Abstract
This technical report provides insight into automating the Windows File Services environment by using the NetApp® PowerShell Toolkit. It discusses the provisioning, modifying, monitoring, and cleaning of the CIFS environment, native auditing, the antivirus environment, and the NetApp FPolicy® component. It also discusses how to manage day-to-day CIFS tasks.
TABLE OF CONTENTS

1 Overview........................................................................................................................................... 4
  1.1 Purpose and Scope .......................................................................................................................... 4
  1.2 Intended Audience .......................................................................................................................... 4

2 PowerShell and NetApp PowerShell Toolkit ONTAP Module ......................................................... 4
  2.1 Windows PowerShell Overview ...................................................................................................... 4
  2.2 NetApp PowerShell Toolkit Overview ............................................................................................. 5
  2.3 Getting Started ............................................................................................................................... 6
  2.4 Help System .................................................................................................................................... 7
  2.5 PowerShell Pipeline ........................................................................................................................ 7

3 Components of Windows File Services ........................................................................................... 9

4 Managing the CIFS Environment ........................................................................................................ 10
  4.1 CIFS Server Prerequisites ............................................................................................................... 10
  4.2 Provisioning the CIFS Environment ................................................................................................. 13
  4.3 Modifying the CIFS Environment ................................................................................................... 17
  4.4 Monitoring the CIFS Environment .................................................................................................. 23
  4.5 Cleaning the CIFS Environment ..................................................................................................... 24

5 Managing Native Auditing .................................................................................................................. 27

6 Managing the Antivirus Environment .................................................................................................. 29

7 Managing FPolicy ................................................................................................................................ 35

8 Managing Day-to-Day Tasks .................................................................................................................. 38

References............................................................................................................................................... 40

LIST OF TABLES
Table 1) Versions of PowerShell. ............................................................................................................. 5
Table 2) Cmdlets for CIFS server prerequisites. .......................................................................................... 12
Table 3) Cmdlets for CIFS server provisioning. .......................................................................................... 13
Table 4) Cmdlets for managing the CIFS server ....................................................................................... 17
Table 5) Cmdlets for monitoring the CIFS server. ....................................................................................... 23
Table 6) Cmdlets for cleaning the CIFS server ......................................................................................... 24
Table 7) Cmdlets for managing native auditing. ........................................................................................ 28
Table 8) Cmdlets for managing the antivirus environment. ...................................................................... 31
Table 9) Cmdlets for managing FPolicy .................................................................................................. 35
LIST OF FIGURES

Figure 1) PowerShell pipeline.................................................................8
Figure 2) PowerShell pipeline by value. ...................................................8
Figure 3) PowerShell pipeline by property name.......................................8
Figure 4) Components of Windows File Services......................................10
Figure 5) CIFS server prerequisites. ......................................................11
Figure 6) Provisioning the CIFS environment..........................................13
Figure 7) Managing native auditing. ......................................................28
Figure 8) Managing the antivirus environment. .......................................30
Figure 9) Managing FPolicy...................................................................35
1 Overview

Microsoft is moving toward a more command-line-based interface to manage Windows servers. This trend started with the Server Core option of Windows Server and is accelerating with the Windows Nano Server. The CLI-first approach does not preclude a user from managing an environment through a GUI; however, the GUI often lacks all the options, flexibility, and power that are offered in the CLI environment. With this limitation in mind, NetApp PowerShell Toolkit (PSTK) was created. PSTK exposes all the functions and features of the NetApp controllers by using the NetApp Manageability SDK open interface, including those features that are not available in our GUI management systems. Windows File Services can be automated and managed by using NetApp PSTK.

1.1 Purpose and Scope

This document provides a brief overview of automating a Windows File Services environment by using the NetIFS PowerShell Toolkit. It discusses the provisioning, modifying, monitoring, and cleaning of the CIFS environment, native auditing, the antivirus environment, and the FPolicy component. It also discusses how to manage day-to-day CIFS tasks.

1.2 Intended Audience

This document is intended for system and storage administrators who want to manage and automate the Windows File Services environment with Windows PowerShell. Experts in PowerShell will find that all the NetApp PSTK commands are self-documented, cross-referenced, and object-based. Therefore, PowerShell experts will immediately be able to produce powerful and robust scripts. This document, however, is for those administrators who are less familiar with PowerShell and its advantages when used to manage an infrastructure.

It assumes that the reader:

- Has a general knowledge about NetApp hardware and software solutions. For details, review the System Administration Guide for Cluster Administrators.
- Has a general knowledge about the file-access protocols SMB and CIFS. For CIFS- and SMB-related information, review the Best Practices Guide for Windows File Services and the CIFS/SMB Configuration Express Guide.
- Has a general knowledge about Windows PowerShell.

2 PowerShell and NetApp PowerShell Toolkit ONTAP Module

2.1 Windows PowerShell Overview

Windows PowerShell is a task automation and configuration management framework from Microsoft. It consists of a command-line shell and a scripting language that are built on the .NET framework. PowerShell enables administrators to perform administrative tasks and to automate tasks on both local and remote Windows systems. Following are definitions of a few PowerShell components:

- **Cmdlet.** A specialized .NET class that implements a particular operation.
- **Script.** A plain text file with a .ps1 file extension that contains one or more PowerShell cmdlets.
- **Function.** A named block of code that can be called once or multiple times anywhere from within a script.
- **Advanced function.** A function written in script that performs operations similar to the way that cmdlets do.
- **Module.** A package that contains Windows PowerShell commands such as cmdlets, providers, functions, workflows, variables, and aliases.
Table 1 summarizes the various PowerShell versions.

Table 1) Versions of PowerShell.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PowerShell 1.0</td>
<td>2006</td>
<td>Windows Server 2003</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>PowerShell 2.0</td>
<td>2009</td>
<td>Windows 7, Windows Server 2008 R2</td>
<td>Windows XP SP3; Windows Server 2003 SP2; Windows Vista SP1, Vista SP2; Windows Server 2008 SP1, 2008 SP2</td>
<td>3.5+</td>
</tr>
<tr>
<td>PowerShell 3.0</td>
<td>2012</td>
<td>Windows 8, Windows Server 2012</td>
<td>Windows 7 SP1; Windows Server 2008 SP2, 2008 R2 SP1</td>
<td>4.0+</td>
</tr>
<tr>
<td>PowerShell 5.0</td>
<td>2014</td>
<td>Windows 10</td>
<td>Windows 7 SP1, Windows 8.1, Windows Server 2012 R2</td>
<td>4.6+</td>
</tr>
</tbody>
</table>

2.2 NetApp PowerShell Toolkit Overview

NetApp PowerShell Toolkit (PSTK) is a PowerShell module that provides end-to-end automation and enables storage administration of NetApp storage controllers. PSTK is a unified package that contains the PowerShell modules for NetApp ONTAP® and SANtricity® software. The ONTAP module contains over 2,000 cmdlets and helps administer NetApp FAS, NetApp All Flash FAS, commodity hardware, and the cloud. The SANtricity module contains over 250 cmdlets and helps administer the NetApp E-Series and EF-Series all-flash arrays.

The toolkit also contains several cmdlets that are aimed at storage administration on the Windows host. These operations include creating virtual disks, resizing virtual disks, reclaiming space in virtual disks, copying files, deleting files, reclaiming space on host volumes, and much more.

Further Reading

- For more information about the NetApp PowerShell Toolkit, visit the [NetApp Support site](https://www.netapp.com) and see [Getting Started with PSTK](https://www.netapp.com) and [Making the Most of PSTK](https://www.netapp.com).
- For the first-steps guide about NetApp PowerShell Toolkit, see the [ONTAP PowerShell Toolkit Primer](https://www.netapp.com).
• For support on NetApp PowerShell Toolkit, visit ONTAP Community Discussions and SANtricity Community Discussions.

System Requirements
Following are requirements for the NetApp PowerShell Toolkit:

• Microsoft Windows 7 or later
• Windows Server 2008 or later
• Microsoft Windows PowerShell v3.0 or later
• Microsoft .NET Framework v4.5.2 or later
• NetApp ONTAP 7.3.x or later

2.3 Getting Started
1. Download the NetApp PowerShell Toolkit .msi installer from the NetApp Support site.
2. Install the NetApp PowerShell Toolkit (make sure that Data ONTAP PowerShell Toolkit is selected for installation under Advanced Setup).
3. Open the PowerShell Console and import the ONTAP module by running the following cmdlet:

   ```
   # Import the ONTAP module
   Import-Module -Name DataONTAP

   # Change the execution policy to trust the scripts to be run
   Set-ExecutionPolicy -ExecutionPolicy RemoteSigned

   # Verify the execution policy
   Get-ExecutionPolicy

   # Connect to a NetApp Controller
   Connect-NcController -Name <name or IP address of controller> -Credential (Get-Credential)
   ```

4. The following window prompts you to enter the credentials for the controller. Enter the credentials to connect the controller.

   ![Windows PowerShell credential request](image-url)
If you want to avoid the prompt for credentials, use credential object as follows:

```powershell
# Controller info
$name = <name or IP address of controller>
$username = <user name of controller>
$password = <password for controller>

# Create controller credential object
$credpassword = ConvertTo-SecureString -AsPlainText -Force $password
$credential = New-Object management.automation.pscredential $username, $credpassword

# Connect to a NetApp Controller
Connect-NcController -Name $name -Credential $credential
```

**Note:** The cmdlet `Connect-NcController` is to connect to an ONTAP controller (previously a NetApp clustered Data ONTAP® controller). To connect to a Data ONTAP operating in 7-Mode controller, use `Connect-NaController`. The prefix `Nc` is for an ONTAP controller, and the prefix `Na` is for a 7-Mode controller. All cmdlets follow this prefix rule.

### 2.4 Help System

The PowerShell Toolkit ONTAP module includes extensive, console-based help with elaborate descriptions and examples. For help, use the following cmdlets:

```powershell
# View the full HTML Help
Show-NcHelp

# view all the available PSTK cmdlets
Get-NcHelp

# view all the available PSTK cmdlets
Get-NcCommand

# view the available categories in PSTK
Get-NcHelp -CategoryList

# view all the cmdlets of a particular category
Get-NcCommand -Category <category>

# view help for a particular cmdlet
Get-Help <cmdlet_name>

# view full help for a particular cmdlet
Get-Help <cmdlet_name> -full

# view example for a particular cmdlet
Get-Help <cmdlet_name> -examples

# Pops up a window with the help information (PS v3.0+)
Get-Help <cmdlet_name> -ShowWindow

# get the members (properties and methods) of the objects returned by a cmdlet
<cmdlet_name> | Get-Member
```

### 2.5 PowerShell Pipeline

A **pipeline** is a series of commands that are connected by the pipeline operator `|`. Each pipeline operator sends the results of the preceding command to the next command (Figure 1).
In PowerShell, a cmdlet can receive objects from the pipeline in two ways:

- **By value.** The output of a cmdlet must be the same type as the `-InputObject` parameter of the other cmdlet, as shown in Figure 2.

- **By property name.** When the object has a property that matches the name of a parameter in the other cmdlet, as shown in Figure 3.

To determine the parameters that accept pipeline input, use `Get-Help`.
-InputObject <ServiceController[]>
Specifies ServiceController objects representing the services to be retrieved. Enter a variable that contains the objects, or type a command or expression that gets the objects. You can also pipe a service object to Get-Service.

Required? false
Position? named
Default value
Accept pipeline input? true (ByValue)
Accept wildcard characters? False

-Name <String[]>
Specifies the service names of services to be retrieved. Wildcards are permitted. By default, Get-Service gets all of the services on the computer.

Required? false
Position? 1
Default value All services
Accept pipeline input? true (ByPropertyName, ByValue)
Accept wildcard characters? true

Examples

# Basic Pipeline example
Get-NcVol -Name <volume name> | Set-NcVol -Offline

# The objects returned from the Get-NcVol cmdlet include a ‘Name’ property
# The Set-NcVol cmdlet accepts pipeline input via the ‘Name’ property (ByPropertyName)

# Basic Filtering example (Filtering based on search criteria)
Get-NcVol | Where-Object { $_.State -eq ‘Offline’ } | Set-NcVol -Online

# Where-Object filters objects passed down the pipeline based on the search criteria you specify
# Basic Filtering example (Filtering based on property)
Get-NcVol | Select-Object -Property Name, TotalSize

# Select-Object allows you to filter the results based on property names, number of objects, or other criteria

3 Components of Windows File Services

Figure 4 shows the components of Windows File Services. A CIFS server must be configured over a NetApp ONTAP storage virtual machine (SVM) over which SMB shares are configured so that SMB clients can access files. After the CIFS server has been set up, several management tasks can be performed, such as:

- Configuring CIFS options
- Managing CIFS server security settings
- Configuring SMB protocol and SMB signing
- Managing CIFS oplocks
- Configuring IPv6 CIFS access
- Applying Group Policy Objects (GPOs) to CIFS servers
- Managing domain controller connections
- Managing CIFS server service
- Managing the `Home` directory
- Managing NetApp FPolicy policies
- Managing antivirus (AV)
- Managing native auditing

Figure 4) Components of Windows File Services.

4 Managing the CIFS Environment

This section provides information about CIFS server prerequisites and the cmdlets that are used for those prerequisites, and it gives examples of the cmdlets.

4.1 CIFS Server Prerequisites

Figure 5 shows the prerequisites for deploying a CIFS server.
Table 2 shows the cmdlets that are used to meet the prerequisites. CIFS licensing, time services, and networking prerequisites must be met before configuring the CIFS server.

Figure 5) CIFS server prerequisites.
### Table 2) Cmdlets for CIFS server prerequisites.

<table>
<thead>
<tr>
<th>Description</th>
<th>Cmdlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a new aggregate</td>
<td>New-NcAggr</td>
</tr>
<tr>
<td>Create a new SVM (called Vserver in the CLI)</td>
<td>New-NcVserver</td>
</tr>
<tr>
<td>Configure network settings: Create a new subnet</td>
<td>New-NcNetSubnet</td>
</tr>
<tr>
<td>Configure network settings: Create a new LIF</td>
<td>New-NcNetInterface</td>
</tr>
<tr>
<td>Configure DNS settings: Configure a DNS</td>
<td>New-NcNetDns</td>
</tr>
<tr>
<td>Configure DNS settings: Enable DDNS</td>
<td>Set-NcNetDdns</td>
</tr>
<tr>
<td>Configure the NTP service</td>
<td>New-NcNtpServer</td>
</tr>
</tbody>
</table>

```bash
# User Inputs
# The below script uses example inputs. Please change it according to your needs

# Aggregate info
$aggr_name = "aggr_cifs"
$aggr_diskcount = 5

# VServer info
$vserver_name = "vserver_cifs"
$rootvolume_name = $vserver_name+"_root"

# Network info
$subnet_name = "cifs_subnet_231"
$lif1_name = "cifs_lif1"

# Domain info
$domain = "netapp.local"

# NTP info
$ntp_server_ip = "10.195.49.25"

# Get the nodes in the cluster
$cluster_nodes = (Get-NcNode).Node

# Get the data ports in cluster
$data_ports = Get-NcNetPort | where {$_._Role -eq 'data' -and $_._LinkStatus -eq 'up'}

# Create a new Aggregate
New-NcAggr -Name $aggr_name -Node $cluster_nodes[1] -DiskCount $aggr_diskcount

# Create a new VServer
New-NcVserver -Name $vserver_name -RootVolumeAggregate $aggr_name -RootVolume $rootvolume_name -RootVolumeSecurityStyle NTFS

# Possible values for RootVolumeSecurityStyle: 'unix', 'ntfs', 'mixed'

# Network Configuration
New-NcNetSubnet -Name $subnet_name -BroadcastDomain Default -Subnet 10.238.231.0/22 -Ipspace Default -Gateway 10.238.231.1 -IpRange 10.238.231.20-10.238.231.100 -ForceUpdateLifAssociations

# Above cmdlet uses example subnet, ip address, gateway and ip range
# Please change it according to your needs
```
New-NcNetInterface -Name $lif1_name -Vserver $vserver_name -Role data -Node $cluster_nodes[0] -Port $data_ports[0].Port -DataProtocols cifs -FirewallPolicy data -Subnet $subnet_name
# Providing subnet will take IP address automatically
# To manually specify the IP, use -Address and -NetMask argument
# Possible values for FirewallPolicy: 'mgmt', 'data', 'cluster', 'intercluster'

# DNS Configuration
# Configure DNS settings for VServer
New-NcNetDns -Domains $domain -NameServers 10.238.231.199 -VserverContext $vserver_name
# Above cmdlet uses example NameServer IP Address
# Please change it according to your needs
# Enable DDNS for VServer
Set-NcNetDdns -VserverContext $vserver_name -DomainName $domain -Enable $true

# NTP Server Configuration
New-NcNtpServer -ServerName $ntp_server_ip -IsPreferred

4.2 Provisioning the CIFS Environment

This section provides detailed information about the cmdlets that are used to provision the CIFS environment. Figure 6 outlines the process, and Table 3 presents the provisioning cmdlets.

Figure 6) Provisioning the CIFS environment.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Cmdlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFS server</td>
<td>Create a new CIFS server</td>
<td>Add NcCifsServer</td>
</tr>
<tr>
<td>CIFS local users and groups</td>
<td>Create a new CIFS local user</td>
<td>New-NcCifsLocalUser</td>
</tr>
<tr>
<td>CIFS local users and groups</td>
<td>Create a new CIFS local group</td>
<td>New-NcCifsLocalGroup</td>
</tr>
<tr>
<td>CIFS local users and groups</td>
<td>Add a CIFS local user to a CIFS local group</td>
<td>Add-NcCifsLocalGroupMember</td>
</tr>
<tr>
<td>CIFS share and permissions</td>
<td>Create a new CIFS share</td>
<td>Add-NcCifsShare</td>
</tr>
<tr>
<td>CIFS share and permissions</td>
<td>Set CIFS share permissions</td>
<td>Add-NcCifsShareAcl</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Create a new NTFS security descriptor</td>
<td>New-NcFileDirectorySecurityNtfs</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Add a discretionary access control list (ACL) to an NTFS security descriptor</td>
<td>Add-NcFileDirectorySecurityNtfsDacl</td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
<td>Cmdlet</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Add a system/audit ACL to an NTFS security descriptor</td>
<td>Add-NcFileDirectorySecurityNtfsSacl</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Create a new file-directory security policy</td>
<td>New-NcFileDirectorySecurityPolicy</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Attach a file-directory security policy task to a path</td>
<td>Add-NcFileDirectorySecurityPolicyTask</td>
</tr>
<tr>
<td>Domain activities</td>
<td>Add a trusted domain</td>
<td>Add-NcCifsTrustedDomain</td>
</tr>
<tr>
<td>Domain activities</td>
<td>Configure the preferred domain controllers</td>
<td>Add-NcCifsPreferredDomainController</td>
</tr>
<tr>
<td>Home directory</td>
<td>Add a CIFS home directory search path</td>
<td>Add-NcCifsHomeDirectorySearchPath</td>
</tr>
<tr>
<td>Shadow copy</td>
<td>Add files to a shadow copy in a share</td>
<td>Add-NcCifsShadowCopyFile</td>
</tr>
</tbody>
</table>

# User Inputs
# The below script uses example inputs. Please change it according to your needs

# CIFS info
$cifsserver_name = "cifs_server"
$vol_name = "cifs_vol"
$qtree_name = "cifs_qtree"
$vol_share_name = "cifs_share_vol"
$qtree_share_name = "cifs_share_qtree"

# Domain info
$domain = "netapp.local"
$domain_username = "Administrator"
$domain_password = "password"

# VServer info
$vservename = "vserv_cifs"

# CIFS Local Users and Group info
$localgroup_name = "cifs_localgroup"
$localuser_name = "cifs_localuser"
$localuser_password = "password"

# File/Folder Permissions
$ntfs_sec_desc_id = "ntfssd"
$policy_name = "policy1"

# Home Directory info
$homedi_volume_name = "home_vol"
$homeshare_name = "home"

# Create a Secure String password for local user
$secure_localuser_password = ConvertTo-SecureString -AsPlainText -Force $localuser_password

# List all cmdlets available for CIFS
Get-NcHelp | where {$_._Category -eq 'cifs'}

# Create a new CIFS server
Add-NcCifsServer -Name $cifsserver_name -Domain $domain -OrganizationalUnit CN=Computers -AdminUsername $domain_username -AdminPassword $domain_password -VserverContext $vservename
# CIFS Local Users and Groups
# Create a new CIFS Local User
New-NcCifsLocalUser -UserName $localuser_name -Password $secure_localuser_password -FullName "$localuser_name $localuser_name" -Description "new cifs local user" -VserverContext $vserver_name
# Above cmdlet uses example Name, FullName and Description
# Please change it according to your needs

# Create a new CIFS Local Group
New-NcCifsLocalGroup -Name $localgroup_name -Description "local group for cifs" -VserverContext $vserver_name
# Above cmdlet uses example GroupName and Description
# Please change it according to your needs

# Add a CIFS Local user to CIFS local group
Add-NcCifsLocalGroupMember -Name $localgroup_name -Member $localuser_name -VserverContext $vserver_name

#-------------------------------------
# CIFS Share Configuration
# Create a new Volume to share
New-NcVol -Name $vol_name -Aggregate $aggr_name -size 10g -JunctionPath "/$vol_name" -SecurityStyle "ntfs" -VserverContext $vserver_name
# Create a new Qtree to share
New-NcQtree -Volume $vol_name -Qtree $qtree_name -VserverContext $vserver_name -SecurityStyle "ntfs"
# Share the volume
Add-NcCifsShare -Name $vol_share_name -Path "/$vol_name" -VserverContext $vserver_name -ShareProperties "browsable"

# Share the qtree
Add-NcCifsShare -Name $qtree_share_name -Path "/$vol_name/$qtree_name" -VserverContext $vserver_name -ShareProperties "browsable"

# CIFS Share Permissions
# Add permission for user or group for a CIFS Share
# Give full control access to the local group on volume share
Add-NcCifsShareAcl -Share $vol_share_name -UserOrGroup $localgroup_name -Permission "full_control" -UserGroupType "windows" -VserverContext $vserver_name
# Give read access to the local user on qtree share
Add-NcCifsShareAcl -Share $qtree_share_name -UserOrGroup $localuser_name -Permission "read" -UserGroupType "windows" -VserverContext $vserver_name
# Possible values for Permission: 'no_access', 'read', 'change', 'full_control'
# Possible values for UserGroupType: 'windows', 'unix_user', 'unix_group'

#-------------------------------------
# File/Folder Permissions
# Create a new NTFS security descriptor
New-NcFileDirectorySecurityNtfs -SecurityDescriptor $ntfs_sec_desc_id -Owner $localuser_name -Group $localgroup_name -VserverContext $vserver_name

# Add a discretionary access control entry to NTFS security descriptor
Add-NcFileDirectorySecurityNtfsDacl -SecurityDescriptor $ntfs_sec_desc_id -Account $localuser_name -AccessType "allow" -Rights "full_control" -ApplyTo "files" -VserverContext $vserver_name

# Add a system/audit access control entry to NTFS security descriptor
Add-NcFileDirectorySecurityNtfsSacl -SecurityDescriptor $ntfs_sec_desc_id -Account $localuser_name -AccessType "success" -Rights "read_and_execute" -AdvancedRights "full_control" -ApplyTo "sub_folders" -VserverContext $vserver_name
# Possible values for -AccessType: "deny", "allow"
# Possible values for -Rights: "no_access", "full_control", "modify", "read_and_execute", "read", "write"
# Possible values for -ApplyTo: "this_folder", "sub_folders", "files"

# Create a new file directory security policy
New-NcFileDirectorySecurityPolicy -Name $policy_name -VserverContext $vserver_name

    # Get the junction path of the file/folder
    $query = Get-NcVol -Template
    $query.Name = $vol_name
    $query.JunctionPath

    $volume_junctionpath = (Get-NcVol-Query $query | select JunctionPath).JunctionPath
    $qtree_junctionpath = "$volume_junctionpath/$qtree_name"

# Add a file security policy task
Add-NcFileDirectorySecurityPolicyTask -Name $policy_name -Path $qtree_junctionpath -SecurityType "ntfs" -NtfsSecurityDescriptor $ntfs_sec_desc_id -VserverContext $vserver_name

# Domain Activities
# Add to the list of trusted domains for name-mapping search
Add-NcCifsTrustedDomain -TrustedDomain $domain -VserverContext $vserver_name

# Add to a list of preferred domain controllers
Add-NcCifsPreferredDomainController -Domain $domain -DomainControllers 10.10.10.10 -VserverContext $vserver_name

# Above cmdlet uses example DomainControllers IP Address
# Please change it according to your needs

# Home Directory
# Create a new volume for Home Directory
New-NcVol -Name $homedir_volume_name -Aggregate $aggr_name -size 10g -JunctionPath "/$homedir_volume_name" -SecurityStyle "ntfs" -VserverContext $vserver_name

# Create a share for the volume
Add-NcCifsShare -Name $homeshare_name -Path "/$homedir_volume_name" -VserverContext $vserver_name

# Add that share as CIFS Home directory search path
Add-NcCifsHomeDirectorySearchPath -Path "/$homedir_volume_name" -VserverContext $vserver_name

# Add Home directory share
Add-NcCifsShare -Name %w -Path %w -VserverContext $vserver_name -ShareProperties "homedirectory"

# Get the VServer LIF
$vserver_lif = (Get-NcNetInterface -Name $lif1_name -VserverContext $vserver_name).Address

# Get all the AD Users by running a remote powershell command on the AD machine
$ad_userlist = Invoke-Command -ComputerName "ad.netapp.local" -ScriptBlock {Get-ADGroupMember -Identity CifsADgroup).name }
# Above cmdlet uses example ComputerName and example Identity
# Please change it according to your needs
# Please use FQDN of the AD machine for ComputerName
# Please provide a valid AD group for Identity

# Create folders for each user in the AD group
for($i=0; $i -lt $ad_userlist.Count; $i++)
{
    $ad_user_dir = $ad_userlist[$i]
    New-Item -Path "\$vserver_lif\$homeshare_name\$ad_user_dir" -ItemType directory
}

# Shadow Copy
# Add files in a share
New-Item -Path "\$vserver_lif\$qtree_share_name\file1.txt" -ItemType file
New-Item -Path "\$vserver_lif\$qtree_share_name\file2.txt" -ItemType file

# Add a list of files to shadow copy in a particular share
Add-NcCifsShadowCopyFile -Id 101 -File "\$qtree_share_name\file1.txt" -VserverContext $vserver_name
4.3 Modifying the CIFS Environment

Table 4 and the following example code provide detailed information about the cmdlets that are used to modify and manage the CIFS server.

Table 4) Cmdlets for managing the CIFS server.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Cmdlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFS server</td>
<td>Start the CIFS server</td>
<td>Start-NcCifsServer</td>
</tr>
<tr>
<td>CIFS server</td>
<td>Stop the CIFS server</td>
<td>Stop-NcCifsServer</td>
</tr>
<tr>
<td>CIFS server</td>
<td>Modify the CIFS server</td>
<td>Set-NcCifsServer</td>
</tr>
<tr>
<td>CIFS server</td>
<td>Set CIFS security tunable parameters</td>
<td>Set-NcCifsSecurity</td>
</tr>
<tr>
<td>CIFS server</td>
<td>Modify the CIFS specific tunable parameters</td>
<td>Set-NcCifsOption</td>
</tr>
<tr>
<td>CIFS local users and groups</td>
<td>Modify a CIFS local user</td>
<td>Set-NcCifsLocalUser</td>
</tr>
<tr>
<td>CIFS local users and groups</td>
<td>Rename a CIFS local user</td>
<td>Rename-NcCifsLocalUser</td>
</tr>
<tr>
<td>CIFS local users and groups</td>
<td>Modify a CIFS local group</td>
<td>Set-NcCifsLocalGroup</td>
</tr>
<tr>
<td>CIFS local users and groups</td>
<td>Rename a CIFS local group</td>
<td>Rename-NcCifsLocalGroup</td>
</tr>
<tr>
<td>CIFS share and permissions</td>
<td>Modify a CIFS share</td>
<td>Set-NcCifsShare</td>
</tr>
<tr>
<td>CIFS share and permissions</td>
<td>Modify CIFS share permissions</td>
<td>Set-NcCifsShareAcl</td>
</tr>
<tr>
<td>CIFS sessions and files</td>
<td>Retrieve the list of the established CIFS sessions</td>
<td>Get-NcCifsSession</td>
</tr>
<tr>
<td>CIFS sessions and files</td>
<td>Retrieve the list of the opened CIFS files</td>
<td>Get-NcCifsSessionFile</td>
</tr>
<tr>
<td>CIFS sessions and files</td>
<td>Close an open CIFS session</td>
<td>Close-NcCifsSession</td>
</tr>
<tr>
<td>CIFS sessions and files</td>
<td>Close an open CIFS file</td>
<td>Close-NcCifsSessionFile</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Modify the NTFS security descriptor</td>
<td>Set-NcFileDirectorySecurityNtfs</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Modify a discretionary ACL of the NTFS security descriptor</td>
<td>Set-NcFileDirectorySecurityNtfsDacl</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Modify a system/audit ACL of the NTFS security descriptor</td>
<td>Set-NcFileDirectorySecurityNtfsSacl</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Modify a file security policy task</td>
<td>Set-NcFileDirectorySecurityPolicyTask</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Apply the security settings of a policy</td>
<td>Set-NcFileDirectorySecurity</td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
<td>Cmdlet</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Domain activities</td>
<td>Enable the CIFS domain password schedule</td>
<td>Enable-NcCifsDomainPasswordSchedule</td>
</tr>
<tr>
<td>Domain activities</td>
<td>Modify the attributes of the CIFS domain password schedule object</td>
<td>Set-NcCifsDomainPasswordSchedule</td>
</tr>
<tr>
<td>Domain activities</td>
<td>Modify the list of trusted domains</td>
<td>Set-NcCifsTrustedDomain</td>
</tr>
<tr>
<td>Domain activities</td>
<td>Disable the CIFS domain password schedule</td>
<td>Disable-NcCifsDomainPasswordSchedule</td>
</tr>
<tr>
<td>Home directory</td>
<td>Modify the CIFS home directory search path</td>
<td>Set-NcCifsHomeDirectorySearchPath</td>
</tr>
<tr>
<td>Home directory</td>
<td>Modify the CIFS home directory configurations</td>
<td>Set-NcCifsHomeDirectoryConfig</td>
</tr>
<tr>
<td>Shadow copy</td>
<td>Save the NetApp Snapshot® copies of the shadow copy set</td>
<td>Save-NcCifsShadowCopySnapshot</td>
</tr>
<tr>
<td>Shadow copy</td>
<td>Restore a shadow copy directory</td>
<td>Restore-NcCifsShadowCopyDirectory</td>
</tr>
</tbody>
</table>

```
# Start a CIFS server on the specified Vserver
Start-NcCifsServer -VserverContext $vserver_name

# Stop a CIFS server on the specified Vserver
Stop-NcCifsServer -VserverContext $vserver_name

# Set the configurations of a CIFS server
Set-NcCifsServer -CifsServer $cifsserver_name -Domain $domain -OU CN=Computers -AdminUsername $domain_username -AdminPassword $domain_password -VserverContext $vserver_name -AdministrativeStatus "up" -ForceAccountOverwrite

# Possible values for AdministrativeStatus: 'up', 'down'

# Set CIFS security tunable parameters
Set-NcCifsSecurity -ClockSkew 5 -TicketAge 10 -RenewAge 7 -IsSigningRequired $true -IsPasswordComplexityRequired $true -UseStartTlsForAdLdap $true -IsAesEncryptionEnabled $true -LmCompatibilityLevel 'krb' -IsSmbEncryptionRequired $true -KerberosKdcTimeout 30s -VserverContext $vserver_name

# -ClockSkew : The clock skew in minutes is the tolerance for accepting tickets with time stamps that do not exactly match the host's system clock
# -TicketAge : This option determines the maximum amount of time in hours that a user's ticket may be used for the purpose of Kerberos authentication
# -RenewAge : This option determines the maximum amount of time in days for which a ticket can be renewed
# -IsSigningRequired : If true, signing is required for incoming CIFS traffic
# -IsPasswordComplexityRequired : If true, password complexity is required for local users
# -UseStartTlsForAdLdap : If true, use start-tls for AD LDAP connections. Default value : false
# -IsAesEncryptionEnabled : Determines whether AES-128 and AES-256 encryption mechanisms are enabled for Kerberos-related CIFS communication.
# -Default value : false
# -LmCompatibilityLevel : This option determines the LM compatibility level. Default value : 'LM, NTLM, NTLMV2 and Kerberos'
```
# Modify the CIFS specific tunables that can be set on a Vserver

# DefaultUnixUser : This is the default UNIX user mapping that will be used if the identity of a CIFS user cannot be mapped using normal name mapping rules.
# -ReadGrantsExecute : On a file with UNIX Style security effective on it, if the file has read permission on it, a CIFS user would be allowed to execute permissions if this option is enabled. Possible values: \"enabled\", \"disabled\".
# -WinsServers : List of Windows Internet Name Service (WINS) IP addresses. The Vserver will send NetBIOS name resolution requests to these addresses.
# -EnableSmb2 : Specify to enable Smb2 on this CIFS server.
# -DisableSmb2 : Specify to disable Smb2 on this CIFS server.
# -MaxMpx : This option controls maximum simultaneous operations the CIFS server reports it can process per TCP connection. The default value for this option is 255.
# -ShadowCopyDirDepth : The maximum depth of directories to shadow copy.
# -IsSmb3Enabled : If true, the CIFS server negotiates the SMB3 version of the CIFS protocol.
# -IsCopyOffloadEnabled : If true, the CIFS server is capable of performing copy offload operation.
# -DefaultUnixGroup : The default UNIX group used if the identity of a CIFS group cannot be mapped using normal group mapping rules.
# -IsShadowCopyEnabled : If true, the CIFS server is capable of performing shadow copy operations.
# -IsReferralEnabled : If true, the CIFS server refers clients to more optimal data access paths (LIFs).
# -IsLocalAuthEnabled : If true, CIFS local users can authenticate.
# -IsLocalUsersAndGroupsEnabled : If true, the CIFS local users and groups feature is enabled on the cluster.
# -IsUseJunctionsAsReparsePointsEnabled : If true, the CIFS server exposes junction points as reparse points to Windows clients.
# -IsExportPolicyEnabled : If true, the CIFS server uses export policies to control client access.
# -IsTrustedDomainEnumSearchEnabled : If true, the CIFS server is capable of performing enumeration of trusted domains and search to map a UNIX user to a Windows user.
- FileSystemSectorSize : Specifies the size of file system sector in bytes reported to SMB clients.
  # Possible values:
  # 512' - Reported file system sector size to SMB clients is 512 bytes,
  # '4096' - Reported file system sector size to SMB clients is 4096 bytes

- IsCopyOffloadDirectCopyEnabled : If true, the direct-copy copy offload mechanism is enabled. The default value is true.

- IsUnixNxAclEnabled : If true, NT ACLs can be set on a volume with UNIX security-style.

- ClientSessionTimeout : The amount of idle time (in seconds) before a CIFS session is disconnected.

- IsAdvancedSparseFileSupportEnabled : Specifies whether advanced sparse file capabilities, such as Query Allocated Ranges and Set Zero Data, are enabled on the CIFS server.

- MaxFileWriteZeroLength : Maximum length of data that the CIFS server can write zero with one CIFS request. Value must be between 4K and 1G

- GuestUnixUser : The specific unix user to which a guest user coming from any untrusted domain can be mapped.

- Smb1MaxBufferSize : Maximum buffer size used for SMB1 message that the server can receive

- MaxOpenSameFilePerTree : Maximum existing opens on the same file per CIFS Tree.

- MaxConnectionsPerSession : Maximum number of connections allowed per Multichannel session.

- PathComponentCacheMaxEntries : This parameter controls the maximum number of cache entries in one instance of path component cache.

- PathComponentCacheMaxSessionTokenSize : This parameter specifies maximum session token size for path component cache for SMB2.

- PathComponentCacheSymlinkExpiration : This parameter controls the time in milliseconds when the path component cache entry that is symlink would be considered fresh.

- WinNameForNullUser : This parameter specifies a valid Windows user or group name that will be added to the CIFS credentials for a NULL user session.
# -IsNetbiosOverTcpEnabled : This optional parameter specifies whether the use of NetBIOS over TCP is enabled or not.

# CIFS Local Users and Groups
# Modify a CIFS local user
Set-NcCifsLocalUser -UserName $localuser_name -FullName $localuser_name -Description "new cifs local user" -VserverContext $vserver_name
# Use -Disable to disable this local user

# Modify the CIFS Local user password
Set-NcCifsLocalUser -UserName $localuser_name -Password $localuser_password -VserverContext $vserver_name

# Rename a CIFS local user
Rename-NcCifsLocalUser -UserName $localuser_name -NewUserName "new name" -VserverContext $vserver_name

# Modify a CIFS local group
Set-NcCifsLocalGroup -Name $localgroup_name -Description "local cifs users' group" -VserverContext $vserver_name

# Rename a CIFS local group
Rename-NcCifsLocalGroup -Name $localgroup_name -NewName $localgroup_name -VserverContext $vserver_name

# CIFS Shares and Permissions
# Modify CIFS Shares
Set-NcCifsShare -Name $qtree_share_name -ShareProperties "browsable" -VserverContext $vserver_name

# Modify CIFS Share Permissions
Set-NcCifsShareAcl -Share $qtree_share_name -UserOrGroup $localuser_name -Permission "write" -VserverContext $vserver_name

# CIFS Sessions and Files
# Retrieve the list of the established CIFS sessions
Get-NcCifsSession

# Retrieve the list of the opened CIFS files
Get-NcCifsSessionFile

# Close an open CIFS session
Close-NcCifsSession -Node $cluster_nodes[0] -VserverContext $vserver_name -Confirm:$false

# Close an open CIFS file
Close-NcCifsSessionFile -Node $cluster_nodes[0] -VserverContext $vserver_name -Confirm:$false

# File/Folder Permissions
# Modify the NTFS security descriptor
Set-NcFileDirectorySecurityNtfs -SecurityDescriptor $ntfs_sec_desc_id -Owner $localuser_name -Group $localgroup_name -ControlFlags 1

# Where -ControlFlags : The security descriptor control flags integer value
# The following are the bit fields of control flags
# 1... . .... .... = Self Relative
# 0... . .... .... = RM Control Valid
# .0... . .... .... = DACL Protected
# ...0 . .... .... = SACL Protected
# .... 0... .... .... = SACL Inherited
# .... .0... .... .... = DACL Inherited
# .... ..0... .... .... = DACL Inherited
# .... ...0 .... .... = DACL Inherit Required
# .... .... 0... .... = DACL Inherit Required
# .... .... ..0... = SACL Inherited
# .... .... ...0 = SACL Defaulted
# .... .... ...0 = SACL Inherit Required
# .... .... ..0. = SACL Defaulted
# .... .... ..0. = SACL Present
# .... .... ..0. = SACL Defaulted
# .... .... ..1. = DACL Present
# .... .... .0. = Group Defaulted
# .... .... ...0 = Owner Defaulted
# At present only the following flags are honored. Others are ignored.
# .0. .... .... .... = SACL Protected
# ...0 .... .... .... = DACL Protected
# .... .... ..0. .... = SACL Defaulted
# .... .... .... 0... = DACL Defaulted
# .... .... .... ..0. = Group Defaulted
# .... .... .... ...0 = Owner Defaulted

# Modify an discretionary access control entry of a file security descriptor
Set-NcFileDirectorySecurityNtfsDacl -SecurityDescriptor $ntfs_sec_desc_id -Account $localuser_name -AccessType "allow" -Rights "full_control" -AdvancedRights "full_control" -ApplyTo "this_folder" -VserverContext $vserver_name

# Modify an system/audit access control entry of a file security descriptor
Set-NcFileDirectorySecurityNtfsSacl -SecurityDescriptor $ntfs_sec_desc_id -Account $localuser_name -AccessType "success" -Rights "read_and_execute" -AdvancedRights "full_control" -ApplyTo "this_folder" -VserverContext $vserver_name

# Possible values for
- AccessType : "deny", "allow"
- Rights : "no_access", "full_control", "modify", "read_and_execute", "read", "write"

# Possible values for
- ApplyTo : "this_folder", "sub_folders", "files"

# Get the junction path of the file/folder
$qtree_junctionpath = (Get-NcVol -Query $query | select JunctionPath).JunctionPath

# Modify a file security policy task
Set-NcFileDirectorySecurityPolicyTask -Name $policy_name -Path $qtree_junctionpath -SecurityType "ntfs" -NtfsMode "replace" -NtfsSecurityDescriptor $ntfs_sec_desc_id -Index 2 -VserverContext $vserver_name

# Possible values for
- SecurityType : "ntfs", "nfsv4"
- NtfsMode : "propagate", "ignore", "replace"
- Index : the target index/position of this task in the policy. If a policy has already 5 tasks and this is the 6th task you are adding and you want to add this task as 2nd task, you can specify the index 2. By default the task gets added as last task. If the index number exceeds the number of positions, it will go at the end

# Apply security settings of a policy
Set-NcFileDirectorySecurity -Name $policy_name -VserverContext $vserver_name

# Domain Activities
# Enable the CIFS domain password schedule
Enable-NcCifsDomainPasswordSchedule -VserverContext $vserver_name

# Modify the attributes of CIFS domain password schedule object
Set-NcCifsDomainPasswordSchedule -WeeklyInterval 8 -RandomizedMinute 150 -DayOfWeek 1 -TimeOfDay 06:30:00 -VserverContext $vserver_name

# Where
- WeeklyInterval : The number of weeks after which the scheduled domain account password change must occur
- RandomizedMinute : The minutes within which the scheduled domain account password start time can be randomized beginning at TimeOfDay
- DayOfWeek : The day of week when the scheduled domain account password change occurs
- TimeOfDay : The time in HH:MM:SS format at which the scheduled domain account password change starts

# Modify the list of trusted domains for name-mapping search
Set-NcCifsTrustedDomain -TrustedDomain $trusted_domain -VserverContext $vserver_name

# Disable the CIFS domain password schedule
Disable-NcCifsDomainPasswordSchedule -VserverContext $vserver_name

# Set the position of a path in the list of paths that will be searched to find a CIFS user's home directory
Set-NcCifsHomeDirectorySearchPath -Path "/$homedir_volume_name" -Position 1 -VserverContext $vserver_name

# Above cmdlet moves the CIFS home directory search path /$homedir_volume_name into the first position

# Modify cifs home directory configurations

# Possible values for -IsAdminAccessEnabled : "true", "false". If true, home directory access is enabled to admin
# Possible values for -IsPublicAccessEnabled : "true", "false". If true, home directory access is enabled to public

# Shadow Copy
Save-NcCifsShadowCopySnapshot -Id 101 -VserverContext $vserver_name

# Create source and destination directory
New-Item -Path "\$vserver_lif\$qtree_share_name\src_dir" -ItemType directory
New-Item -Path "\$vserver_lif\$qtree_share_name\dest_dir" -ItemType directory

# Request the storage system to restore a directory
Restore-NcCifsShadowCopyDirectory -Volume $vol_name -SourceDirectory "/$qtree_share_name/src_dir" -DestinationDirectory "/$qtree_share_name/dest_dir" -DirectoryOnly

# DirectoryOnly : When specified, only the directory and no content will be restored, otherwise all content will be restored

4.4 Monitoring the CIFS Environment

Table 5 provides detailed information about the cmdlets that are used to monitor the CIFS server.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Cmdlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFS server</td>
<td>List all the CIFS servers</td>
<td>Get-NcCifsServer</td>
</tr>
<tr>
<td>CIFS server</td>
<td>List all the CIFS security tunable parameters</td>
<td>Get-NcCifsSecurity</td>
</tr>
<tr>
<td>CIFS server</td>
<td>List the CIFS specific tunable parameters</td>
<td>Get-NcCifsOption</td>
</tr>
<tr>
<td>CIFS local users and groups</td>
<td>List all the local users</td>
<td>Get-NcCifsLocalUser</td>
</tr>
<tr>
<td>CIFS local users and groups</td>
<td>List all the local groups</td>
<td>Get-NcCifsLocalGroup</td>
</tr>
<tr>
<td>CIFS local users and groups</td>
<td>List all the local users in a local group</td>
<td>Get-NcCifsLocalGroupMember</td>
</tr>
<tr>
<td>CIFS share and permissions</td>
<td>List all the CIFS shares</td>
<td>Get-NcCifsShare</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Get information about the NTFS security descriptor</td>
<td>Get-NcFileDirectorySecurityNtfs</td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
<td>Cmdlet</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Get information about discretionary access control entries</td>
<td>Get-NcFileDirectorySecurityNtfsDacl</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Get information about system/audit access control entries</td>
<td>Get-NcFileDirectorySecurityNtfsSacl</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Get security information about a file/folder</td>
<td>Get-NcFileDirectorySecurity</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Get information about policy tasks</td>
<td>Get-NcFileDirectorySecurityPolicyTask</td>
</tr>
<tr>
<td>Domain activities</td>
<td>Trigger the discovery of domain servers</td>
<td>Find-NcCifsDomainServer</td>
</tr>
<tr>
<td>Domain activities</td>
<td>Retrieve the list of discovered trusted domains</td>
<td>Get-NcCifsDomainTrust</td>
</tr>
<tr>
<td>Domain activities</td>
<td>List the CIFS domain password schedule objects</td>
<td>Get-NcCifsDomainPasswordSchedule</td>
</tr>
<tr>
<td>Domain activities</td>
<td>List the preferred domain controllers that are associated with an active directory</td>
<td>Get-NcCifsPreferredDomainController</td>
</tr>
<tr>
<td>Domain activities</td>
<td>Trigger rediscovery of trusted domains</td>
<td>Invoke-NcCifsDomainTrustsDiscovery</td>
</tr>
<tr>
<td>Domain activities</td>
<td>List the trusted domains</td>
<td>Get-NcCifsTrustedDomain</td>
</tr>
<tr>
<td>Home directory</td>
<td>List the CIFS home directory search path</td>
<td>Get-NcCifsHomeDirectorySearchPath</td>
</tr>
<tr>
<td>Shadow copy</td>
<td>List shadow copy EMS messages</td>
<td>Get-NcCifsShadowCopyEmsMessage</td>
</tr>
</tbody>
</table>

4.5 Cleaning the CIFS Environment

Table 6 and the following example code provide detailed information about the cmdlets that are used to clean the CIFS server.

Table 6) Cmdlets for cleaning the CIFS server.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Cmdlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home directory</td>
<td>Delete the home directory share</td>
<td>Remove-NcCifsShare</td>
</tr>
<tr>
<td>Home directory</td>
<td>Delete the CIFS home directory search path</td>
<td>Remove-NcCifsHomeDirectorySearchPath</td>
</tr>
<tr>
<td>Domain activities</td>
<td>Remove a trusted domain</td>
<td>Remove-NcCifsTrustedDomain</td>
</tr>
<tr>
<td>Domain activities</td>
<td>Remove a preferred domain controller</td>
<td>Remove-NcCifsPreferredDomainController</td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
<td>Cmdlet</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Delete an NTFS security descriptor</td>
<td>Remove-NcFileDirectorySecurityNtfs</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Remove a discretionary access control entry from an NTFS security descriptor</td>
<td>Remove-NcFileDirectorySecurityNtfsDacl</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Remove a system/audit access control entry from an NTFS security descriptor</td>
<td>Remove-NcFileDirectorySecurityNtfsSacl</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Remove the security policy</td>
<td>Remove-NcFileDirectorySecurityPolicy</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Remove a task from the policy</td>
<td>Remove-NcFileDirectorySecurityPolicyTask</td>
</tr>
<tr>
<td>File/folder permissions</td>
<td>Remove the Storage Level Access Guard (SLAG) from the specified volume junction path</td>
<td>Remove-NcFileDirectorySecuritySlag</td>
</tr>
<tr>
<td>CIFS share and permissions</td>
<td>Remove permissions for the local user on a share</td>
<td>Remove-NcCifsShareAcl</td>
</tr>
<tr>
<td>CIFS share and permissions</td>
<td>Remove permissions for the local group on a share</td>
<td>Remove-NcCifsShareAcl</td>
</tr>
<tr>
<td>CIFS share and permissions</td>
<td>Delete a CIFS share</td>
<td>Remove-NcCifsShare</td>
</tr>
<tr>
<td>CIFS local users and groups</td>
<td>Remove a CIFS local user from a CIFS local group</td>
<td>Remove-NcCifsLocalGroupMember</td>
</tr>
<tr>
<td>CIFS local users and groups</td>
<td>Remove a CIFS local user</td>
<td>Remove-NcCifsLocalUser</td>
</tr>
<tr>
<td>CIFS local users and groups</td>
<td>Remove a CIFS local group</td>
<td>Remove-NcCifsLocalGroup</td>
</tr>
<tr>
<td>CIFS server</td>
<td>Delete a CIFS server</td>
<td>Remove-NcCifsServer</td>
</tr>
</tbody>
</table>

# Destroy Home Directory
# Get the VServer LIF
$vserver_lif = (Get-NcNetInterface -Name $lif1_name -VserverContext $vserver_name).Address

# Destroy all the folders created for each user in the AD Group
for($i=0; $i -lt $ad_userlist.Count; $i++)
{
    $ad_user_dir = $ad_userlist[$i]
    Remove-Item -Path "$vserver_lif\$homeshare_name\$ad_user_dir" -Force
}

# Destroy the Home directory share
Remove-NcCifsShare -Name %w -VserverContext $vserver_name -Confirm:$false

# Destroy the CIFS Home directory search path
Remove-NcCifsHomeControllerDirectorySearchPath -Path "/$homedir_volume_name" -VserverContext $vserver_name -Confirm:$false

# Destroy the CIFS share
Remove-NcCifsShare -Name $homeshare_name -VserverContext $vserver_name -Confirm:$false
# Destroy the Volume
# Unmount the volume
Dismount-NcVol -Name $homedir_volume_name -VserverContext $vserver_name -Force -Confirm:$false
# Bring the Volume offline
Set-NcVol -Name $homedir_volume_name -Offline -VserverContext $vserver_name
# Destroy the Volume
Remove-NcVol -Name $homedir_volume_name -VserverContext $vserver_name -Confirm:$false

# Destroy domain related
# Remove from the list of trusted domains for name=mapping search
Remove-NcCifsTrustedDomain -TrustedDomain $trusted_domain -VserverContext $vserver_name
# Remove from a list of preferred domain controllers
Remove-NcCifsPreferredDomainController -Domain $trusted_domain -VserverContext $vserver_name

# Remove File/Folder permissions
$ntfs_sec_desc_id = "ntfssd"
$policy_name = "policy1"
$account = (Get-NcCifsLocalUser -UserName $localuser_name).UserName
# Delete a NTFS security descriptor
# Remove a discretionary access control entry from NTFS security descriptor
Remove-NcFileDirectorySecurityNtfsDacl -SecurityDescriptor $ntfs_sec_desc_id -Account $account -AccessType "allow" -VserverContext $vserver_name
# Remove a system/audit access control entry from NTFS security descriptor
# Remove the security policy
Remove-NcFileDirectorySecurityPolicy -Name $policy_name -VserverContext $vserver_name
# Get the junction path of the file/folder
$query = Get-NcVol -Template
$query.Name = $vol_name
$query.JunctionPath
$volume_junctionpath = (Get-NcVol -Query $query | select JunctionPath).JunctionPath
$qtree_junctionpath = "$volume_junctionpath/$qtree_name"
# Remove a task from the policy of a vserver
Remove-NcFileDirectorySecurityPolicyTask -Name $policy_name -Path $qtree_junctionpath -VserverContext $vserver_name
# Removes SLAG from the specified volume junction path
Remove-NcFileDirectorySecuritySlag -Path $qtree_junctionpath -VserverContext $vserver_name

# Remove CIFS Share permissions
# Remove permissions for the local user on qtree share
Remove-NcCifsShareAcl -Share $qtree_share_name -UserOrGroup $localuser_name -VserverName $server_name -Confirm:$false
# Remove permissions for the local group on volume share
Remove-NcCifsShareAcl -Share $vol_share_name -UserOrGroup $localgroup_name -VserverName $server_name -Confirm:$false
# Destroy CIFS Share
# Destroy the volume share
Remove-NcCifsShare -Name $qtree_share_name -VserverContext $vserver_name
# Destroy the qtree share
Remove-NcCifsShare -Name $vol_share_name -VserverContext $vserver_name

# Destroy Qtree and Volume
# Destroy Qtree
Remove-NcQtree -Volume $vol_name -Qtree qtree_name -VserverContext $vserver_name -Force -Confirm:$false

# Destroy Volume
# Unmount the volume
Dismount-NcVol -Name $vol_name -VserverContext $vserver_name -Force -Confirm:$false
# Bring the Volume offline
Set-NcVol -Name $vol_name -Offline -VserverContext $vserver_name
# Destroy the Volume
Remove-NcVol -Name $vol_name -VserverContext $vserver_name -Confirm:$false

# Destroy Qtree
# Destroy Volume
# Destroy Volumeoffline
Set-NcVol -Name $vol_name -Offline -VserverContext $vserver_name
# Destroy the Volume
Remove-NcVol -Name $vol_name -VserverContext $vserver_name -Confirm:$false

# Destroy CIFS Local Users and Groups
# Remove a CIFS Local user from CIFS local group
Remove-NcCifsLocalGroupMember -Name $localgroup_name -Member $localuser_name -VserverContext $vserver_name

# Destroy CIFS Local User
Remove-NcCifsLocalUser -UserName $localuser_name -VserverContext $vserver_name

# Destroy CIFS Local Group
Remove-NcCifsLocalGroup -Name $localgroup_name -VserverContext $vserver_name

# Destroy CIFS Server
Remove-NcCifsServer -VserverContext $vserver_name -AdminUsername $domain_username -AdminPassword $domain_password

# Destroy VServer
# Destroy LIF
Remove-NcNetInterface -Name $lif1_name -Vserver $vserver_name -Confirm:$false

# Destroy Subnet
Remove-NcNetSubnet -Name $subnet_name
# Bring the volume offline
Set-NcVol -Name $rootvolume_name -Offline -VserverContext $vserver_name
# Destroy the volume
Remove-NcVol -Name $rootvolume_name -VserverContext $vserver_name

# Destroy VServer
Remove-NcVserver -Name $vserver_name

# Destroy aggregate
# Remove-NcAggr -Name $aggr_name -Confirm:$false

------------------------------------------------------------------------------------

5 Managing Native Auditing

This section provides detailed information about the cmdlets that are used to manage native auditing.
Figure 7 summarizes the process, and Table 7 lists the cmdlets.
Figure 7) Managing native auditing.

Table 7) Cmdlets for managing native auditing.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Cmdlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provisioning native auditing</td>
<td>Create an audit configuration</td>
<td>New-NcFileserviceAudit</td>
</tr>
<tr>
<td>Modifying native auditing</td>
<td>Enable auditing</td>
<td>Enable-NcFileserviceAudit</td>
</tr>
<tr>
<td>Modifying native auditing</td>
<td>Rotate the file service audit log</td>
<td>Invoke-NcFileserviceAuditLogRotate</td>
</tr>
<tr>
<td>Modifying native auditing</td>
<td>Repair the audit infrastructure</td>
<td>Repair-NcFileserviceAudit</td>
</tr>
<tr>
<td>Modifying native auditing</td>
<td>Modify the audit configuration</td>
<td>Set-NcFileserviceAudit</td>
</tr>
<tr>
<td>Modifying native auditing</td>
<td>Disable auditing</td>
<td>Disable-NcFileserviceAudit</td>
</tr>
<tr>
<td>Monitoring native auditing</td>
<td>List audit configuration details</td>
<td>Get-NcFileserviceAudit</td>
</tr>
<tr>
<td>Cleaning native auditing</td>
<td>Remove the file service audit configuration</td>
<td>Remove-NcFileserviceAudit</td>
</tr>
</tbody>
</table>

# Provision Native Auditing
# Create a new Qtree to store Audit Log file
New-NcQtree -Volume $vol_name -Qtree "auditlog" -VserverContext $vserver_name -SecurityStyle "ntfs"

# Get the junction path the Qtree created to store Audit Log file
$_query = Get-NcVol -Template
$query.Name = $vol_name
$query.JunctionPath
$volume_junctionpath = (Get-NcVol -Query $query | select JunctionPath).JunctionPath
$auditlog_path = "$volume_junctionpath/auditlog"

# Create audit configuration for a Vserver
New-NcFileserviceAudit -Destination $auditlog_path -RotateLimit 20 -RotateSize 50MB -VserverContext $vserver_name

# Modify Native Auditing
# Enable Auditing for Vserver
Enable-NcFileserviceAudit -VserverContext $vserver_name

# Rotate fileservice audit log
Invoke-NcFileserviceAuditLogRotate -VserverContext $vserver_name

# Repair the audit infrastructure
Repair-NcFileserviceAudit -VserverContext $vserver_name

# Get the junction path the Qtree created to store Audit Log file
$_query = Get-NcVol -Template
$query.Name = $vol_name
$query.JunctionPath

$volume_junctionpath = (Get-NcVol -Query $query | select JunctionPath).JunctionPath
$auditlog_path = "$volume_junctionpath/auditlog"

# Modify audit configuration
Set-NcFileserviceAudit -Destination $auditlog_path -AuditEvents "cifs_logon_logoff" -RotateLimit 20 -AuditGuarantee $true -RotateSize 50MB -Format "evtx" -VserverContext $vserver_name

# Possible values for -AuditEvents:
# 'file_ops' - File Operation Events,
# 'cifs_logon_logoff' - CIFS Logon and Logoff Events,
# 'cap_staging' - Central Access Policy Staging Events
# If not specified, then default is 'file_ops,cifs_logon_logoff,cap_staging'

# Possible values for -AuditGuarantee : "true", "false". Default value is true

# Possible values for -Format:
# 'xml' - Data ONTAP-Specific XML Log Format,
# 'evtx' - Microsoft Windows EVTX Log Format
# If not specified, then default is 'evtx'

# -RotateSize : Default value is 100MB

# Disable Auditing for Vserver
Disable-NcFileserviceAudit -VserverContext $vserver_name

# Clean Native Auditing
# Remove the fileservice audit configuration
Remove-NcFileserviceAudit -VserverContext $vserver_name

# Remove the Qtree used to store Audit Log file
Remove-NcQtree -Volume $vol_name -Qtree "auditlog" -VserverContext $vserver_name -Force -Confirm:$false

6 Managing the Antivirus Environment

This section provides detailed information about the cmdlets that are used to manage the antivirus environment. Figure 8 summarizes the process, and
Table 8 lists the cmdlets.

Figure 8) Managing the antivirus environment.
## Table 8: Cmdlets for managing the antivirus environment.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Cmdlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provisioning the antivirus environment</td>
<td>Create a virus scanner pool</td>
<td><code>New-NcVscanScannerPool</code></td>
</tr>
<tr>
<td>Provisioning the antivirus environment</td>
<td>Create a Vscan on-access policy</td>
<td><code>New-NcVscanOnAccessPolicy</code></td>
</tr>
<tr>
<td>Provisioning the antivirus environment</td>
<td>Create a new antivirus log entry</td>
<td><code>New-NcAntivirusLogEntry</code></td>
</tr>
<tr>
<td>Provisioning the antivirus environment</td>
<td>Create an on-demand task</td>
<td><code>New-NcVscanOnDemandTask</code></td>
</tr>
<tr>
<td>Modifying the antivirus environment</td>
<td>Enable the Vscan scanner</td>
<td><code>Enable-NcVscan</code></td>
</tr>
<tr>
<td>Modifying the antivirus environment</td>
<td>Discard cached information about scanned files</td>
<td><code>Reset-NcVscan</code></td>
</tr>
<tr>
<td>Modifying the antivirus environment</td>
<td>Resolve the host names that are configured in the scanner pool</td>
<td><code>Resolve-NcVscanHostnames</code></td>
</tr>
<tr>
<td>Modifying the antivirus environment</td>
<td>Disable the Vscan scanner</td>
<td><code>Disable-NcVscan</code></td>
</tr>
<tr>
<td>Modifying the antivirus environment</td>
<td>Enable a Vscan on-access policy</td>
<td><code>Enable-NcVscanOnAccessPolicy</code></td>
</tr>
<tr>
<td>Modifying the antivirus environment</td>
<td>Disable a Vscan on-access policy</td>
<td><code>Disable-NcVscanOnAccessPolicy</code></td>
</tr>
<tr>
<td>Modifying the antivirus environment</td>
<td>Modify a Vscan scanner pool</td>
<td><code>Set-NcVscanScannerPool</code></td>
</tr>
<tr>
<td>Modifying the antivirus environment</td>
<td>Modify a Vscan on-access policy</td>
<td><code>Set-NcVscanOnAccessPolicy</code></td>
</tr>
<tr>
<td>Modifying the antivirus environment</td>
<td>Set the attributes of the antivirus engine</td>
<td><code>Set-NcAntivirusEngine</code></td>
</tr>
<tr>
<td>Modifying the antivirus environment</td>
<td>Modify the attributes of antivirus engine options</td>
<td><code>Set-NcAntivirusEngineOption</code></td>
</tr>
<tr>
<td>Modifying the antivirus environment</td>
<td>Set quarantine information for the antivirus engine</td>
<td><code>Set-NcAntivirusRemedy</code></td>
</tr>
<tr>
<td>Modifying the antivirus environment</td>
<td>Set version information for the antivirus engine</td>
<td><code>Set-NcAntivirusVersion</code></td>
</tr>
<tr>
<td>Modifying the antivirus environment</td>
<td>Start an antivirus update job</td>
<td><code>Start-NcAntivirusUpdate</code></td>
</tr>
<tr>
<td>Modifying the antivirus environment</td>
<td>Run an on-demand task</td>
<td><code>Invoke-NcVscanOnDemandTask</code></td>
</tr>
<tr>
<td>Modifying the antivirus environment</td>
<td>Modify an on-demand task</td>
<td><code>Set-NcVscanOnDemandTask</code></td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
<td>Cmdlet</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>--------------------------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Monitoring the antivirus environment</td>
<td>List Vscan scanner pools</td>
<td>Get-NcVscanScannerPool</td>
</tr>
<tr>
<td>Monitoring the antivirus environment</td>
<td>List Vscan on-access policies</td>
<td>Get-NcVscanOnAccessPolicy</td>
</tr>
<tr>
<td>Monitoring the antivirus environment</td>
<td>List the Vscan connection status</td>
<td>Get-NcVscanConnection</td>
</tr>
<tr>
<td>Monitoring the antivirus environment</td>
<td>List the extended status for Vscan connections</td>
<td>Get-NcVscanConnectionStats</td>
</tr>
<tr>
<td>Monitoring the antivirus environment</td>
<td>List the contents of the Vscan event log</td>
<td>Get-NcVscanEvents</td>
</tr>
<tr>
<td>Monitoring the antivirus environment</td>
<td>List the active scanner pool configuration</td>
<td>Get-NcVscanScannerPoolActive</td>
</tr>
<tr>
<td>Monitoring the antivirus environment</td>
<td>Get information about the Vscan status</td>
<td>Get-NcVscanStatus</td>
</tr>
<tr>
<td>Monitoring the antivirus environment</td>
<td>List the attributes of the antivirus engine</td>
<td>Get-NcAntivirusEngine</td>
</tr>
<tr>
<td>Monitoring the antivirus environment</td>
<td>List the antivirus engine options</td>
<td>Get-NcAntivirusEngineOption</td>
</tr>
<tr>
<td>Monitoring the antivirus environment</td>
<td>List the antivirus objects</td>
<td>Get-NcAntivirusLog</td>
</tr>
<tr>
<td>Monitoring the antivirus environment</td>
<td>Get remedy information about the antivirus engine</td>
<td>Get-NcAntivirusRemedy</td>
</tr>
<tr>
<td>Monitoring the antivirus environment</td>
<td>Get version information about the antivirus engine</td>
<td>Get-NcAntivirusVersion</td>
</tr>
<tr>
<td>Monitoring the antivirus environment</td>
<td>List on-demand tasks</td>
<td>Get-NcVscanOnDemandTask</td>
</tr>
<tr>
<td>Monitoring the antivirus environment</td>
<td>List on-demand task reports</td>
<td>Get-NcVscanOnDemandReport</td>
</tr>
<tr>
<td>Cleaning the antivirus environment</td>
<td>Delete a Vscan on-access policy</td>
<td>Remove-NcVscanOnAccessPolicy</td>
</tr>
<tr>
<td>Cleaning the antivirus environment</td>
<td>Delete an on-demand report</td>
<td>Remove-NcVscanOnDemandReport</td>
</tr>
<tr>
<td>Cleaning the antivirus environment</td>
<td>Delete an on-demand task</td>
<td>Remove-NcVscanOnDemandTask</td>
</tr>
<tr>
<td>Cleaning the antivirus environment</td>
<td>Delete a Vscan scanner pool</td>
<td>Remove-NcVscanScannerPool</td>
</tr>
</tbody>
</table>

# Provision Antivirus
$vscan_pool_name = "vscanpool1"
$vscan_server = "1.1.1.1"
$vscan_policy_name = "vscanpolicy1"
# This sets the attributes of the antivirus engine installed

# This sets the attributes of the antivirus engine installed

# This sets the attributes of the antivirus engine installed

# This sets the attributes of the antivirus engine installed

# This sets the attributes of the antivirus engine installed

# This sets the attributes of the antivirus engine installed

# This sets the attributes of the antivirus engine installed

# This sets the attributes of the antivirus engine installed
Modify the attributes of antivirus engine options
Set-NcAntivirusEngineOption -GroupArchiveUnpack $true -MaxRecursionDepth 5 -HeuristicAnalysis $true -MacroAnalysis $true -ScanMime $true -MimeLinesToScan 5 -CacheSize 512k -ProxyHost "http://proxy_server_name" -ProxyPort 2552 -ProxyLogin "admin" -ProxyPassword "netapp1!" -DecompressedFileCountLimit 256 -DecompressedFileSizeLimit 512 -DecompressionSizeFactor 256 -IsSpywareEnabled $true

Set the quarantine information of the Anti-Virus engine installed
Set-NcAntivirusRemedy -Action "repair" -Extension ".vir" -Repair "quarantine" -Directory "/quarantine"

Set the version information of the antivirus engine installed
Set-NcAntivirusVersion -ClusterVersion "8.4" -ComponentUpdated "both" -Vsapi32BitVersion "xxx-x" -Vsapi64BitVersion "xxx-x" -VirusPatternVersion "xxx-x" -SpywarePatternVersion "xxx-x"

Start an antivirus update job
Start-NcAntivirusUpdate -Sync

Clean Antivirus
$vscan_pool_name = "vscanpool1"
$vscan_policy_name = "vscanpolicy1"

# Delete a Vscan On-Access policy
Remove-NcVscanOnAccessPolicy -Name $vscan_policy_name -VserverContext $vserver_name

# Delete an On-Demand report
Remove-NcVscanOnDemandReport

# Delete an On-Demand Task
Remove-NcVscanOnDemandTask

# Delete a Vscan scanner pool
Remove-NcVscanScannerPool -Name $vscan_pool_name -VserverContext $vserver_name
7 Managing FPolicy

This section provides detailed information about the cmdlets that are used to manage the NetApp FPolicy feature. Figure 9 shows an overview of the process, and Table 9 lists the cmdlets.

Figure 9) Managing FPolicy.

Table 9) Cmdlets for managing FPolicy.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Cmdlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provisioning FPolicy</td>
<td>Create a new event for the CIFS protocol</td>
<td>New-NcFpolicyEvent</td>
</tr>
<tr>
<td>Provisioning FPolicy</td>
<td>Create a new FPolicy policy external engine</td>
<td>New-NcFpolicyExternalEngine</td>
</tr>
<tr>
<td>Provisioning FPolicy</td>
<td>Create a new FPolicy policy</td>
<td>New-NcFpolicyPolicy</td>
</tr>
<tr>
<td>Provisioning FPolicy</td>
<td>Create FPolicy scope options</td>
<td>New-NcFpolicyScope</td>
</tr>
<tr>
<td>Modifying FPolicy</td>
<td>Enable a specific named policy</td>
<td>Enable-NcFpolicyPolicy</td>
</tr>
<tr>
<td>Modifying FPolicy</td>
<td>Disable a specific named policy</td>
<td>Disable-NcFpolicyPolicy</td>
</tr>
<tr>
<td>Modifying FPolicy</td>
<td>Make a connection to an FPolicy server</td>
<td>Connect-NcFpolicyServer</td>
</tr>
<tr>
<td>Modifying FPolicy</td>
<td>Terminate a connection to an FPolicy server</td>
<td>Disconnect-NcFpolicyServer</td>
</tr>
<tr>
<td>Modifying FPolicy</td>
<td>Set FPolicy event options</td>
<td>Set-NcFpolicyEvent</td>
</tr>
<tr>
<td>Modifying FPolicy</td>
<td>Modify an FPolicy external engine</td>
<td>Set-NcFpolicyExternalEngine</td>
</tr>
<tr>
<td>Modifying FPolicy</td>
<td>Modify a policy</td>
<td>Set-NcFpolicyPolicy</td>
</tr>
<tr>
<td>Modifying FPolicy</td>
<td>Create an export policy</td>
<td>New-NcExportPolicy</td>
</tr>
<tr>
<td>Modifying FPolicy</td>
<td>Set FPolicy scope options</td>
<td>Set-NcFpolicyScope</td>
</tr>
<tr>
<td>Monitoring FPolicy</td>
<td>List FPolicy event options</td>
<td>Get-NcFpolicyEvent</td>
</tr>
<tr>
<td>Monitoring FPolicy</td>
<td>Get information about external engines</td>
<td>Get_NcFpolicyExternalEngine</td>
</tr>
<tr>
<td>Monitoring FPolicy</td>
<td>Get information about policies</td>
<td>Get-NcFpolicyPolicy</td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
<td>Cmdlet</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Monitoring FPolicy</td>
<td>List FPolicy scope options</td>
<td>Get-NcFpolicyScope</td>
</tr>
<tr>
<td>Monitoring FPolicy</td>
<td>Get information about the passthrough read connection status</td>
<td>Get-NcFpolicyPassthroughReadConnection</td>
</tr>
<tr>
<td>Monitoring FPolicy</td>
<td>Get information about the FPolicy server status</td>
<td>Get-NcFpolicyServerStatus</td>
</tr>
<tr>
<td>Monitoring FPolicy</td>
<td>Get information about the FPolicy policy status</td>
<td>Get-NcFpolicyStatus</td>
</tr>
<tr>
<td>Cleaning FPolicy</td>
<td>Delete an FPolicy event</td>
<td>Remove-NcFpolicyEvent</td>
</tr>
<tr>
<td>Cleaning FPolicy</td>
<td>Delete an FPolicy external engine</td>
<td>Remove-NcFpolicyExternalEngine</td>
</tr>
<tr>
<td>Cleaning FPolicy</td>
<td>Delete an FPolicy policy scope</td>
<td>Remove-NcFpolicyScope</td>
</tr>
<tr>
<td>Cleaning FPolicy</td>
<td>Delete a policy</td>
<td>Remove-NcFpolicyPolicy</td>
</tr>
</tbody>
</table>

# Provision FPolicy

```powershell
# Create a new event for the CIFS protocol
New-NcFpolicyEvent -Name $event_name -Protocol cifs -FileOperation create,delete,read,write -VserverContext $vserver_name
```

```powershell
# Create a new FPolicy policy external engine
New-NcFpolicyExternalEngine -Name $external_engine_name -PrimaryServer 1.1.1.1 -Port 2357 -SslOption no_auth -VserverContext $vserver_name
```

```powershell
# Create a new FPolicy policy
New-NcFpolicyPolicy -Name $fpolicy_name -Event $event_name -EngineName $external_engine_name -NonMandatory -VserverContext $vserver_name
```

```powershell
# Create FPolicy scope options
New-NcFpolicyScope -PolicyName $fpolicy_name -VserverContext $vserver_name
```

# Modify FPolicy

```powershell
# Enable a specific named policy
Enable-NcFpolicyPolicy -Name $fpolicy_name -SequenceNumber 1 -VserverContext $vserver_name
```

```powershell
# Disable a specific named policy
Disable-NcFpolicyPolicy -Name $fpolicy_name -VserverContext $vserver_name
```

```powershell
# Make a connection to FPolicy server
Connect-NcFpolicyServer -PolicyName $fpolicy_name -FpolicyServer $fpolicy_server -Node $cluster_nodes[0] -VserverContext $vserver_name
```

```powershell
# Terminate connection to FPolicy server
Disconnect-NcFpolicyServer -PolicyName $fpolicy_name -FpolicyServer $fpolicy_server -Node $cluster_nodes[0] -VserverContext $vserver_name
```

```powershell
# Set FPolicy event options
Set-NcFpolicyEvent -Name $event_name -Protocol cifs -FileOperation create,delete,read,write -Filter "first_read,first_write" -VolumeOperation $true -VserverContext $vserver_name
```

Possible values for -Protocol : "cifs", "nfsv3", "nfsv4"

Possible values for -FileOperation:

- "close" - File close operation
- "create" - File create operation
- "create_dir" - File create directory operation
- "delete" - File delete operation
"delete_dir" - Directory delete operation
"getattr" - Get attribute operation
"link" - Link operation
"lookup" - Lookup operation
"open" - File open operation
"read" - File read operation
"write" - File write operation
"rename" - File rename operation
"rename_dir" - Directory rename operation
"setattr" - Set attribute operation
"symlink" - Symbolic link operation

Possible values for -Filter:
# "monitor_ads" - Monitor alternate data stream
# "close_with_modification" - Filter close with modification
# "close_without_modification" - Filter close without modification
# "first_read" - Filter first read
# "first_write" - Filter first write
# "offline_bit" - Filter offline bit set
# "open_with_delete_intent" - Filter open with delete intent
# "open_with_write_intent" - Filter open with write intent
# "write_with_size_change" - Filter write with size change

Possible values for -VolumeOperation: "true", "false". If true, the volume operation required for the event

Modify an FPolicy external engine

# Modify an FPolicy policy
Set-NcFpolicy -Name $fpolicy_name -Event $event_name -EngineName $external_engine_name -Mandatory -AllowPrivilegedAccess true -PrivilegedUserName $localuser_name -IsPassThroughReadEnabled True -VserverContext $vserver_name

Create export policies
$export_policy1 = "export_policy1"
$export_policy2 = "export_policy2"
New-NcExportPolicy -Name $export_policy1 -VserverContext $vserver_name
New-NcExportPolicy -Name $export_policy2 -VserverContext $vserver_name

# Set FPolicy scope options
Set-NcFpolicyScope -PolicyName $fpolicy_name -SharesToInclude $qtree_share_name -SharesToExclude $vol_share_name -VolumesToInclude "vol*" -VolumesToExclude $vol_name -ExportPoliciesToInclude $export_policy1 -ExportPoliciesToExclude $export_policy2 -FileExtensionsToInclude "vhd, vhdx, vmdk" -FileExtensionsToExclude "txt" -CheckExtensionsOnDirectories $true -IsMonitoringOfObjectsWithNoExtensionEnabled $true -VserverContext $vserver_name

# -CheckExtensionsOnDirectories : If true, directory names are subjected to extensions check, similar to file names
# -IsMonitoringOfObjectsWithNoExtensionEnabled : Indicates whether monitoring of Objects with no extension is enabled or not

# Clean Fpolicy
$event_name = "cifs_event"
$external_engine_name = "engine1"
$fpolicy_name = "fpolicy1"

# Delete FPolicy event
Remove-NcFpolicyEvent -Name $event_name -VserverContext $vserver_name

# Delete an FPolicy external engine
Remove-NcFpolicyExternalEngine -Name $external_engine_name -VserverContext $vserver_name

# Delete an FPolicy policy scope
Remove-NcFpolicyScope -PolicyName $fpolicy_name -VserverContext $vserver_name

# Delete a policy
Remove-NcFpolicyPolicy -Name $fpolicy_name -VserverContext $vserver_name

#------------------------------------------------------------------------------------
# Clean Fpolicy
#------------------------------------------------------------------------------------

8 Managing Day-to-Day Tasks

This section provides detailed information about managing day-to-day tasks. Regular tasks include:

- Reviewing critical share permissions
- Displaying share NTFS ACLs (Get-NcFileDirectorySecurity)

![Output of Get-NcFileDirectorySecurity]

- Listing current CIFS sessions and details (Get-NcCifsSession)
- Determining which client opened the file (`Get-NcCifsSession, Get-NcCifsSessionFile`)

- Displaying locks (`Get-NcLock`)

- Invoking the CLI commands from NetApp PowerShell Toolkit (PSTK)

```powershell
# CLI commands can be invoked from PSTK using cmdlet 'Invoke-NcSSh'
# Basic example to invoke a CLI command for secd
$cmd_set_diag = "set diag -confirmations off"
for($i=0; $i -lt $cluster_nodes.Count; $i++)
{
    $node = $cluster_nodes[$i]
    $cmd_secd = "statistics secd show -node $node -vserver $vserver_name"
    Invoke-NcSsh -Command $cmd_set_diag
    Invoke-NcSsh -Command "$cmd_set_diag; $cmd_secd"
}
```

- Retrieving the CIFS performance counters
# Retrieve CIFS performance objects
$perf_object = Get-NcPerfObject | where {$_.Name -match 'cifs'}
# This will return the following perf objects : cifs, cifs:node, cifs:vserver, cifs_cap,
cifs_cap:constituent, cifs_shadowcopy, cifs_watch and nblade_cifs

# For each of the retrieved performance objects, get the perf counters
foreach($object in $perf_object) {
    # Get list of performance counters
    $perf_counters = Get-NcPerfCounter -Name $object.Name
    # Get list of performance object instances
    $instance = Get-NcPerfInstance -Name $object.Name
    $mod_perf_counters = @()
    foreach($counter in $perf_counters) {
        $mod_counter = [pscustomobject]@{Name=$counter.Name;Unit=$counter.Unit}
        $mod_perf_counters += $mod_counter
    }
    $arr = @()
    for ($i=0; $i -lt $mod_perf_counters.Count; $i++) {
        $name = $mod_perf_counters[$i].Name
        $unit = $mod_perf_counters[$i].Unit
        # Get list of current counter values of instances of an object
        $perf_data = Get-NcPerfData -Name $object.Name -Instance $instance.Name -Counter $mod_perf_counters[$i].Name
        $arritem = [pscustomobject]@{Name=$name;Unit=$unit;Value=$perf_data.Counters.Value}
        $arr += $arritem
    }
    Write-Host "---------------------------------------------------------------------"
    Write-Host " Perf Object : " $object.Name
    Write-Host "---------------------------------------------------------------------"
    $arr
    Write-Host "---------------------------------------------------------------------"
}

References

The following references were used in this document.

- **System Administration Guide for Cluster Administrators**
- **Best Practices Guide for Windows File Services**
- **CIFS/SMB Configuration Express Guide**
  [https://library.netapp.com/ecm/ecm_get_file/ECMP1547457](https://library.netapp.com/ecm/ecm_get_file/ECMP1547457)
- **ONTAP File Access and Protocols Management Guide**
  [https://fieldportal.netapp.com/content/200567](https://fieldportal.netapp.com/content/200567)
- **Getting Started with PSTK**
- **Making the Most of PSTK**
- **ONTAP PowerShell Toolkit Primer**
  https://fieldportal.netapp.com/content/316607
Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Copyright Information

Copyright © 1994–2017 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark Information

NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.