Project topic: Automating Common IT Tasks with PowerShell

Project Overview:

This project demonstrates how PowerShell can be used to automate common IT administrative tasks on Windows systems, improving efficiency, consistency, and security in enterprise environments. The scripts cover system monitoring, user management, file backups, and network configuration. By automating these processes, IT teams can minimize manual errors, enforce standardized policies, and reduce the time required to maintain corporate networks.

Project Agenda: To develop and deploy PowerShell scripts to manage an IT corporate network by automating common administrative tasks on a Windows operating system and maintaining a secure and organized system. Generate reports for auditing and monitoring purposes.

Project Description:

The following tasks were automated;

1.) Monitored system events

- Implemented a script that monitors security logs and generates a report file containing filtered security events.

2.) Managed users

- Implemented a script that adds multiple users, sets their passwords, and assigns them to groups.

3.) Backed Up files

- Implemented a script that automatically creates backups of files from a specific directory to another location.

4.) Configured network

- Implemented a script that sets a static IP address, subnet mask, default gateway, and DNS server for a network adapter.

Tools Required:

Windows Server 2022 VM
PowerShell (built-in on windows)
Notepad for script editing

Deliverables:

Multiple PowerShell scripts that:

- Generate event log reports
- Automate user creation & group assignments
- Backup files automatically
- Configure static IP settings

Output reports/files for validation and auditing

Usage: Created and run the following PowerShell scripts as an administrator;

1.) Created a script for automated system monitoring and reporting:

```
# Script to monitor system events for security logs
$BeginTime = (Get-Date).AddHours(-24)

$Events = Get-WinEvent -LogName Security | Where-Object {
    $_.TimeCreated -ge $BeginTime -and ($_.LevelDisplayName -eq "Information" -or $_.LevelDisplayName -eq "Error")
}

$Events | Select-Object TimeCreated, Id, LevelDisplayName, Message |
  Format-Table -AutoSize |
  Out-File "$env:USERPROFILE\Desktop\SecurityLogReport.txt"

Write-Host "Security Log Report generated at $env:USERPROFILE\Desktop\SecurityLogReport.txt"
```

Run the script for event monitoring using the command below;

.\MonitorSystemEvents.ps1

2.) Created a script to optimize user management:

```
# Script to add multiple users and assign them to groups
$Users = @(
@{ Name = "JohnDoe"; Password = "P@ssw0rd1"; Group = "Administrators" },
@{ Name = "JaneSmith"; Password = "P@ssw0rd2"; Group = "Users" }
)
foreach ($User in $Users) {
$SecurePassword = ConvertTo-SecureString -String $User.Password -AsPlainText -Force
New-LocalUser -Name $User.Name -Password $SecurePassword -FullName $User.Name -Description "Added by script" -AccountNeverExpires
Add-LocalGroupMember -Group $User.Group -Member $User.Name
}
Write-Host "Users successfully created and assigned to respective groups."
```

Run the script for user management using the command:

.\ManageUsers.ps1

3.) Created the script below to streamline file management and backup:

```
# Script to backup files
$SourceDir = "C:\ImportantFiles"
$BackupRoot = "E:\Backups"
$BackupDir = "$BackupRoot\$(Get-Date -Format 'yyyy-MM-dd')"
# Check if backup directory exists
if (-Not (Test-Path $BackupDir)) {
New-Item -ItemType Directory -Path $BackupDir -Force
Write-Host "Backup directory created at: $BackupDir"
```

```
}
# Copy files
Copy-Item -Path "$SourceDir\*" -Destination $BackupDir -Recurse -Force
Write-Host "Files copied from $SourceDir to $BackupDir successfully."
```

Run the backup script using the command below;
.\BackupFiles.ps1

4.) Designed an approach for network configuration;

```
# Script to configure network settings
$InterfaceAlias = "Ethernet"
$IPAddress = "192.168.1.100"
$SubnetPrefix = 24
$DefaultGateway = "192.168.1.1"
$DNSServer = "8.8.8.8"
# Set IP address and gateway
New-NetIPAddress -InterfaceAlias $InterfaceAlias -IPAddress $IPAddress -PrefixLength $SubnetPrefix -DefaultGateway $DefaultGateway -Confirm:$false
# Set DNS server
Set-DnsClientServerAddress -InterfaceAlias $InterfaceAlias -ServerAddresses $DNSServer
Write-Host "Network configuration updated successfully."
```

Run the network configuration script using the command below;
.\ConfigureNetwork.ps1

Final Outcome

- Automated system monitoring with reports
- Automated user creation & group management
- Reliable backup process for critical files
- Standardized network configurations

Conclusion and Relevance

In modern IT environments, efficiency, scalability, and security are critical. Manual execution of routine administrative tasks increases the risk of human error, wastes valuable time, and makes it difficult to maintain consistent security practices across systems. By automating these tasks with PowerShell, this project demonstrates how IT teams can:

- **Enhance Security Monitoring**: Automated log collection and reporting ensures timely detection of suspicious activity, supporting incident response and compliance requirements.

- **Improve User Management**: Scripting account creation and group assignments reduces onboarding time and enforces standardized access controls.

- **Streamline File Management**: Automated backups safeguard sensitive data, ensuring business continuity and resilience against data loss.

- **Simplify Network Configuration**: Configuring network adapters through scripts minimizes misconfiguration risks and ensures consistent deployment across multiple devices.

This project highlights the **industry relevance of automation in system administration**, showing how PowerShell can be leveraged as a practical tool for IT operations, compliance, and cybersecurity readiness. It reflects real-world practices where enterprises use scripting to improve **efficiency, reduce risk, and enforce policy consistency** across their infrastructure.

Prepared by: Janefrances Nwachukwu
Date: 08/19/2025