CYBR 510- Homework 2: File and Directory Access Permissions Manipulation in PowerShell

Due on Sunday 9/8/2024

Background

This homework focuses on providing you an introduction to object creation, management, and access permissions in Windows Powershell. This homework has been adapted from lessons contained in Dent, C. 2021. *Mastering Powershell Scripting*. https://learning.oreilly.com/library/view/mastering-powershell-scripting/9781800206540/

Requirements: Windows 10

If you do not have a Windows 10 PC, please work with a team partner who has a Windows machine, or you can set up a Windows 10 VM using VirtualBox, and a free copy of Windows 10 provided to WVU students:

> List of free software for WVU students:
> https://wvu.atlassian.net/servicedesk/customer/portal/5/article/298680653
>
> Oracle VirtualBox:
>
> https://www.virtualbox.org/

Submission:

Create a new word document, add a cover, list group members, homework number, and date. For each of the questions below, provide question number and the answer.

Windows Permissions:

*Be sure you are running PowerShell as an administrator. Search for PowerShell in the start menu/search bar, right click on Windows PowerShell and click on "Run as Administrator". Once your PowerShell prompt has popped up, copy each of command in the colored text below and paste it in the PowerShell, then hit enter to execute. You can highlight this text, copy it to the clipboard. Powershell does not recognize Ctrl+V as the paste function. To paste data from the clipboard into powershell, just right click anywhere on the powershell prompt. It may take a second, but the code will be copied. Once you right click, the code will automatically execute.

```
New-Item C:\Temp\ACL -ItemType Directory -Force
1..5 | ForEach-Object {
    New-Item C:\Temp\ACL\$_ -ItemType Directory -Force
    'content' | Out-File "C:\Temp\ACL\$_\$_.txt"
```

```
    New-Item C:\Temp\ACL\$_\$_ -ItemType Directory -
Force
    'content' | Out-File "C:\Temp\ACL\$_\$_\$_.txt"
    }
```

1. What happened? What did this command just do? Check your C:\Temp\ACL folder if you aren't sure.

2. Type in `Get-Acl C:\Temp\ACL\1 -Audit | Format-List`
   Hit Enter. Take a screenshot of the output. What permissions exist on this file? Also, notice the Sddl output at the bottom. Look at the S-1-5… (it may be slightly different on your system). What do you think this string represents? Look up SIDs if you aren't sure.

3. Now navigate to the Temp folder. You can type cd\ to go up a directory, and then cd Temp\ACL to get to the ACL directory. Type **dir** or **ls** hit enter. You should see 5 directories. Navigate to directory "1". Your prompt should look like "PS C:\temp\ACL\1. Look at the attributes for the file 1.txt. Does this file have any specific attributes?

4. Lets add a read only attribute. Type:

   ```
   Set-ItemProperty .\1.txt -Name IsReadOnly -Value
   $true
   ```

   Then type **dir** and hit enter again. What do you think we did to this file? How can you tell this command worked? How could you alter the command to remove this permission?

5. We can also add other attributes like the hidden attribute. Type in:

   ```
   (get-item 1.txt -force).Attributes +=
   'Hidden'
   ```

   Type in **dir**. Hit enter. Take a screenshot of the output. What do you think happened here. How can we modify this command to reverse what we just did? List the correct commands to remove the hidden and read only attributes and post a screenshot of the restored 1.txt file.

6. Now lets set up some rule protections. First, lets check the access permissions for the file 2.txt
   Type:

   ```
   Get-Acl C:\Temp\ACL\2 | Select-Object -
   ExpandProperty Access | Select-Object
   FileSystemRights, IdentityReference,
   IsInherited
   ```

Take a screenshot of the output.

ACLs by default obtain rules from their parent objects. We can enable rule protection for the access ACL using the SetAccessRuleProtection method, or the SetAuditRuleProtection method on the ACL itself. Type in:

```
$acl = Get-Acl C:\Temp\ACL\2
$acl.SetAccessRuleProtection($true, $true)
Set-Acl C:\Temp\ACL\2 -AclObject $acl
```

Copied rules will only appear on the ACL (as explicit rules) after Set-Acl has been run.

If access rule protection is subsequently re-enabled, copied rules are not removed. The resulting ACL will contain both inherited and explicit versions of each of the rules. Inheritance can be re-enabled as follows:

Then type:

```
$acl = Get-Acl C:\Temp\ACL\2
$acl.SetAccessRuleProtection($false, $false)
Set-Acl C:\Temp\ACL\2 -AclObject $acl
```

7. Now check the ACL of that file again using the command at the start of step 6. What do you notice?

8. Now we can also discard access rules as well. Take a look at the access rules for 3.txt: Modify the command at the top of Step 7 to check the ACL of 3.txt. Take a screenshot of the output.

   Let's remove the access permissions from folder 3:

   ```
   $acl = Get-Acl C:\Temp\ACL\3
   $acl.SetAccessRuleProtection($true, $false)
   Set-Acl C:\Temp\ACL\3 -AclObject $acl
   ```

9. Now type in:

   `Get-ChildItem C:\Temp\ACL\3`

10. What happened? Post a screenshot of the output.

11. What are your observations about windows file and directory permissions? Why are ACLs important? Do you think it's necessary for all files or directories to have an ACL? Based upon what you just observed, explain why or why not?
12. How can we modify the above commands to restore access? Write down your solution.

Grading Rubric

Question No/ detail	**Points**
1-10	8 points each
11-12	10 points each