

```
try { Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True }  
catch { "Firewall on, probably" }  
# explanation: try { } executes anything in brackets  
# in this case it's a firewall on rule on the Domain, Public and Private profiles  
# if it somehow breaks, it catches the .NET error code.  
# code does work if firewall is already on  
# to turn firewall off, turn True to False
```

#Secpol modification

```
secedit /export /cfg "20240923-Test.cfg" #creates config file to later be modified  
secedit /configure /db c:\windows\security\local.sdb /cfg "20240923-Test.cfg" /areas  
SECURITYPOLICY  
#Make sure you're in the same directory as your config file (config file here is named after date)  
# can modify more areas with /areas [insert area in this case SECURITYPOLICY]  
# ideal situation is that I make a template config file, you upload file, and have that in place of  
step one, you just skip to command 2
```

#turning off features

```
#this turns off FTP, first research the FeatureName of the feature you which to switch off  
#swap it in the quotes  
Disable-WindowsOptionalFeature -Online -FeatureName "IIS-FTPSvc" -NoRestart  
Disable-WindowsOptionalFeature -Online -FeatureName "SMB1Protocol" -NoRestart  
Disable-WindowsOptionalFeature -Online -FeatureName "TelnetClient" -NoRestart  
Disable-WindowsOptionalFeature -Online -FeatureName "WorkFolders-Server" -NoRestart  
Disable-WindowsOptionalFeature -Online -FeatureName "IIS-WebServerRole" -NoRestart
```

Install the Windows Update module

```
Install-Module -Name PSWindowsUpdate -Force
```

Import the Windows Update module

```
Import-Module PSWindowsUpdate
```

Check for updates

```
Get-WindowsUpdate -AcceptAll -Install -AutoReboot
```

Restart the system if updates require a reboot

```
Restart-Computer -Force
```

#spam Y

#Services

```
Set-Service -Name Spooler -StartupType Disabled  
Set-Service -Name SNMPTRAP -StartupType Disabled  
Set-Service -Name SSDPSRV -StartupType Disabled  
Set-Service -Name Imhosts -StartupType Disabled
```

```
Set-Service -Name tapisrv -StartupType Disabled
Set-Service -Name upnphost -StartupType Disabled
Set-Service -Name EventLog -StartupType Automatic
Set-Service -Name wuauserv -StartupType Automatic
Set-Service -Name mpssvc -StartupType Automatic
Set-Service -Name WinDefend -StartupType Automatic
Set-Service -Name Sense -StartupType Automatic
```

```
#Hashes
#Need file hashes?
#Use Get-FileHash
# [-Path] <String[]>
# [[-Algorithm] <String>]
# [<CommonParameters>]
```

```
#User passwords
#instructions, put name of each valid user you want to change passwords in the apostrophes
```

```
$Password = (ConvertTo-SecureString -AsPlainText "Th3Ultim@t3P@ssw0rd" -Force)
$UserList = @('dovahkiin', 'delphine', 'esbern', 'arngeir', 'paarthurnax', 'borri', 'einarth', 'wulfgar',
'ulfric', 'tullius', 'ralof', 'hadvar', 'balgruuf', 'farengar', 'irileth', 'lydia')
foreach ($user in $UserList)
{
    Set-LocalUser -Name "$user" -Password $Password
}
#Deleting Users WIP
$appUserList = @('dovahkiin', 'delphine', 'esbern', 'arngeir', 'paarthurnax', 'borri', 'einarth',
'wulfgar', 'ulfric', 'tullius', 'ralof', 'hadvar', 'balgruuf', 'farengar', 'irileth', 'lydia', 'Guest',
'Administrator', 'WDAGUtilityAccount')
$totalUserList = @('dovahkiin', 'delphine', 'esbern', 'arngeir', 'paarthurnax', 'borri', 'einarth',
'wulfgar', 'ulfric', 'tullius', 'ancano', 'tolfdur', 'ralof', 'hadvar', 'balgruuf', 'farengar', 'irileth', 'lydia',
'Guest', 'Administrator', 'WDAGUtilityAccount')

foreach ($user in $totalUserList)
{
    if ( -not ( $user in $appUserList ) )
    {
        Remove-LocalUser -Name "$user" -Confirm
    }
}
```