Forensics : Drawings on the walls

Description : My friend started having problems with his head and he began to draw some nonsense on the walls.

Can you make out thes notes ?

Attachment : **memory.tgz**

Solutions :

Download and extract the attachment file «memory.tgz» and we get a file called «memory.dmp».

Using «file» on it reveal the file type.

```
root@kali:~# file memory.dmp
memory.dmp: MS Windows 64bit crash dump, full dump, 524288 pages
```

As we can see, the file seem to be a Windows 64bit crash dump. So we will use the tool «**Volatility**» and get the right OS used with the plugin «**imageinfo**».

```
root@kali:~# volatility -f memory.dmp imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search…
WARNING : volatility.debug    : Alignment of WindowsCrashDumpSpace64 is too
small, plugins will be extremely slow
          Suggested Profile(s) : Win8SP0x64, Win10x64_17134, Win81U1x64,
Win10x64_10240_17770, Win2012R2x64_18340, Win10x64_14393, Win10x64,
Win2016x64_14393, Win10x64_16299, Win2012R2x64, Win2012x64,
Win8SP1x64_18340, Win10x64_10586, Win8SP1x64, Win10x64_15063
(Instantiated with Win10x64_15063)
                     AS Layer1 : SkipDuplicatesAMD64PagedMemory (Kernel AS)
                     AS Layer2 : WindowsCrashDumpSpace64 (Unnamed AS)
                     AS Layer3 : FileAddressSpace (/root/Téléchargements/memory.dmp)
                      PAE type : No PAE
                           DTB : 0x187000L
```

```
                      KDBG : 0xf80002c010a0L
          Number of Processors : 2
     Image Type (Service Pack) : 1
               KPCR for CPU 0 : 0xfffff80002c02d00L
               KPCR for CPU 1 : 0xfffff880009ef000L
            KUSER_SHARED_DATA : 0xfffff78000000000L
          Image date and time : 2020-02-29 01:21:09 UTC+0000
    Image local date and time : 2020-02-29 01:21:09 +0000
```

I started the challenge by using the profile «**Win10x64**» but i get some error later. Someone told me it was «**Win7SP2x64**». So let's using this profile and run «**volatility**» to get the process list with the plugin «**pslist**».

In the output i only will show all interesting process, or the result will take three pages…

```
root@kali:~# volatility -f memory.dmp pslist --profile=Win7SP1x64
Volatility Foundation Volatility Framework 2.6
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start
------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------
…

0xfffffa8001ca8b30 notepad++.exe        2836   3764      7      130      1      0 2020-02-24 14:39:32
UTC+0000
0xfffffa8001a5f060 mspaint.exe          2080   1392      7      146      1      0 2020-02-28 14:50:41
UTC+0000
0xfffffa8001ca3060 svchost.exe          3644    472      7      109      0      0 2020-02-28 14:50:41
UTC+0000
0xfffffa8001cbab30 mspaint.exe          2804   1392      7      132      1      0 2020-02-28 15:12:05
UTC+0000
0xfffffa8001dd8b30 mspaint.exe          3416   1392      6      128      1      0 2020-02-28 15:12:07
UTC+0000
0xfffffa800274d060 mspaint.exe           704   1392      6      129      1      0 2020-02-28 15:12:09
UTC+0000
0xfffffa8001bfe060 mspaint.exe          2964   1392      6      130      1      0 2020-02-28 15:14:03
UTC+0000
0xfffffa8002098060 mspaint.exe          2124   1392      6      129      1      0 2020-02-28 15:14:28
UTC+0000
0xfffffa8001c36060 svchost.exe          3504    472      5       65      0      0 2020-02-29 01:15:26
UTC+0000
0xfffffa8001df3060 LogonUI.exe           736    520      8      172      1      0 2020-02-29 01:21:06
UTC+0000
```

As i never used «**Volatility**» and «**GIMP**» before, i started to look on google some writeup about those tools. I based my approche with this writeup bellow :

Source : https://www.rootusers.com/google-ctf-2016-forensic-for1-write-up/

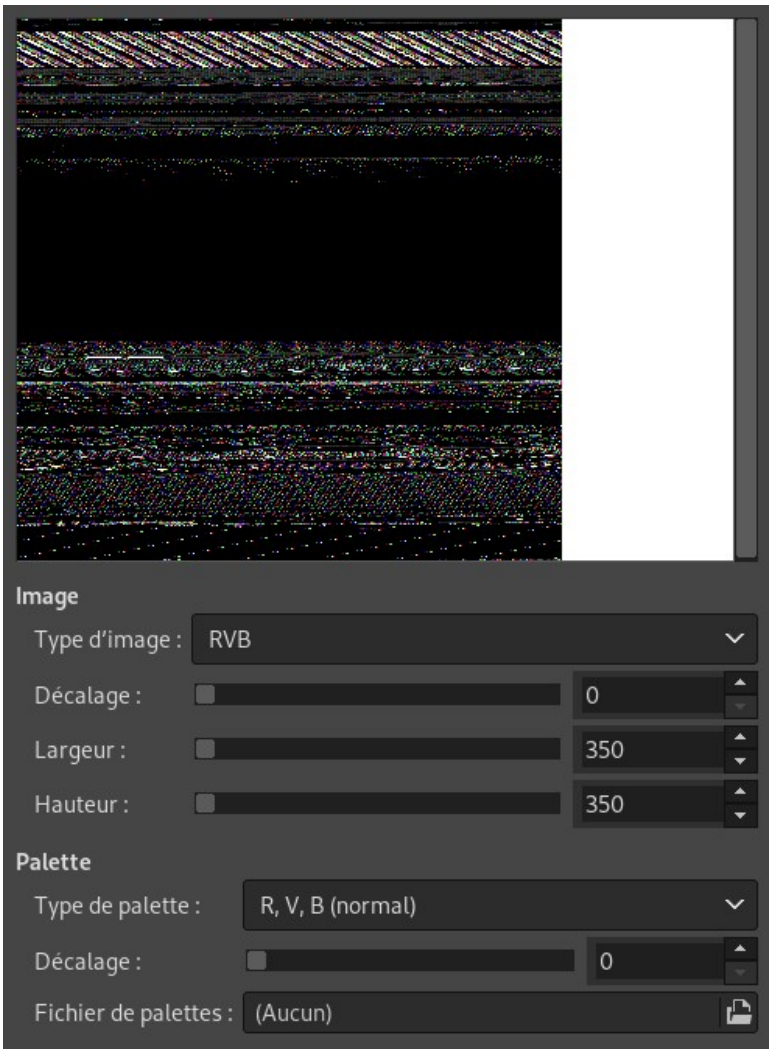
Now we will dump the memory of the process «**mspaint.exe**» with volatility and his plugin «memodump», as there is 6 process of it, we will dump all of them.

```
root@kali:~# mkdir dump
root@kali:~# volatility -f memory.dmp memdump -p 2080 --dump-dir dump/ --
profile=Win7SP1x64
Volatility Foundation Volatility Framework 2.6
************************************************************************
Writing mspaint.exe [  2080] to 2080.dmp
```

Now i've make a copy of the file from «**2080.dmp**» to «**2080.data**» and open the copy with **GIMP**.

```
root@kali:~/dump# file 2080.dmp
2080.dmp: data
root@kali:~/dump# cp 2080.dmp 2080.data
root@kali:~/dump# gimp 2080.data
```

Once opened into gimp, we have something like that.



Now it’s time to analyse all of them, i am a newbie in GIMP it was a pain for me. Took me few hours, we need to change the three value «**Offset**» «**Width**» and «**Height**».

When i solved this challenge, i find 4 part of flag, inside the pid 2080. But now actually writing this writeup, i dont know why but i dont find them, but i find «Rick Astley», i think they add few time in every files all the flag, and we need to find them all.

Here is how i solve it now.

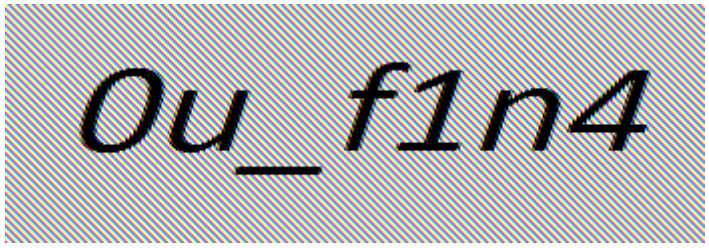
Mspaint.exe PID 2080

Offset : 0
Width : 2250
Height : 21158

First part of the flag : Aero{g00dj0b_y

Mspaint.exe PID 704

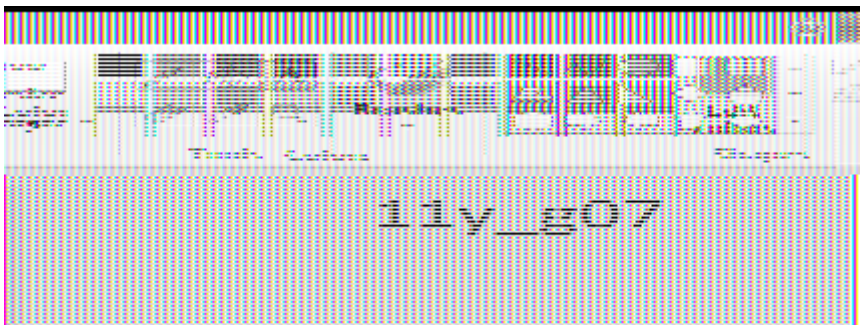
Offset : 277729297
Width : 1475
Height : 500



Second part of the flag : 0u_f1n4

Mspaint.exe 2080

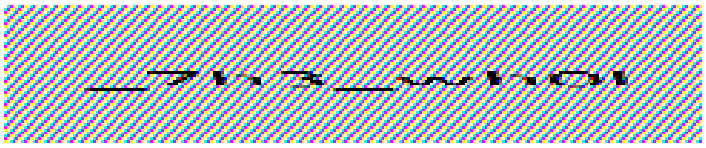
Offset : 0
Width : 1984
Height : 21158



Third part of the flag : 11y_g07

Mspaint.exe 2080

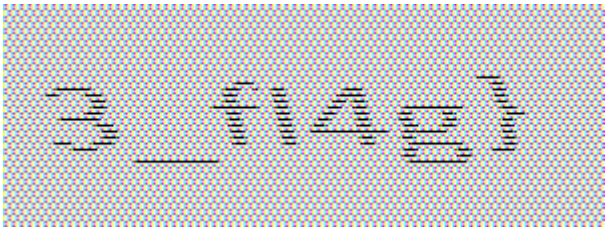
Offset :
Width :
Height :



Fourth part of the flag : _7h3_wh0l

Mspaint.exe 2080

Offset : 0
Width : 2702
Height : 21158



Fifth part of the flag : 3_fl4g}

Final Flag : Aero{g00dj0b_y0u_f1n411y_g07_7h3_wh0l3_fl4g}

Bonus – Unintend Way

Once we get a part of flag through GIMP, we can use «**strings**» combined with «**grep**» for find all the flag. Then we only need to guess his format, and which one of the list is correct.

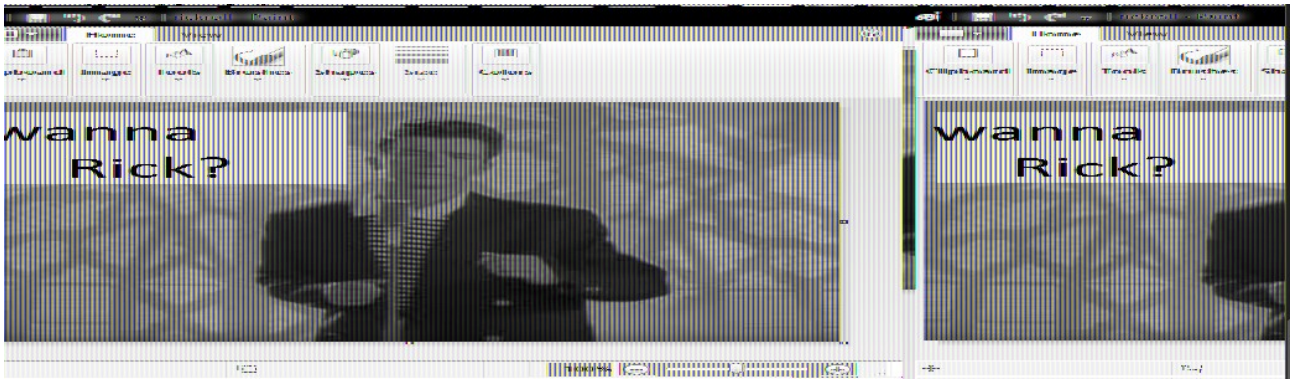
```
root@kali:~/Téléchargements# strings memory.dmp | grep "g00dj0b"
.g00dj0b_y
g00dj0b_y
g00dj0b_y0u_f1n411y_g07_7h3_wh0l3_fl4g
g00dj0b_y
g00dj0b_y
g00dj0b_y
g00dj0b_y
g00dj0b_y
Bg00dj0b_y
g00dj0b_y
g00dj0b_y0u_f1n411y_g07_7h3_wh0l3_fl4g_
```

Bonus – Wanna Rick ?

As i said previously, while you resolve this challenge throught GIMP, you will be a lot trolled by the famous «Rick Roll» of Rick Astley. One of those pictures bellow.

Mspaint.exe PID 2080

Offset : 0
Width : 1528
Height : 3200



Bonus – Binwalk

Using «Binwalk» against «memory.dmp» and we can extract an iso file.

```
root@kali:~# binwalk -e memory.dmp

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
111432        0x1B348         Intel x86 or x64 microcode, pf_mask 0x00, 1AE4-03-08, size 33288
163840        0x28000         Microsoft executable, portable (PE)
380247        0x5CD57         Cisco IOS microcode, for "S"
380294        0x5CD86         Cisco IOS microcode, for "M"
467069        0x7207D         Certificate in DER format (x509 v3), header length: 4, sequence length:
2916
470084        0x72C44         Certificate in DER format (x509 v3), header length: 4, sequence length:
1226
664610        0xA2422         Copyright string: "Copyright 1985-1998,Phoenix Technologies Ltd.All
rights reserved."
827522        0xCA082         Copyright string: "Copyright (C) 2003-2014  VMware, Inc."
827561        0xCA0A9         Copyright string: "Copyright (C) 1997-2000  Intel Corporation"
950876        0xE825C         ISO 9660 Boot Record,

root@kali:~# cd _memory.dmp.extracted/
root@kali:~/_memory.dmp.extracted# ls
E825C.iso
root@kali:~/_memory.dmp.extracted# file E825C.iso
E825C.iso: ISO 9660 CD-ROM filesystem data
```

Bonus – Foremost

Using «**Foremost**» against «**memory.dmp**» and we can extract many files.

```
root@kali:~# foremost memory.dmp
Processing: memory.dmp
|***foundat=
foundat=
foundat=
foundat=
foundat=
foundat=
foundat=
foundat=
foundat=
foundat=
foundat=
foundat=
```

```
*foundat=����3�;��B
**foundat=��~
*WMV err num_header_objs=-1073683314 headerSize=1283430543591726241
foundat=����3�;��B
**foundat=_@
*foundat=
**foundat=
*WMV err num_header_objs=-1073683570 headerSize=1283430435463824309
WMV err num_header_objs=-1610565708 headerSize=1284296802347103227
**WMV err num_header_objs=-1610599280 headerSize=1283866582801590751
******|
```

Files listing using «tree» commands. I only put a part of the result, or my writeup will be «the bible version 2.0».

```
root@kali:~/output# tree
.
├── audit.txt
├── avi
│   ├── 00057324.avi
│   ├── 00382064.avi
│   ├── 00468264.avi
│   ├── 00468266.avi
│   ├── 00548841.avi
│   ├── 00548842.avi
│   ├── 00548843.avi
│   ├── 00548845.avi
│   ├── 00548846.avi
│   ├── 00548847.avi
│   ├── 00753649.avi
│   ├── 00864635.avi
│   ├── 00876026.avi
│   ├── 00881732.avi
│   ├── 01065254.avi
│   ├── 01065255.avi
│   ├── 01185246.avi
│   ├── 01276754.avi
│   ├── 01345348.avi
│   ├── 01385845.avi
│   ├── 01403777.avi
│   ├── 01403778.avi
│   ├── 01403779.avi
│   ├── 01403781.avi
│   ├── 01405807.avi
│   ├── 01552706.avi
│   ├── 01688849.avi
│   ├── 01717848.avi
│   ├── 01719305.avi
│   ├── 01743329.avi
│   ├── 01743335.avi
```

```
│   ├── 02066490.avi
│   ├── 02066491.avi
│   ├── 02066492.avi
│   ├── 02066494.avi
│   ├── 02122424.avi
│   ├── 02426383.avi
│   ├── 02478099.avi
│   ├── 02665738.avi
│   ├── 02811315.avi
│   ├── 02917352.avi
│   ├── 03172325.avi
│   ├── 03428655.avi
│   ├── 03472401.avi
│   ├── 03647191.avi
│   ├── 03711897.avi
│   ├── 03808299.avi
│   ├── 03880581.avi
│   ├── 03971691.avi
│   └── 04023117.avi
├── bmp
│   ├── 00807638_1.bmp
│   ├── 00807638.bmp
│   ├── 01113592.bmp
│   ├── 01318431.bmp
│   ├── 01481632.bmp
│   ├── 01481634.bmp
│   ├── 01481637.bmp
│   ├── 01481638.bmp
│   ├── 01714744.bmp
│   ├── 01769064.bmp
│   ├── 02225565.bmp
│   ├── 02487171.bmp
│   ├── 02487173.bmp
│   ├── 03394328.bmp
│   ├── 03442976.bmp
│   ├── 03556232.bmp
│   ├── 03815152.bmp
│   ├── 03815154.bmp
│   ├── 03815156.bmp
│   └── 03815157.bmp
├── dll
│   ├── 00005337.dll
│   ├── 00037112.dll
│   ├── 00040200.dll
│   ├── 00042722.dll
│   ├── 00044784.dll
│   ├── 00045288.dll
│   ├── 00070418.dll
│   ├── 00074208.dll
```