Jonathan Torrence

Kevin Olson

CNIT 361-001

12-02-2024

PowerShell Automated Backup & Partial Restore of Security Configurations

Introduction

This project focuses on creating PowerShell scripts for backing up and partially restoring key system security configurations. The automation includes backing up Windows Firewall rules, registry keys, and folder permissions. To maintain safety and reproducibility, the project was developed and tested in a virtualized environment.

The tools used include:

- **Oracle VirtualBox** running Windows 10 Enterprise

- **PowerShell ISE** as the scripting environment
- **VirtualBox** for virtual machine setup and management.

This project demonstrates how PowerShell can be leveraged for system administration to securely back up critical configurations and restore them when needed.

Requirements

1. **Host System:**
   - A Windows 10 or 11 box with Oracle VirtualBox installed.
2. **Virtual Machine:**
   - Windows 10 Enterprise ISO (trial version).
   - Oracle VirtualBox configured with 4 GB RAM, 2 processors, 40 GB disk space.
3. **Scripts:**
   - BackupFirewall.ps1
   - RestoreFirewall.ps1
   - BackupAll.ps1

- EnhancedBackup.ps1

4. **Administrator Access:**

- PowerShell ISE must be run as Administrator.

5. **Files Created:**

- Firewall rules backup: FirewallRulesBackup.wfw
- Registry backup: RegistryBackup.reg
- Folder permissions backup: FolderPermissions.txt

Setup Instructions

1. **Install Oracle VirtualBox**

- Download VirtualBox from virtualbox.org.
- Complete the installation wizard and verify functionality.

2. **Set Up the Virtual Machine**

- Download the Windows 10 Enterprise ISO from Microsoft’s evaluation site.

- In VirtualBox, create a virtual machine with the following settings: 4 GB RAM, 2 processors, 40 GB disk space
- Attach the ISO and install Windows 10.

3. **Prepare the Environment**

- Install PowerShell Ise if it is not already installed.
- Create a working directory for your scripts and backups:

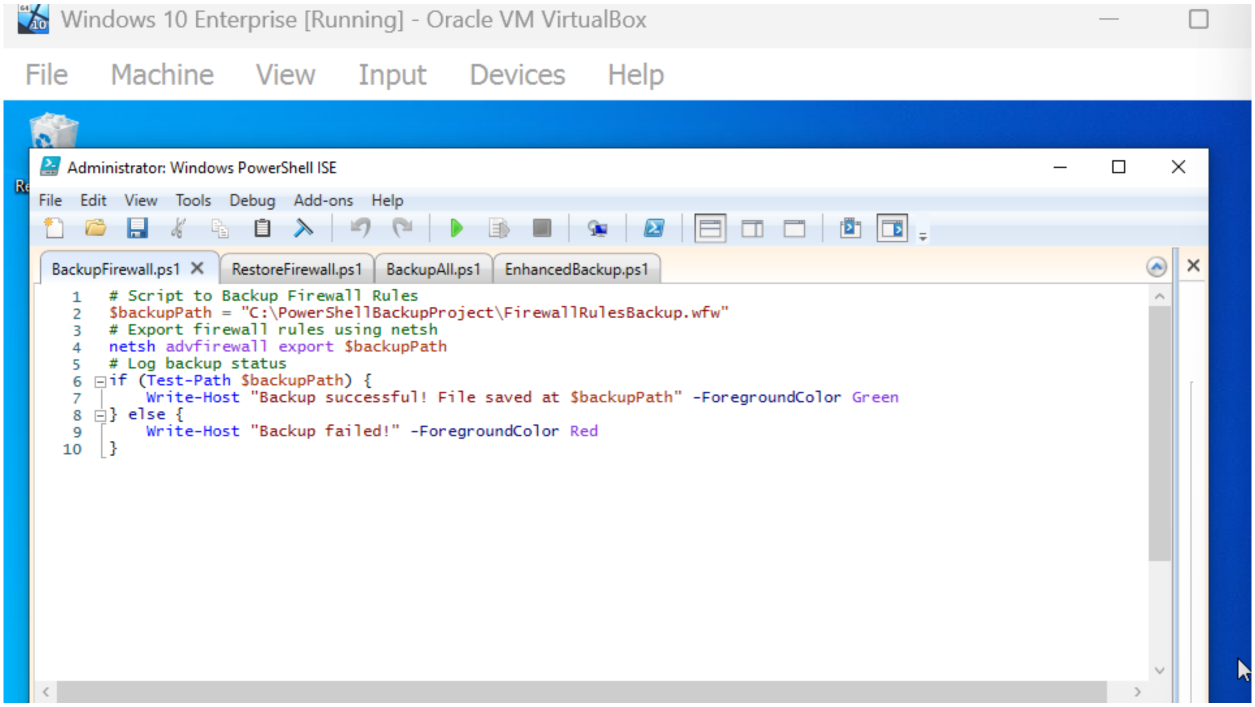
```
New-Item -ItemType Directory -Path "C:\PowerShellBackupProject"
```

Script Descriptions and Usage

BackupFirewall.ps1

- **Purpose:** Exports the existing Windows Firewall rules to a .wfw file.

Script Code:


Windows 10 Enterprise [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
BackupFirewall.ps1
RestoreFirewall.ps1
BackupAll.ps1
EnhancedBackup.ps1

```
# Script to Backup Firewall Rules
$backupPath = "C:\PowerShellBackupProject\FirewallRulesBackup.wfw"
# Export firewall rules using netsh
netsh advfirewall export $backupPath
# Log backup status
if (Test-Path $backupPath) {
    Write-Host "Backup successful! File saved at $backupPath" -ForegroundColor Green
} else {
    Write-Host "Backup failed!" -ForegroundColor Red
}
```

- **How to Run:**

    1. Save the script in C:\PowerShellBackupProject.

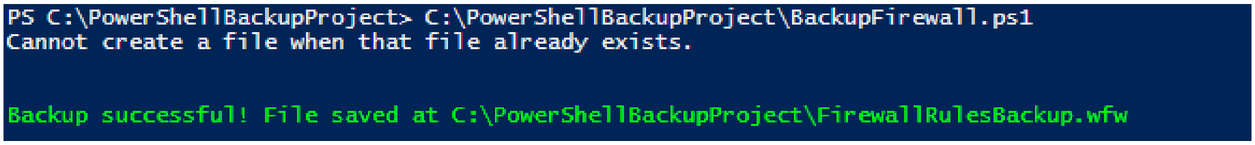
    2. Open PowerShell ISE as Administrator

    3. Execute the script:

        .\BackupFirewall.ps1

- **Expected Output:**

    A file named FirewallRulesBackup.wfw is created in the project folder.



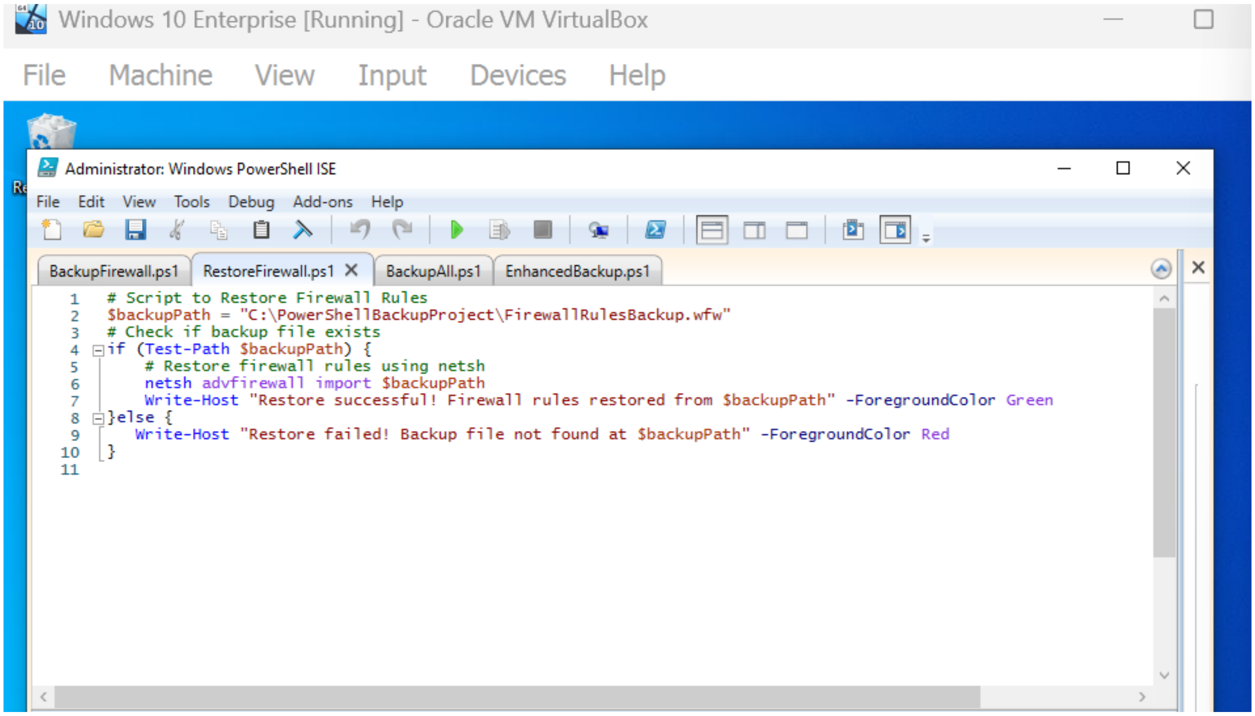
```
PS C:\PowerShellBackupProject> C:\PowerShellBackupProject\BackupFirewall.ps1
Cannot create a file when that file already exists.

Backup successful! File saved at C:\PowerShellBackupProject\FirewallRulesBackup.wfw
```

RestoreFirewall.ps1

- **Purpose:** Restores firewall rules from a previously created .wfw file.

Script Code:



```
# Script to Restore Firewall Rules
$backupPath = "C:\PowerShellBackupProject\FirewallRulesBackup.wfw"
# Check if backup file exists
if (Test-Path $backupPath) {
    # Restore firewall rules using netsh
    netsh advfirewall import $backupPath
    Write-Host "Restore successful! Firewall rules restored from $backupPath" -ForegroundColor Green
}else {
   Write-Host "Restore failed! Backup file not found at $backupPath" -ForegroundColor Red
}
```

- **How to Run:**

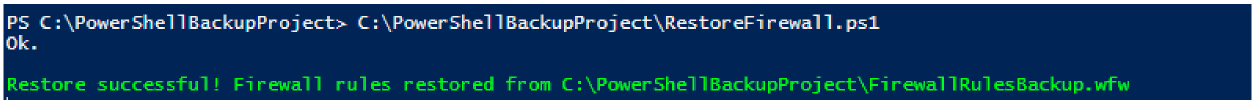
1. Save the script in C:\PowerShellBackupProject.
2. Open PowerShell ISE as Administrator

3. Execute the script:

   .\RestoreFirewall.ps1

- **Expected Output:**

  The firewall rules are restored to their original state.



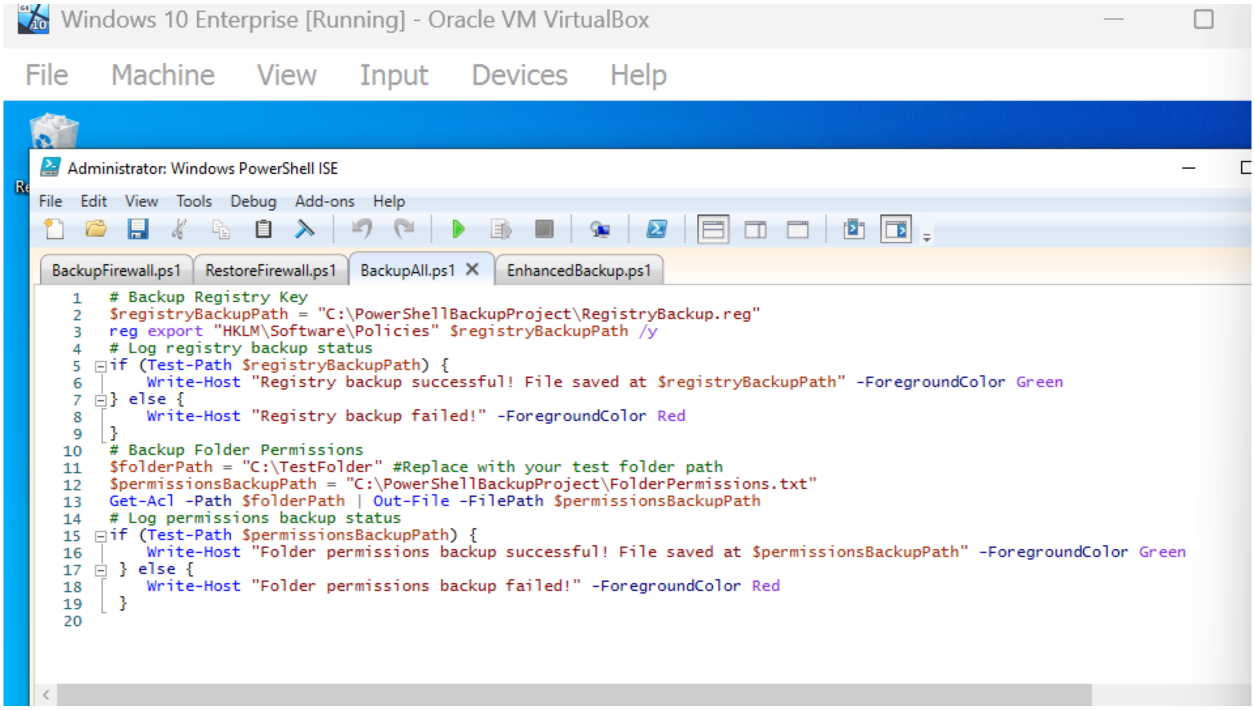
```
PS C:\PowerShellBackupProject> C:\PowerShellBackupProject\RestoreFirewall.ps1
Ok.

Restore successful! Firewall rules restored from C:\PowerShellBackupProject\FirewallRulesBackup.wfw
```

BackupAll.ps1

- **Purpose:** Combines multiple backup tasks, including Exporting Windows Firewall rules. Backing up registry keys using reg export. Saving folder permissions with Get-Acl.

Script Code:


Windows 10 Enterprise [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
BackupFirewall.ps1
RestoreFirewall.ps1
BackupAll.ps1
EnhancedBackup.ps1

```
# Backup Registry Key
$registryBackupPath = "C:\PowerShellBackupProject\RegistryBackup.reg"
reg export "HKLM\Software\Policies" $registryBackupPath /y
# Log registry backup status
if (Test-Path $registryBackupPath) {
    Write-Host "Registry backup successful! File saved at $registryBackupPath" -ForegroundColor Green
} else {
    Write-Host "Registry backup failed!" -ForegroundColor Red
}
# Backup Folder Permissions
$folderPath = "C:\TestFolder" #Replace with your test folder path
$permissionsBackupPath = "C:\PowerShellBackupProject\FolderPermissions.txt"
Get-Acl -Path $folderPath | Out-File -FilePath $permissionsBackupPath
# Log permissions backup status
if (Test-Path $permissionsBackupPath) {
    Write-Host "Folder permissions backup successful! File saved at $permissionsBackupPath" -ForegroundColor Green
 } else {
    Write-Host "Folder permissions backup failed!" -ForegroundColor Red
 }
```

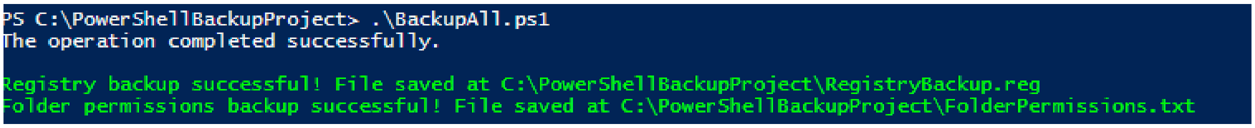
- **How to Run:**

1. Save the script in C:\PowerShellBackupProject.
2. Open PowerShell ISE as Administrator
3. Execute the script:

```
.\BackupAll.ps1
```

- **Expected Output:**

The following files are created: RegistryBackup.reg & FolderPermissions.txt



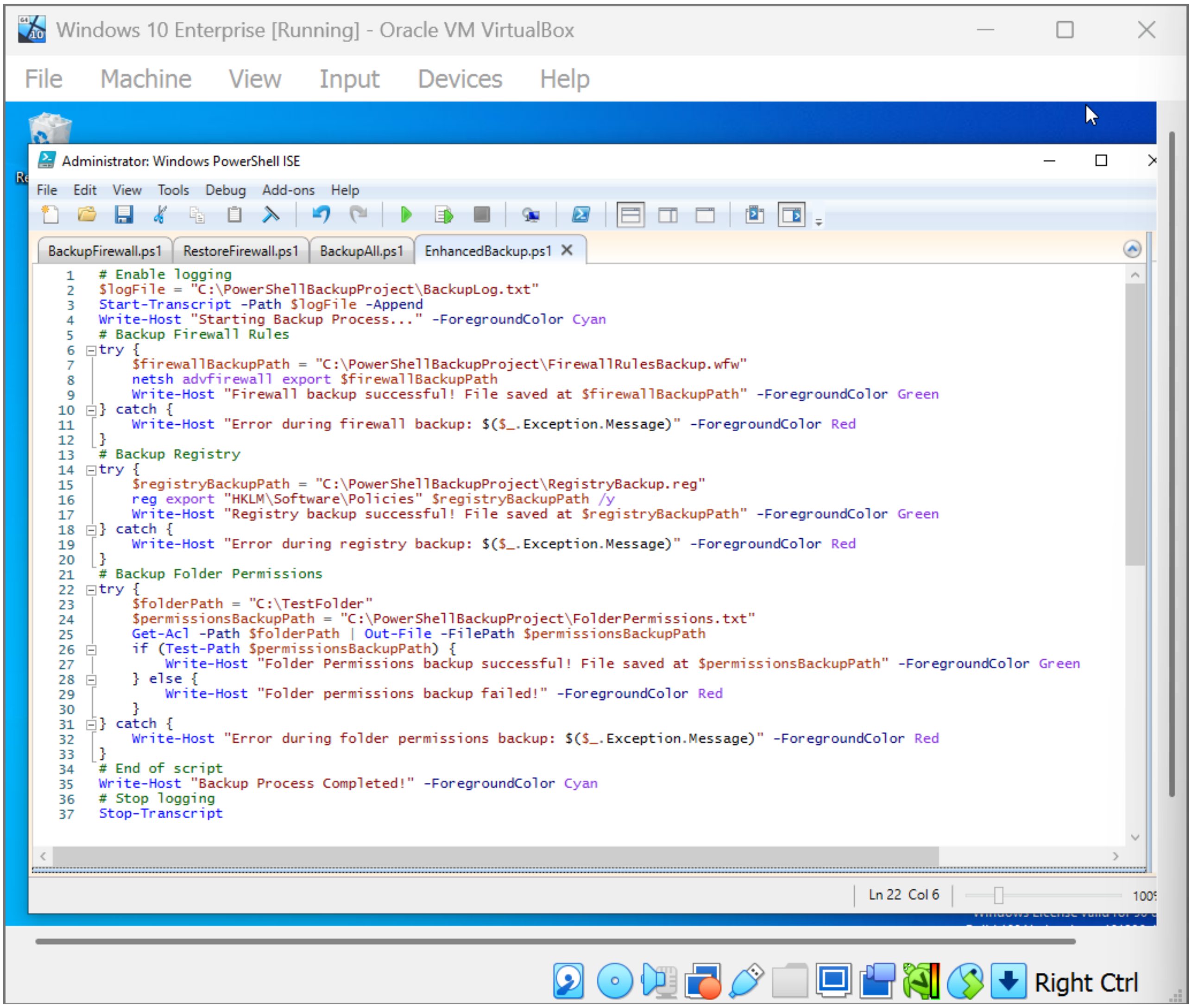
```
PS C:\PowerShellBackupProject> .\BackupAll.ps1
The operation completed successfully.

Registry backup successful! File saved at C:\PowerShellBackupProject\RegistryBackup.reg
Folder permissions backup successful! File saved at C:\PowerShellBackupProject\FolderPermissions.txt
```

EnhancedBackup.ps1

- **Purpose:** Adds logging and error handling to the backup process.

Script Code:


Windows 10 Enterprise [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
BackupFirewall.ps1
RestoreFirewall.ps1
BackupAll.ps1
EnhancedBackup.ps1
Ln 22 Col 6
Right Ctrl

```powershell
# Enable logging
$logFile = "C:\PowerShellBackupProject\BackupLog.txt"
Start-Transcript -Path $logFile -Append
Write-Host "Starting Backup Process..." -ForegroundColor Cyan
# Backup Firewall Rules
try {
    $firewallBackupPath = "C:\PowerShellBackupProject\FirewallRulesBackup.wfw"
    netsh advfirewall export $firewallBackupPath
    Write-Host "Firewall backup successful! File saved at $firewallBackupPath" -ForegroundColor Green
} catch {
    Write-Host "Error during firewall backup: $($_.Exception.Message)" -ForegroundColor Red
}
# Backup Registry
try {
    $registryBackupPath = "C:\PowerShellBackupProject\RegistryBackup.reg"
    reg export "HKLM\Software\Policies" $registryBackupPath /y
    Write-Host "Registry backup successful! File saved at $registryBackupPath" -ForegroundColor Green
} catch {
    Write-Host "Error during registry backup: $($_.Exception.Message)" -ForegroundColor Red
}
# Backup Folder Permissions
try {
    $folderPath = "C:\TestFolder"
    $permissionsBackupPath = "C:\PowerShellBackupProject\FolderPermissions.txt"
    Get-Acl -Path $folderPath | Out-File -FilePath $permissionsBackupPath
    if (Test-Path $permissionsBackupPath) {
        Write-Host "Folder Permissions backup successful! File saved at $permissionsBackupPath" -ForegroundColor Green
    } else {
        Write-Host "Folder permissions backup failed!" -ForegroundColor Red
    }
} catch {
    Write-Host "Error during folder permissions backup: $($_.Exception.Message)" -ForegroundColor Red
}
# End of script
Write-Host "Backup Process Completed!" -ForegroundColor Cyan
# Stop logging
Stop-Transcript
```

- **How to Run:**

  1. Save the script in C:\PowerShellBackupProject.

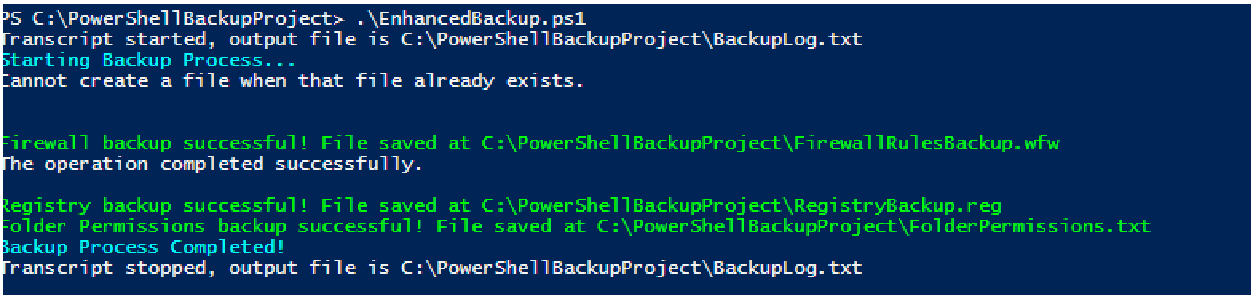
  2. Open PowerShell ISE as Administrator

  3. Execute the script:

.\EnhancedBackup.ps1

- **Expected Output:**

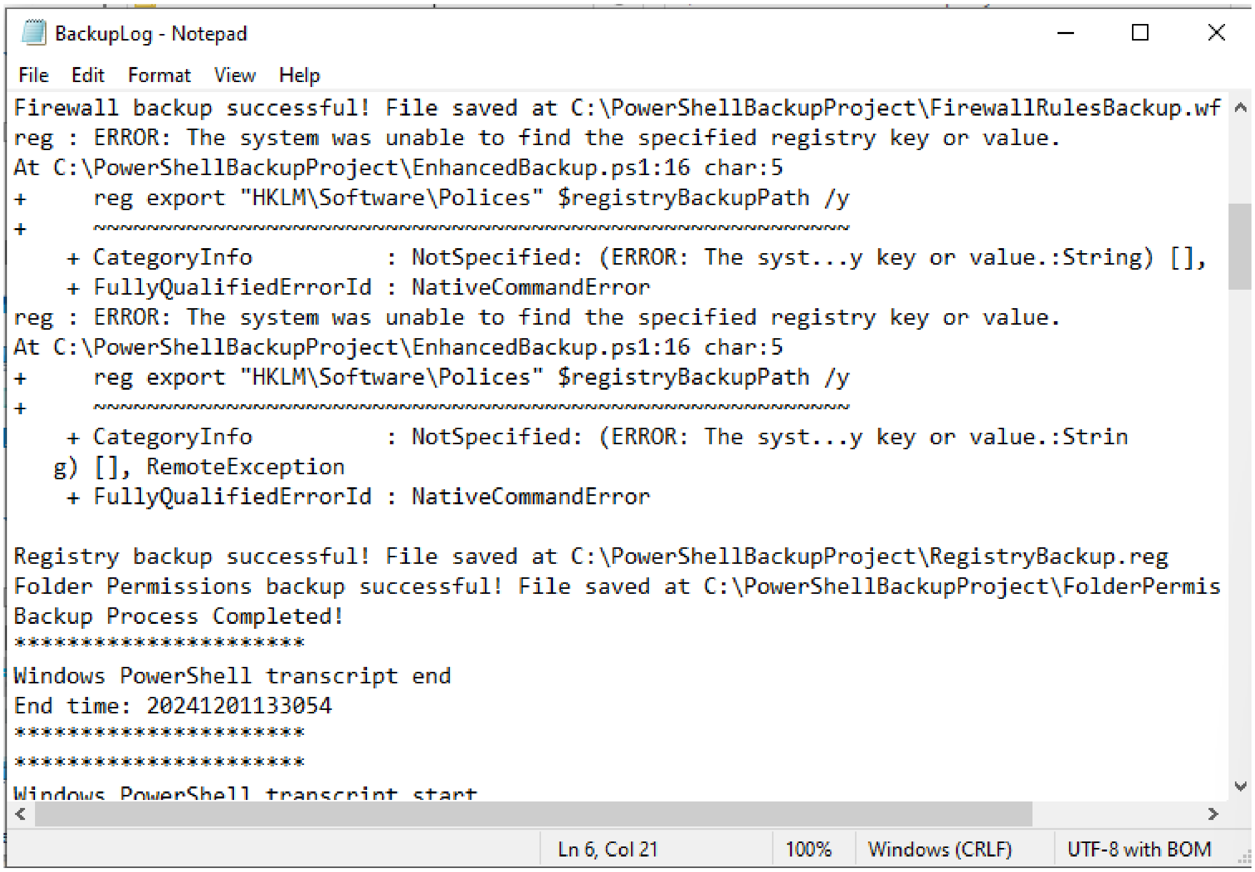
  Logs are created in BackupLog.txt detailing success and errors for each backup step.



```
PS C:\PowerShellBackupProject> .\EnhancedBackup.ps1
Transcript started, output file is C:\PowerShellBackupProject\BackupLog.txt
Starting Backup Process...
Cannot create a file when that file already exists.

Firewall backup successful! File saved at C:\PowerShellBackupProject\FirewallRulesBackup.wfw
The operation completed successfully.

Registry backup successful! File saved at C:\PowerShellBackupProject\RegistryBackup.reg
Folder Permissions backup successful! File saved at C:\PowerShellBackupProject\FolderPermissions.txt
Backup Process Completed!
Transcript stopped, output file is C:\PowerShellBackupProject\BackupLog.txt
```



```
Firewall backup successful! File saved at C:\PowerShellBackupProject\FirewallRulesBackup.wf
reg : ERROR: The system was unable to find the specified registry key or value.
At C:\PowerShellBackupProject\EnhancedBackup.ps1:16 char:5
+     reg export "HKLM\Software\Polices" $registryBackupPath /y
+     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (ERROR: The syst...y key or value.:String) [],
    + FullyQualifiedErrorId : NativeCommandError
reg : ERROR: The system was unable to find the specified registry key or value.
At C:\PowerShellBackupProject\EnhancedBackup.ps1:16 char:5
+     reg export "HKLM\Software\Polices" $registryBackupPath /y
+     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (ERROR: The syst...y key or value.:Strin
   g) [], RemoteException
    + FullyQualifiedErrorId : NativeCommandError

Registry backup successful! File saved at C:\PowerShellBackupProject\RegistryBackup.reg
Folder Permissions backup successful! File saved at C:\PowerShellBackupProject\FolderPermis
Backup Process Completed!
**********************
Windows PowerShell transcript end
End time: 20241201133054
**********************
**********************
Windows PowerShell transcript start
```

Test Cases

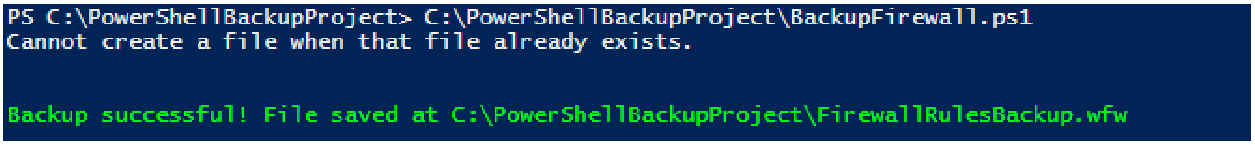
Test Case 1: Backup of Firewall Rule

Script Used: BackupFirewall.ps1

Steps:

1. Run the script.
2. Check PowerShell for output messages.
3. Verify that FirewalRulesBackup.wfw exists in the project folder.

Expected Outcome: The .wfw file is successfully created.



```
PS C:\PowerShellBackupProject> C:\PowerShellBackupProject\BackupFirewall.ps1
Cannot create a file when that file already exists.


Backup successful! File saved at C:\PowerShellBackupProject\FirewallRulesBackup.wfw
```

Test Case 2: Registry Backup

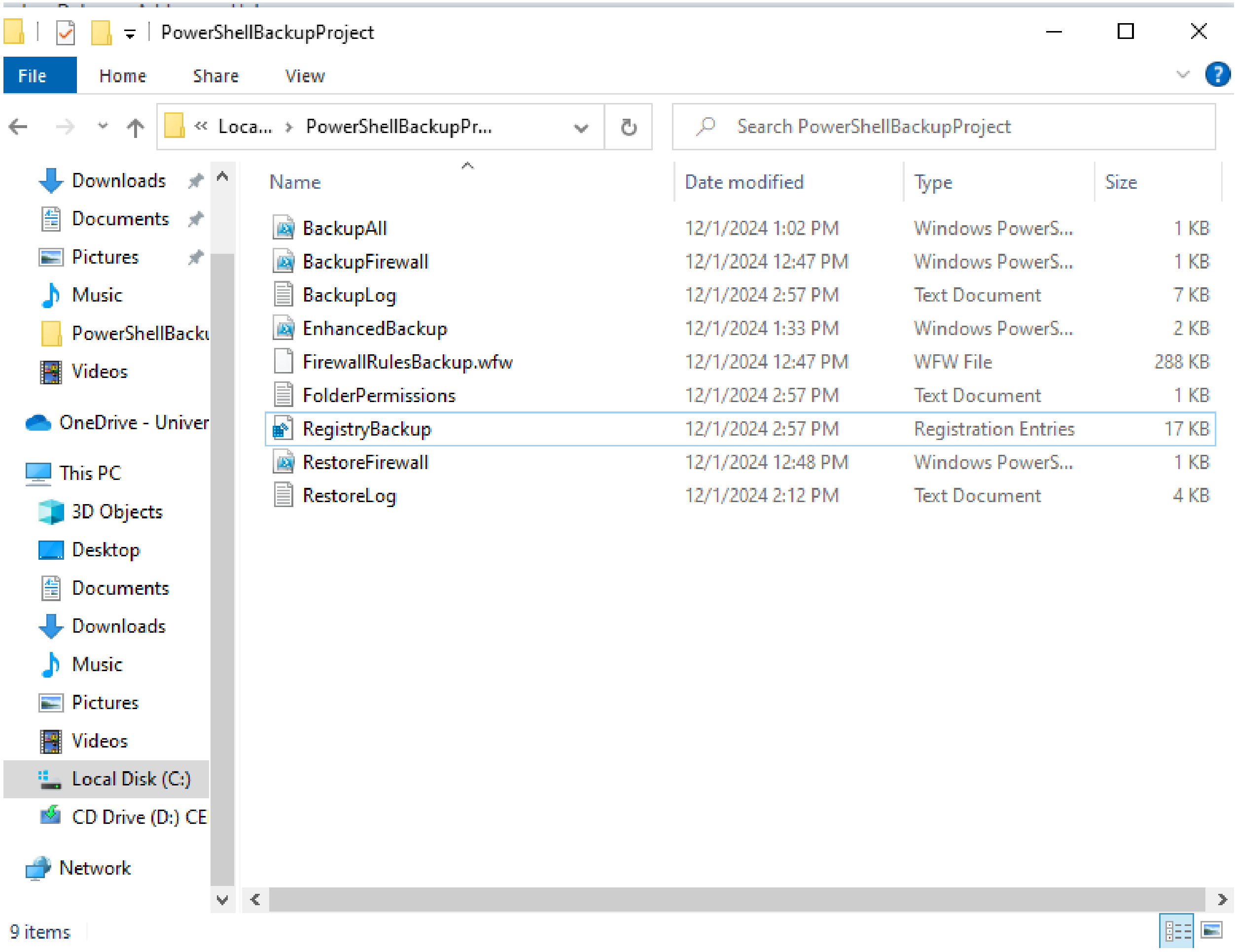
Script Used: BackupAll.ps1

Steps:

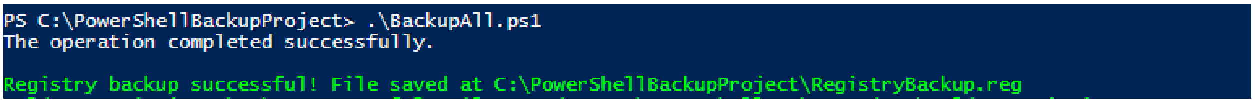
1. Run the script.

2. Verify that RegistryBackup.reg exists in the project folder.

Expected Outcome: Valid .reg file is created.



Name	Date modified	Type	Size
BackupAll	12/1/2024 1:02 PM	Windows PowerS...	1 KB
BackupFirewall	12/1/2024 12:47 PM	Windows PowerS...	1 KB
BackupLog	12/1/2024 2:57 PM	Text Document	7 KB
EnhancedBackup	12/1/2024 1:33 PM	Windows PowerS...	2 KB
FirewallRulesBackup.wfw	12/1/2024 12:47 PM	WFW File	288 KB
FolderPermissions	12/1/2024 2:57 PM	Text Document	1 KB
RegistryBackup	12/1/2024 2:57 PM	Registration Entries	17 KB
RestoreFirewall	12/1/2024 12:48 PM	Windows PowerS...	1 KB
RestoreLog	12/1/2024 2:12 PM	Text Document	4 KB



```
PS C:\PowerShellBackupProject> .\BackupAll.ps1
The operation completed successfully.

Registry backup successful! File saved at C:\PowerShellBackupProject\RegistryBackup.reg
```

Conclusion

This project demonstrated automating the backup of critical system configurations using PowerShell. The finished scripts ensure firewall rules registry keys, and folder permissions are securely backed up. With enhanced error handling and logging, the EnhancedBackup.ps1 script provides a robust and reusable solution. Through this project, I gained deeper insights into PowerShell’s capabilities and its practical applications in system administration.