



# GUIDE

## Using Access Permissions and Rights to Secure a Folder

*v1.0.0*

Author:

**Eldon Gabriel**

August 28, 2025

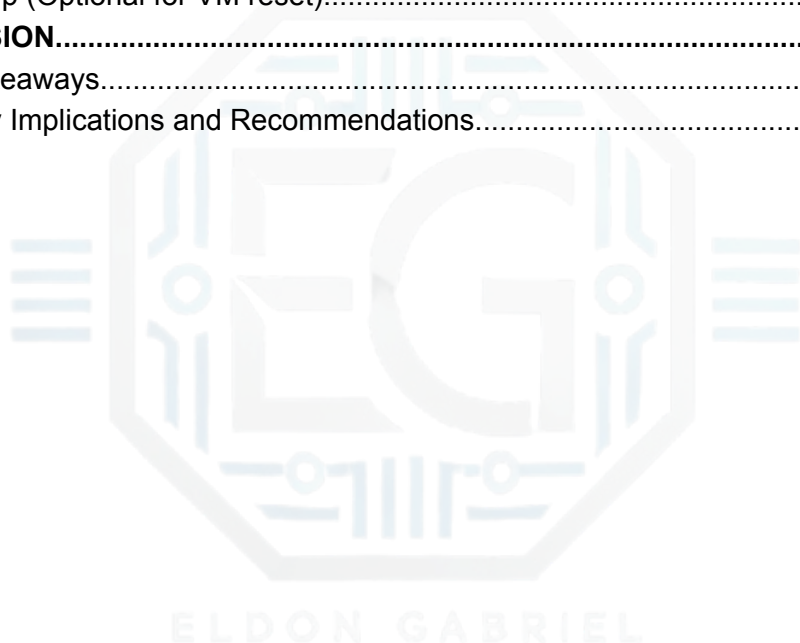
This guide is based on my independent practice and understanding of Windows access permissions and Group Policy, intended for portfolio demonstration.



Cybersecurity Professional | IT Security Consultant

## TABLE OF CONTENTS


<b>TABLE OF CONTENTS</b> .....	<b>1</b>
<b>REVISION HISTORY</b> .....	<b>2</b>
<b>1.0 SECURING FOLDERS WITH ACCESS PERMISSIONS</b> .....	<b>3</b>
1.1 Project Description.....	3
1.2 Show Existing Users and Groups.....	3
1.3 Create Users and Groups.....	4
1.4 Create Confidential Folder & Apply Permissions.....	5
1.5 Restrict PowerShell Access via Group Policy.....	6
<b>2.0 TESTING AND VERIFICATION</b> .....	<b>7</b>
2.1 Test Permissions.....	7
2.2 Clean Up (Optional for VM reset).....	8
<b>3.0: CONCLUSION</b> .....	<b>9</b>
3.1 Key Takeaways.....	9
3.2 Security Implications and Recommendations.....	9





Cybersecurity Professional | IT Security Consultant

## REVISION HISTORY

Version	Date	 Author	Description of Changes
v1.0.0	08/28/2025	Eldon G.	Initial draft.





Cybersecurity Professional | IT Security Consultant

## 1.0 SECURING FOLDERS WITH ACCESS PERMISSIONS

### 1.1 Project Description

This guide demonstrates how to create and manage local users, groups, and access permissions in Windows. The objective is to secure a confidential folder so that only authorized groups can access it. Additionally, PowerShell will be restricted for specific groups using Local Group Policy.

---

### 1.2 Show Existing Users and Groups

Run the following commands to display current users and groups:

```
bash
net user
net localgroup
```

✓ **Visual:** A list of users and groups is displayed (Admin, Finance, HR, and other defaults).

---



## 1.3 Create Users and Groups

Create new groups and assign users to them:

```
bash

:: Create groups

net localgroup [Group Name] /add

:: Create a [User Name], [New Password] and add to a [Group Name]

net user user1 [Input New Password] /add

net localgroup [Group Name] [User Name] /add
```

**Note:** Repeat for multiple users and groups.

### To Verify

```
bash

net user

net localgroup
```

---



## 1.4 Create Confidential Folder & Apply Permissions

```
bash

:: Create folder
mkdir C:\[Folder Name]

:: Remove inherited permissions
icacls "C:\[Folder Name]" /inheritance:r

:: Remove generic Users
icacls "C:\[Folder Name]" /remove "Users"
icacls "C:\[Folder Name]" /remove "Authenticated Users"

:: Grant access
icacls "C:\[Folder Name]" /grant [Group Name):(OI)(CI)(F)
icacls "C:\[Folder Name]" /grant Administrators:(OI)(CI)(F)

:: Verify
icacls "C:\[Folder Name]"
```





## 1.5 Restrict PowerShell Access via Group Policy

1. Open Local Group Policy Editor (`gpedit.msc`)
  - Navigate to:  
User Configuration → Windows Settings → Security Settings → Software Restriction Policies
  - If no policy exists: Right-click **Software Restriction Policies** → **New Software Restriction Policies**
2. Create Path Rules to block PowerShell for specific groups:

`%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe`

`%SystemRoot%\SysWOW64\WindowsPowerShell\v1.0\powershell.exe`

- Set Security Level = **Disallowed**
- Apply to: `[Group Name]`



## 2.0 TESTING AND VERIFICATION

### 2.1 Test Permissions

**As [User Name] (Member of [Group Name]):**

```
bash

runas /user:[ComputerName]\[User Name] cmd
cd C:\[Folder Name] → should succeed/fail if granted/blocked
powershell          → should succeed/fail if granted/blocked
```

**Note:** Repeat for multiple users.

**As Administrator:**

```
bash

cd C:\[Folder Name] → should succeed
powershell          → should succeed
```

---





## 2.2 Clean Up (Optional for VM reset)

```
bash

takeown /F C:\[Folder Name] /R /D Y
rmdir /S /Q C:\[Folder Name]
net user [User Name] /delete
net localgroup [Group Name] /delete
```

✓ Visual: Folder, users, and groups removed.

### To Verify

```
bash

net user

net localgroup

icacls "C:\[Folder Name]"
```

---

ELDON GABRIEL



## 3.0: CONCLUSION

### 3.1 Key Takeaways

- Access permissions can restrict sensitive data to specific groups.
- Group Policy rules prevent misuse of administrative tools.
- Testing with multiple accounts confirms the effectiveness of security controls.

### 3.2 Security Implications and Recommendations

- Use the principles of least privilege when granting access to data.
- Apply consistent folder and tool restrictions for business-critical assets.
- Document user/group structures for auditing and incident response.

