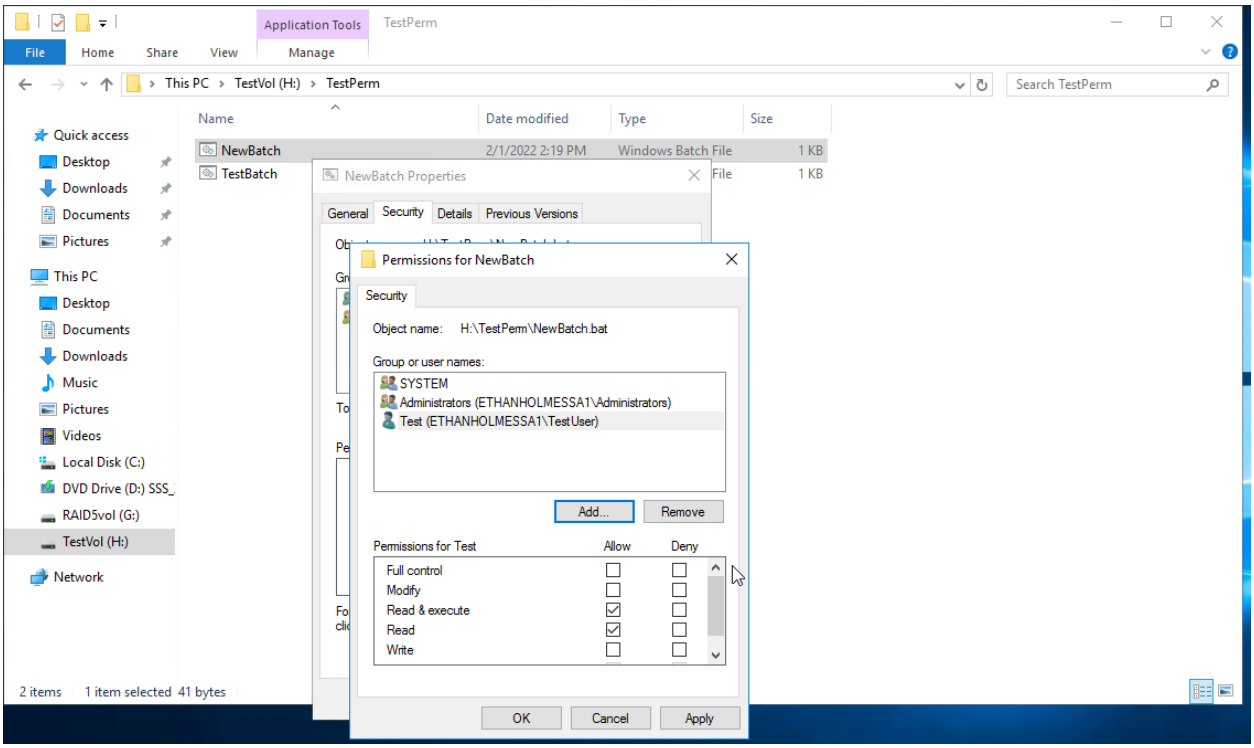

Figure 22: Adding permissions back

Summary:

Over the course of this lab, I learned methods for control permissions that I found to be helpful, however they seem to be somewhat lacking in terms of total control using the controls that we learned today. Hopefully going further forward we can learn even more effective manors of dealing with file permissions.