9. Obteniendo HASH de un directorio

Tuesday, November 10, 2020 5:35 PM

Script de PowerShell

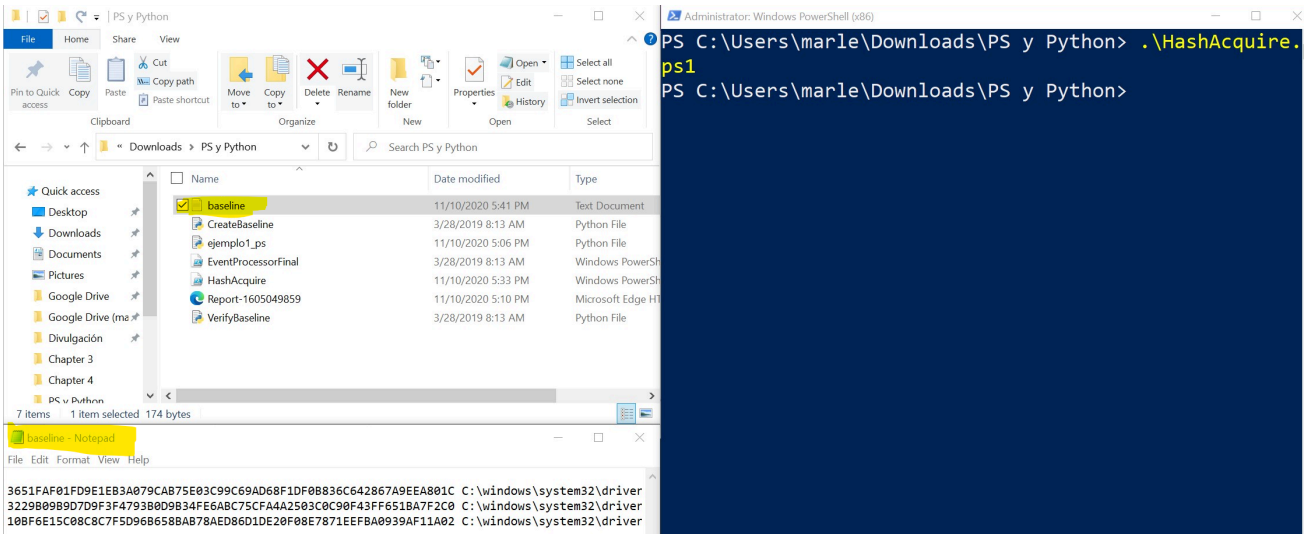
```powershell
<#
.synopsis
Collect Hash and Filenames from specified folder

- User Specifies the target computer
- User Specifies the target folder

The script will produce a simple ascii output file containing
SHA-256Hash and FilePath

.Description
This script collects Hash and Filenames from specified computer and folder

.parameter TargetFolder
Folder where we are going to explore the files and get he hashes

.parameter ResultFile
File where all the hash and complete file names will be explored

.example

HashAcquire
Collects the File Hashes on the target Computer
#>


 # Parameter Definition Section
param(
    [string]$TargetFolder="c:\windows\system32\drivers\",
    [string]$ResultFile="baseline.txt"
)


Get-ChildItem $TargetFolder | Get-FileHash | Select-Object -Property Hash, Path | Format-Table -HideTableHeaders | Out-File $ResultFile -Encoding ascii
```

Ejecución

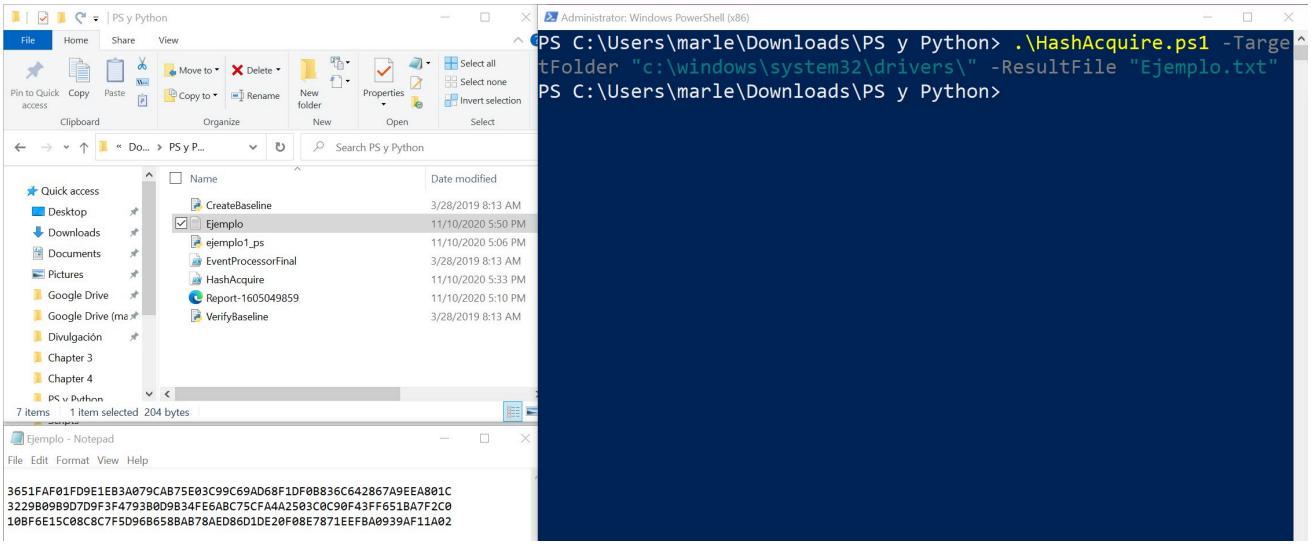
Ayuda

```
PS C:\Users\marle\Downloads\PS y Python> Get-Help .\HashAcquire.ps1

NAME
    C:\Users\marle\Downloads\PS y Python\HashAcquire.ps1

SYNOPSIS
    Collect Hash and Filenames from specified folder

    - User Specifies the target computer
    - User Specifies the target folder

    The script will produce a simple ascii output file containing
    SHA-256Hash and FilePath


SYNTAX
    C:\Users\marle\Downloads\PS y Python\HashAcquire.ps1 [[-TargetFolder] <String>] [[-ResultFile] <String>]
    [<CommonParameters>]


DESCRIPTION
    This script collects Hash and Filenames from specified computer and folder


RELATED LINKS

REMARKS
    To see the examples, type: "get-help C:\Users\marle\Downloads\PS y Python\HashAcquire.ps1 -examples".
    For more information, type: "get-help C:\Users\marle\Downloads\PS y Python\HashAcquire.ps1 -detailed".
    For technical information, type: "get-help C:\Users\marle\Downloads\PS y Python\HashAcquire.ps1 -full".
```

Con parámetros por default



Con parámetros personalizados



HashAcq...

cript de Python

```
'''
Step One Create a baseline hash list of target folder
December 2018, Python Forensics

'''

''' LIBRARY IMPORT SECTION '''

import subprocess          # subprocess library
import argparse            # argument parsing library
import os                  # Operating System Path
import pickle              # Python object serialization
```

```
'''ARGUMENT PARSING SECTION '''

def ValidatePath(thePath):
    ''' Validate the Folder thePath
        it must exist and we must have rights
        to read from the folder.
        raise the appropriate error if either
        is not true
    '''
    # Validate the path exists
    if not os.path.exists(thePath):
        raise argparse.ArgumentTypeError('Path does not exist')

    # Validate the path is readable
    if os.access(thePath, os.R_OK):
        return thePath
    else:
        raise argparse.ArgumentTypeError('Path is not readable')

#End ValidatePath ===================================
```

```
''' Specify and Parse the command line, validate the arguments and return results
info = 'File System Baseline Creator with PowerShell- Version 1.0 December 2018'
parser = argparse.ArgumentParser(info)
parser.add_argument('-b', '--baseline',   required=True,
                    help="Specify the resulting baseline file")
parser.add_argument('-p', '--Path',       type= ValidatePath,
                    required=True, help="Specify the target folder to baseline")
parser.add_argument('-t', '--tmp',        required=True,
                    help="Specify a temporary result file for the PowerShell Scri

args = parser.parse_args()

baselineFile = args.baseline
targetPath   = args.Path
tmpFile      = args.tmp
```

```
''' MAIN SCRIPT SECTION '''
if __name__ == '__main__':

    try:
        ''' POWERSHELL EXECUTION SECTION '''
        print()
        command = "powershell -ExecutionPolicy ByPass -File HashAcquire.ps1 -TargetF
                  targetPath + "\" -ResultFile \"" + tmpFile  +"\""
        print(command)
        powerShellResult = subprocess.run(command, stdout=subprocess.PIPE)
        if powerShellResult.stderr == None:
            input("ya")

            ''' DICTIONARY CREATION SECTION '''
            baseDict = {}

            with open(tmpFile, 'r') as inFile:
                for eachLine in inFile:
                    lineList = eachLine.split()
                    if len(lineList) == 2:
                        hashValue = lineList[0]
```

```
                    fileName   = lineList[1]
                    baseDict[hashValue] = fileName
                else:
                    continue

        with open(baselineFile, 'wb') as outFile:
            pickle.dump(baseDict, outFile)
            print("Baseline: ", baselineFile, " Created with:",
                  "{:,}".format(len(baseDict)), "Records")
            print("Script Terminated Successfully")
    else:
        print("PowerShell Error:", p.stderr)
```

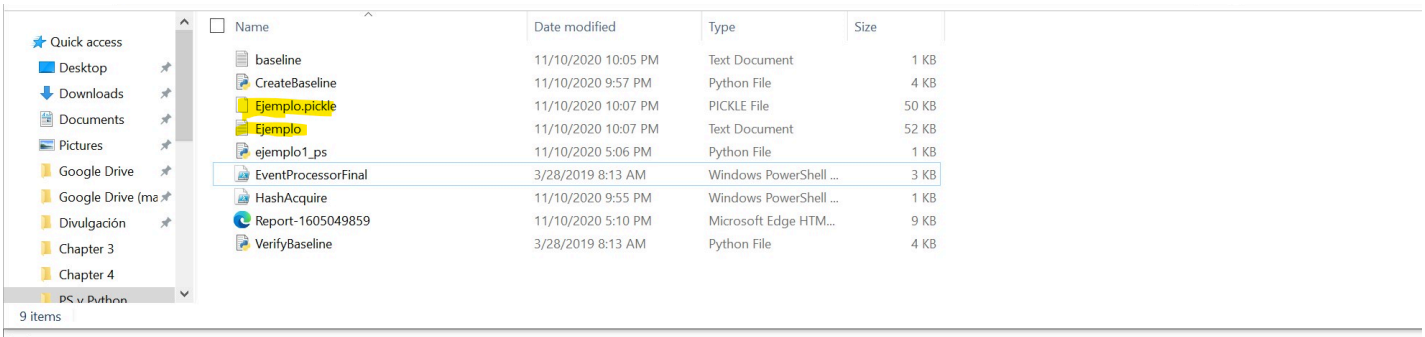
```
except Exception as err:
    print ("Cannot Create Output File: "+str(err))
    quit()
```

Ejecución

Con ayuda

```
C:\Users\marle\Downloads\PS y Python>python CreateBaseline.py -h
usage: File System Baseline Creator with PowerShell- Version 1.0 December 2018 [-h] -b BASELIN
                                                                                TMP

optional arguments:
  -h, --help            show this help message and exit
  -b BASELINE, --baseline BASELINE
                        Specify the resulting baseline file
  -p PATH, --Path PATH  Specify the target folder to baseline
  -t TMP, --tmp TMP     Specify a temporary result file for the PowerShell Script
```

Con argumentos



Administrator: Command Prompt

```
C:\Users\marle\Downloads\PS y Python>CreateBaseline.py -b Ejemplo.pickle -p C:/Windows/System32/dri
mplo.txt

powershell -ExecutionPolicy ByPass -File HashAcquire.ps1 -TargetFolder C:/Windows/System32/drivers/
 Ejemplo.txt
ya
Baseline:  Ejemplo.pickle  Created with: 467 Records
Script Terminated Successfully
```

Ejemplo - Notepad

File Edit Format View Help

```
E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855 C:\Windows\System32\drivers\!
3D6E0579821BFA91B7F0A6E6DDC6E03BD3389202AD1A079B825D18D2A76250A0 C:\Windows\System32\drivers\1394ohci.sys
5A29D80AF47D08998125CB81BC1D4E84093291A74DE422B63F7BBDA7BDE95311 C:\Windows\System32\drivers\3ware.sys
916C2525BF27CEC3C29D51FC0AE67E6CB8D73D5A323118145DC0BAD00B592609 C:\Windows\System32\drivers\acpi.sys
1D576471A8035AD3FF5B0616F47B79E43AA367ECDF009D7CADDA0F11F13A1345 C:\Windows\System32\drivers\AcpiDev.sys
```

```
34856C805B67F3EE4ABFD81B61879112344C343BC7E76A7A466FAD276E0E5165 C:\Windows\System32\drivers\acpiex.sys
3A9A504797FD22BB5447BB36597D5001320ABC0D4A1853D478C038EAC6847913 C:\Windows\System32\drivers\acpipagr.sys
1E7C5FADA2486EE31289A4BEFB70AEA173190671C64995441651903CF31E5033 C:\Windows\System32\drivers\acpipmi.sys
72D35F5DB8714D38E4050A7F7A457C4AD99E3EA212040704F1C1ECBB70E865E9 C:\Windows\System32\drivers\acpitime.sys
897A9C367AD464F0CB4DEB4E53CD788D75673B0F84241D5CEE2DBE64BE038818 C:\Windows\System32\drivers\Acx01000.sys
B83072D77685F973701EC6629D8AC2626FDEFD657A4DB9AA7D532960A29FC67C C:\Windows\System32\drivers\adp80xx.sys
C50D7B309C446058140F0A714F9267B58B144E63AE845B09E51572D09960236B C:\Windows\System32\drivers\afd.sys
D83B788A59F84071260695A6C71ACF6AD4760C11F0E249E266A666E4648B3C9A C:\Windows\System32\drivers\afunix.sys
30BB5CA062AC3DD7478720FDB92EED76BDD2E893732AE481D01CF5C862A8C1EB C:\Windows\System32\drivers\agilevpn.sys
BC3A37CEB86D2B1970A9F4ABA31F958A1FE07C246F4F127334BB19719893ED91 C:\Windows\System32\drivers\ahcache.sys
46A53925BAA34FA9D87E7C3157504A4557D81CD8B8608E7AB6CAF02F482F7792 C:\Windows\System32\drivers\amdgpio2.sys
387811D57DEF06C9736D9F0BAB0DFB0F83DBAB19E5489BF9A6DCDCBD682DD8FE C:\Windows\System32\drivers\amdi2c.sys
1DE9419C351546F4B8747AA46422311F8D1610CCA4FD050D2E2D63B6A5A839C3 C:\Windows\System32\drivers\amdk8.sys
DBAA893F1889C5B433786A1F0A5491389A8ED465E1BF2E9C486605F0D4F054CF C:\Windows\System32\drivers\amdppm.sys
D5231F97E5432234A8A19904E59C324E825AF04881AA195C19CCC9E6A7684B14 C:\Windows\System32\drivers\amdsata.sys
71C7E7E5AA74596A6725D8F70F1DE9A0C63D3C3E120D9CCF8A508854AC340A23 C:\Windows\System32\drivers\amdsbs.sys
1FC1D4287DB56A387BDF917C0CB3BFC30CA5D792A350E2EDBBDDEBF8127E1AF9 C:\Windows\System32\drivers\amdxata.sys
44B1727270D03795B639BC42B879BB3268BA6111E00DA228191CAD8E9BD90D72 C:\Windows\System32\drivers\appid.sys
1E8729E07A039CB6D0FC911911FE8E5BB1DA977181703B48C9F95DE8B1756629 C:\Windows\System32\drivers\applockerfltr.sys
15763D9C6EC89DBA974C41BDB362D82F23A550FF78943D8B2E158518DB899A14 C:\Windows\System32\drivers\AppVStrm.sys
FCF60C3C9A5A864AAAFBEF0FC3BE098EF3B28F89B92CCDAF7902D5022E9BB2CF C:\Windows\System32\drivers\AppvVemgr.sys
E073804F721AD6067D54C6604945C5365E45935C76AC5EC5DBC1ED1F91CA1878 C:\Windows\System32\drivers\AppvVfs.sys
96DCA25AE619F38640B22702A10BC3191626F3A36DE0E1B0EDA3B079EA9DEB24 C:\Windows\System32\drivers\arcsas.sys
14985D6D2D52689C1B012F64ED0D7C9C5F6BADB51C4528BF6456D3EAE2FE69A7 C:\Windows\System32\drivers\asyncmac.sys
640B9E84035441BF4B116A9D6A31B457F6A07EEFF6E7CAD0FE6E688B219F0275 C:\Windows\System32\drivers\atapi.sys
20E6C7CF603585057F2DDC859C4F4193A8C5B8B73AB444E876DBAFF6B60602EA C:\Windows\System32\drivers\ataport.sys
C2D9CF2189C10A837A98CFF9D878AD1864C3DD79AF41EDE472C2B8488FF9A39E C:\Windows\System32\drivers\ax88179_178a.sys
7D06B6499FE915480DF4DAD658281C8B85F7AD71F49B089A270AE0B45713F2E9 C:\Windows\System32\drivers\bam.sys
F6FCD2CD58D1079EA276F512811531FA202163DE99E82E4073474D664DF7AA24 C:\Windows\System32\drivers\battc.sys
9AD12E18A042C5B8EFB19297BC2E7BD1FEF75A138FEFB64C6BF0261FD3E53AB1 C:\Windows\System32\drivers\bcmfn2.sys
121AF03BBE6ECC1622C2540805A30AE9555EB5D5FE25B55939C045ECE7FC37EB C:\Windows\System32\drivers\beep.sys
071E0B7204947AC13CAF3994267D0ADA31EA862F62BD3A0BC98F6A96C29EAC61 C:\Windows\System32\drivers\bindflt.sys
78312CD9CADB24DEE22413F2F0642A8FF10C91200D7ABB17C1C2D2D7B30E18D8 C:\Windows\System32\drivers\bowser.sys
A3AA5D51E34657479CFCDC3DBB7821B7255F7CB57D5686B7F709A7953AD537EB C:\Windows\System32\drivers\bridge.sys
ADEFD74835E3077DE2D58430955EAC9A7CE4FB53BB4976AEE286D10DC463FBC9 C:\Windows\System32\drivers\BtaMPM.sys
0442A18BBED4E323265C66561C8F8C171D8E934E9089C12B94D1DFDBB057B737 C:\Windows\System32\drivers\BthA2dp.sys
3F0BB374C945608F65317F102575DC13F5F396AE81BE94E982F62E50A5DE91BF C:\Windows\System32\drivers\bthenum.sys
91C72C54142A0D4E5A5F33268850CEB8315AA30C2F0B74A9FFA962887ABAC797 C:\Windows\System32\drivers\BthHfEnum.sys
359D002E3DF422F79C3923A14D22F02D0719ED7AA61D2C0FD98BF898E374A72F C:\Windows\System32\drivers\BthMini.SYS
9412DC92F16C0B8A937D6FB1AD83D7169F4EC0F08FAE0E2B244346428CE99EE1 C:\Windows\System32\drivers\bthmodem.sys
C39E471BD757CA12635F283FFC44079894477 39D36830E81E8DD3E63E363B3D7 C:\Windows\System32\drivers\bthpan.sys
D9983ABF59B55DE955A3ADE7292DA7152EDD93B44E8CD69D3DE18298ED6669E9 C:\Windows\System32\drivers\bthport.sys
07BBED68F17FABF4E3040B07BD39CF1566C16A8B3C2CC3C8B95CC7C589128A1C C:\Windows\System32\drivers\BTHUSB.SYS
8DE3B7C87D88CF375417355A7C5052B2DE38805B563D61D0E483DB4AD96BD741 C:\Windows\System32\drivers\bttflt.sys
16A900FBAB30D008F01F4CAF96347BF313D9D13C7FF430249A0BF4322534CB18 C:\Windows\System32\drivers\buttonconverte...
```

CreateBa...

Comprobar los HASH

```
'''
Step Twp Verify a baseline hash list against a target folder
December 2018, Python Forensics

'''
''' LIBRARY IMPORT SECTION '''

import subprocess       # subprocess library
import argparse         # argument parsing library
import os               # Operating System Path
import pickle           # Python object serialization

'''ARGUMENT PARSING SECTION '''

def ValidatePath(thePath):
    ''' Validate the Folder thePath
        it must exist and we must have rights
```

```
        to read from the folder.
        raise the appropriate error if either
        is not true
    '''
    # Validate the path exists
    if not os.path.exists(thePath):
        raise argparse.ArgumentTypeError('Path does not exist')

    # Validate the path is readable
    if os.access(thePath, os.R_OK):
        return thePath
    else:
        raise argparse.ArgumentTypeError('Path is not readable')

#End ValidatePath ===================================
```

```
''' Specify and Parse the command line, validate the arguments and return results'''

info = 'File System Baseline Validation with PowerShell- Version 1.0 December 2018'
parser = argparse.ArgumentParser(info)
parser.add_argument('-b', '--baseline',   required=True,
                    help="Specify the source baseline file to verify")
parser.add_argument('-p', '--Path',       type= ValidatePath,
                    required=True, help="Specify the target folder to verify")
parser.add_argument('-t', '--tmp',        required=True,
                    help="Specify a temporary result file for the PowerShell Script")

args = parser.parse_args()

baselineFile = args.baseline
targetPath   = args.Path
tmpFile      = args.tmp
```

```
def TestDictEquality(d1,d2):
    """ return True if all keys and values are the same
        otherwise return False
    """
    if all(k in d2 and d1[k] == d2[k] for k in d1):
        if all(k in d1 and d1[k] == d2[k] for k in d2):
            return True
        else:
            return False
    else:
        return False

    '''
    return all(k in d2 and d1[k] == d2[k]
               for k in d1) \
        and all(k in d1 and d1[k] == d2[k]
               for k in d2)
    '''
```

```
def TestDictDiff(d1, d2):
    """ return the subset of d1 where the keys don't exist in d2 o
        the values in d2 are different, as a dict """
```

```
    diff = {}

    for k,v in d1.items():
        if k in d2 and v in d2[k]:
            continue
        else:
            diff[k+v] = "Baseline Missmatch"

    return diff


''' MAIN SCRIPT SECTION '''
if __name__ == '__main__':

    try:
        ''' POWERSHELL EXECUTION SECTION '''
        print()
        command = "powershell -ExecutionPolicy ByPass -File HashAc
                  " -TargetFolder \""+ targetPath+"\" -ResultFile
                  "\""
        print(command)
        print()
        powerShellResult = subprocess.run(command, stdout=subproce
        if powerShellResult.stderr == None:

            ''' DICTIONARY CREATION SECTION '''
            # Load in the baseline dictionary

            with open(baselineFile, 'rb') as baseIn:
                baseDict = pickle.load(baseIn)

            # Create a new dictionary for the target folder
            newDict  = {}


            with open(tmpFile, 'r') as inFile:
                for eachLine in inFile:
                    lineList = eachLine.split()
                    if len(lineList) == 2:
                        hashValue = lineList[0]
                        fileName  = lineList[1]
                        newDict[hashValue] = fileName
                    else:
                        continue

            ''' DICTIONARY TEST SECTION '''
            if TestDictEquality(baseDict, newDict):
                print("No Changes Detected")
            else:
                diff = TestDictDiff(newDict, baseDict)
                print(diff)
```

```
        else:
            print("PowerShell Error:", p.stderr)

    except Exception as err:
        print ("Cannot Create Output File: "+str(err))
        quit()
```

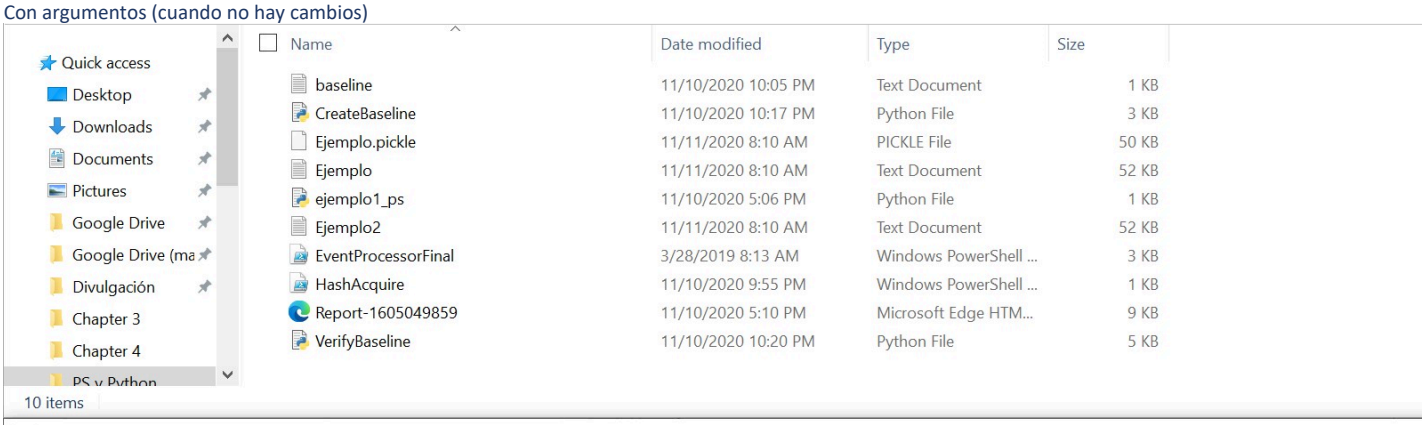
Ejecución

Con ayuda

Administrator: Command Prompt

```
C:\Users\marle\Downloads\PS y Python>VerifyBaseline.py -h
usage: File System Baseline Validation with PowerShell- Version 1.0 December 201


optional arguments:
  -h, --help            show this help message and exit
  -b BASELINE, --baseline BASELINE
                        Specify the source baseline file to verify
  -p PATH, --Path PATH  Specify the target folder to verify
  -t TMP, --tmp TMP     Specify a temporary result file for the PowerShell Scrip
```

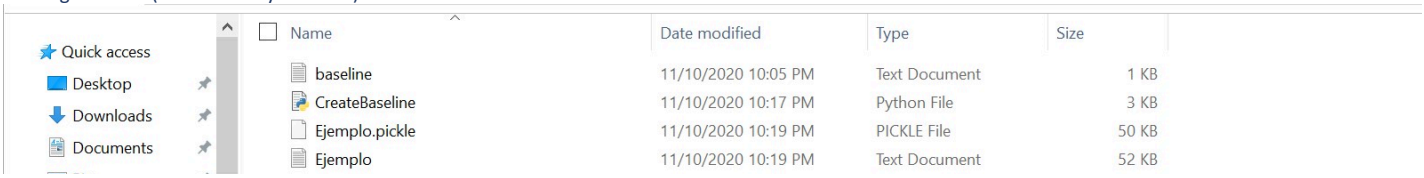
Con argumentos (cuando no hay cambios)

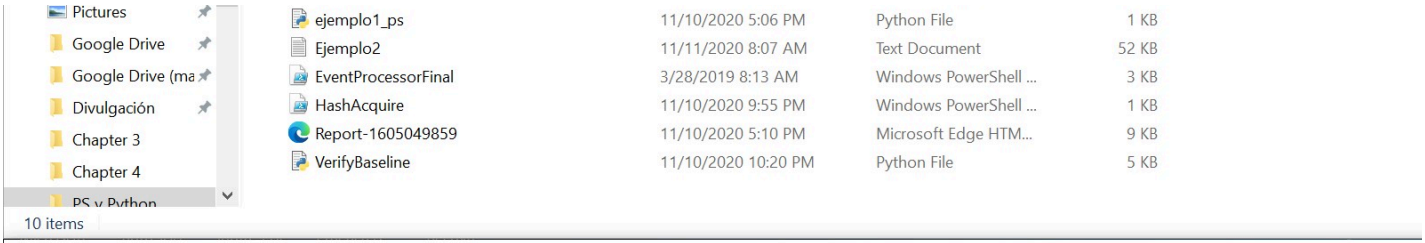


Administrator: Command Prompt

```
C:\Users\marle\Downloads\PS y Python>VerifyBaseline.py -b Ejemplo.pickle -p "C:\
jemplo2.txt

powershell -ExecutionPolicy ByPass -File HashAcquire.ps1 -TargetFolder "C:\Windo
le "Ejemplo2.txt"

No Changes Detected

C:\Users\marle\Downloads\PS y Python>_
```

Con argumentos (cuando si hay cambios)





Administrator: Command Prompt

```
C:\Users\marle\Downloads\PS y Python>VerifyBaseline.py -b Ejemplo.pickle -p "C:\
jemplo2.txt

powershell -ExecutionPolicy ByPass -File HashAcquire.ps1 -TargetFolder "C:\Windo
le "Ejemplo2.txt"

{'35C819F74CE7E93A7E14482D53729653F88277FD93D452F73A6A635CB8B98229C:\\Windows\\S
': 'Baseline Missmatch', '1E8729E07A039CB6D0FC911911FE8E5BB1DA977181703B48C9F95D
2\\drivers\\applockerfltr...': 'Baseline Missmatch', '16A900FBAB30D008F01F4CAE96
2534CB18C:\\Windows\\System32\\drivers\\buttonconvert...': 'Baseline Missmatch',
9D446BFE7F2BFCD15718A5E09D9776E0A562C:\\Windows\\System32\\drivers\\clfs.sys': '
6F9A81FBFF61D2AC0058EBBA6397DBEA9AF585048026210B211DDB1EC:\\Windows\\System32\\d
ssmatch', '9E829A36C2B342336D2F20521804A5B6140C2798D99181B8C5AD44FBCDC2FB09C:\\W
2HostContr...': 'Baseline Missmatch', '88D98AF6B0725667A59D37D4F3F7C06D736FB6795
dows\\System32\\drivers\\devauthe.sys': 'Baseline Missmatch', 'A137861C52A52A5CA
```

```
dows\\System32\\drivers\\SurfaceTouchS...': 'Baseline Missmatch', 'E667D58DC7B2A
F31908004FF780113BCDFC:\\Windows\\System32\\drivers\\tcpip.sys': 'Baseline Missm
B44CF5B089BE6BD0EAC48637B109972B29B0FE5F7DC:\\Windows\\System32\\drivers\\TeeDri
ch', '3626760AEE42EE36E047DA6899A81E0646DFBA344A234270EAE5D635F049BE37C:\\Window
cpiCl...': 'Baseline Missmatch', '9000E15B7ABA7E7407FDE2A6EC025E50FCF838ADD66A96
\System32\\drivers\\UevAgentDrive...': 'Baseline Missmatch', 'CB8501336C0D979DA8
2F19B0B4937CB6BAC:\\Windows\\System32\\drivers\\vhdmp.sys': 'Baseline Missmatch'
8CEC7501F795DE8FA86857790F4CCD5AF7C18C:\\Windows\\System32\\drivers\\WdmCompanio
'64EEB8093BA2590E83D83C5AF7C2A025B88AF5681143BCA83671104266FEEA99C:\\Windows\\Sy
...': 'Baseline Missmatch', '090C4CE6B76B3904B5AE73E4F1EEBCE619194C358874D758453
em32\\drivers\\WindowsTruste...': 'Baseline Missmatch', 'C815326E2CC17D56E6BEEF9
F7F3B2C91DFC:\\Windows\\System32\\drivers\\WirelessKeybo...': 'Baseline Missmatc
12B9A3AE9D4680E59476D1286F6C0767C73E3DAC:\\Windows\\System32\\drivers\\xboxgip.s
82CA4D087739CE004D1033970A036616A1EEBA618D1033D45FEC895D08D0C6C:\\Windows\\Syste
'Baseline Missmatch'}
```

VerifyBas...