<http://www.lazywinadmin.com/>

<https://lazywinadmin.github.io/>

<https://techtalk.gfi.com/11-most-useful-powershell-commands-for-remote-management/>

<http://searchwindowsserver.techtarget.com/tip/Top-25-Windows-PowerShell-commands-for-administrators>

<https://www.reddit.com/r/usefulscripts/?count=50&after=t3_7luw2a>

<https://blogs.technet.microsoft.com/heyscriptingguy/2015/08/19/parsing-netstat-information-with-powershell-5/>

$filename = (get-content files.txt)

get-childitem -path c:\ -include $filename -file -recurse -force -erroraction SilentlyContinue

$ips = (Get-Content ips.txt)

$winips = $ips | %{test-wsman $\_ 2>$null}

$reg = $(Get-Item HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run).property

$reg | ?{$(Get-ItemProperty HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run)."$\_" -match ".exe"} |

%{$property=@{"$\_" = $(Get-ItemProperty HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run)."$\_"}; $property

}

**Ping sweep to a text file:**

$ping = new-object system.net.networkinformation.ping

$(1..254 | %{$ping.send(“192.168.13.$\_”) | select status,address | ?{$\_.status -match “Success”}} | %{$\_.address.tostring()}) > ips.txt

**Ping sweep to an array variable named success:**

$ping = new-object system.net.networkinformation.ping

$success = @( $(1..254 | %{$ping.send(“192.168.13.$\_”) | select status,address | ?{$\_.status -match “Success”}} | %{$\_.address.tostring()}))

Get-childitem | %{get-filehash $\_.FullName}| %

**Getting strings on a file:**

$thing = [system.io.file]::ReadAllBytes(“c:\fullpath\file.exe)

$strings1 = [system.text.encoding]::ASCII.GetString($thing)

$strings2 = [system.text.encoding]::Unicode.Getstring($thing)

Get-content “c:\path\file” | out-string > strings.txt

$a = Get-Content iTunesHelper.exe

$a -split " " | Select-String -Pattern "\%{1,30}.exe"

$a = Get-Content .\iTunesHelper.exe

$a -split " " | Select-String -Pattern "(\d{1,3}\.){3}\d{1,3}"

**Named pipes:**

[system.io.directory]::getfiles(“\\.\pipe”)

**Encoding/Decoding:**

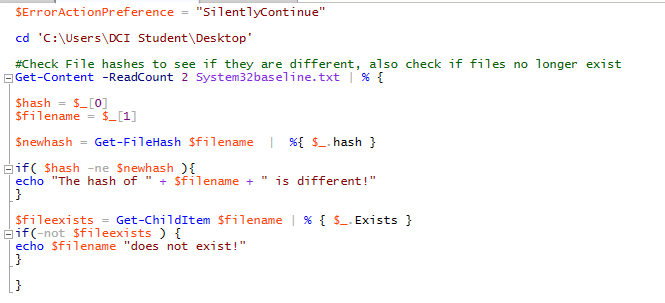
$string = "burrito"

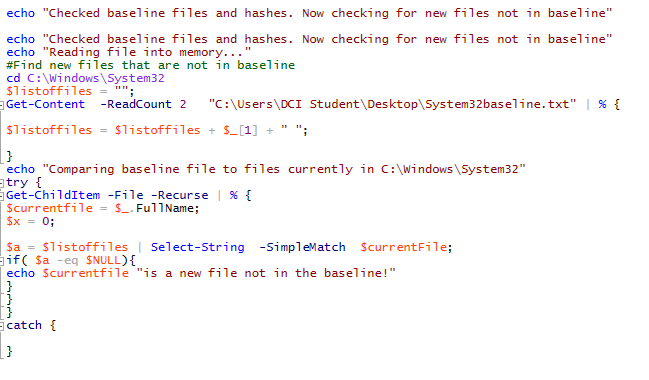
$stringbytes = [System.Text.encoding]::ASCII.Getbytes($string)

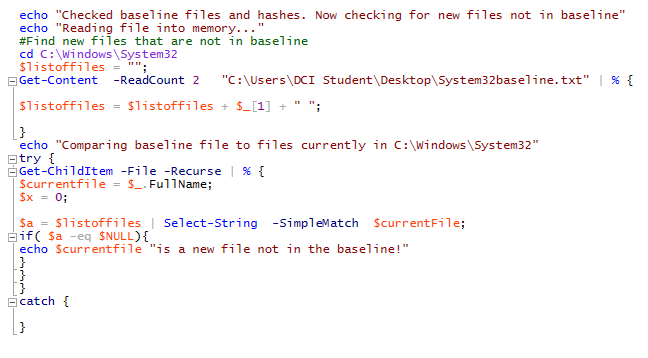
$encodedstring = [convert]::ToBase64String($stringbytes)

$decodedstringbytes = [convert]::FromBase64String($encodedstring)

$decodedstring = [text.encoding]::ASCII.GetString($decodedstringbytes)









**Compare two text files:**

Compare-object (get-content ‘c:\path\file.txt’) (get-content ‘c:\path\file.txt’) | fl | ‘c:\path\file.txt’

**Compare a list of given hashes in $bhfile to hashes in a directory:**

$bhfile = get-content c:\path\givenhashes.text

ls -Force -File C:\Users\pesty\Documents |

%{

$hash = (Get-FileHash $\_.fullname);

$bhfile |

%{

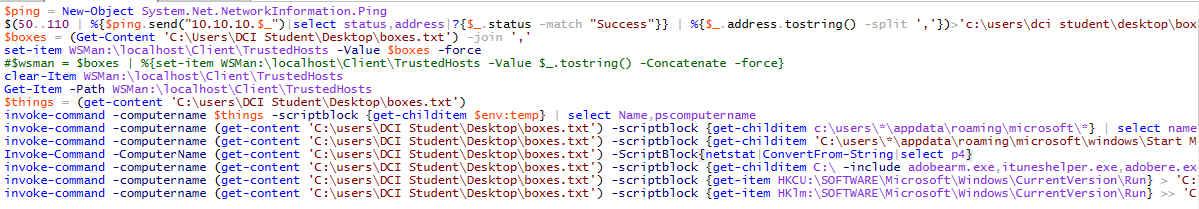
if($\_ -eq $hash.Hash){

echo $hash.Path $hash.Hash;

}

}

}



Schtasks /query /v /fo csv | convertfrom-csv | ?{$\_.TaskName -ne “TaskName”}

