PV204 Security Technologies

6th assignment – Part2 – Blackbox maware analysis

Supervisor: Dr. Vít Bukač

Student: Pedro Gomes, 490830

Malware 1

1.1. Description of external behavior (e.g., what windows are shown to the user,if any).

This malware seems to be more stealth than any other malware we analyzed during the lab. The two first remarks is that no window opens up, and the malware executable is deleted after a certain set of operations described below.

1.2. Created, modified and deleted files. Emphasize what files are critical for the malware. Focus on distinguishing between original malware files and operating system files.

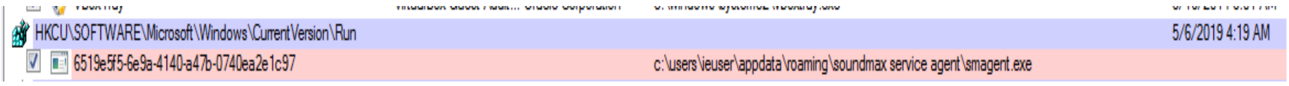
It seems that the malware starts by creating the following folder: *C:\Users\IEUser\AppData\Roaming\SoundMAX service agent* , within this folder there is an executable file called: *smagent*

After make some investigations[1], we can see that, the smagent makes part of the audio drivers for Asus machines. This can either be a coincidence or the malware is really doing its job correctly, because I have an Asus machine. One way or another, the file is clearly malicious, because I was able to see that malware.exe launched it, with process explorer. Furthermore, this folder did not exist in the none-infected version of the virtual machine.

After being launched, the smagent opens up an instance of internet explorer, and closes. The IE instance remains open.

1.3 Persistence methods. How malware makes sure it is executed again after reboot.

I had to reboot the machine in order to see some persistence evidence. After doing that with the help of autoruns, I was able to see that:

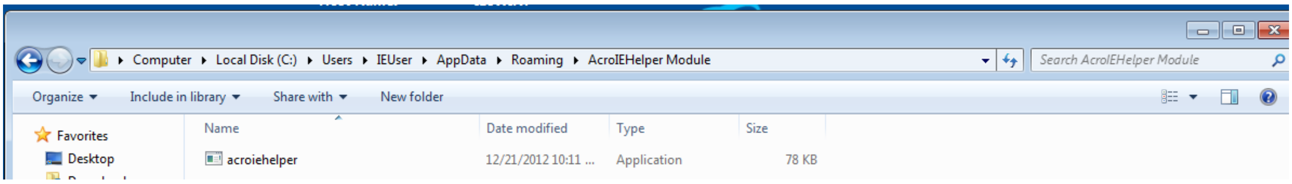


```
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run                                          5/6/2019 4:19 AM
☑ 6519e5f5-6e9a-4140-a47b-0740ea2e1c97        c:\users\ieuser\appdata\roaming\soundmax service agent\smagent.exe
```

Illustration 1: Persistence Evidence

We can see that the path of the file to be executed is the smagent.exe, that was already mentioned during the execution of the malware.

If we run malware1 a couple of more times, we can see that it creates another executable within the exact same location of the above picture, but the folder that has the executable file and the executable file itself have a different name:



Computer ▸ Local Disk (C:) ▸ Users ▸ IEUser ▸ AppData ▸ Roaming ▸ AcroIEHelper Module

Name	Date modified	Type	Size
acroiehelper	12/21/2012 10:11 ...	Application	78 KB

Illustration 2: alternative created executable file

1.4. Network communication. With whom and how is malware trying to communicate.

I was not able to detect any behavior of the malware that leaded me to think that it is trying to communicate with some online resource.

Using wireshark, it was possible to see some M-search packets, but I had no way to confirm that those packets were created by the malware. M-search packets usually have the functionality to check if the router the machine is connected to supports Upnp, allowing the malware to infect machines in the network that support it too, and hence add a couple more zombies to some botnet. As I said, this is most likely not what the malware is doing, but this was the only suspicious network activity I found.

There are also a lot of IPv6 requests, I am not sure this is normal, because I cannot see my IPv6 configured in the virtual machine.

1.5. Defense mechanisms used by the malware to prevent the analysis. Approaches how you were able to circumvent these mechanisms.

I think that the malware tries to hide itself, faking an authentic process of the machine, in this case the smagent.exe

It also deleted itself, some time after its execution.

I simply noticed that smagent.exe was not a normal file and extracted its location, finally, I was able to see what the fake smagent does (opens instance of internet explorer)

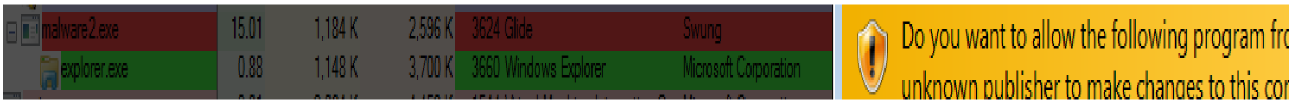
1.6. Conclusion

I was able to detect what the malware does after its execution and how it gains persistence. However, I was not able to find any malicious network activity directly related with the malware. Furthermore, I still do not understand what the malware wants to do exactly.

Malware 2

I was not able to gather a lot of information about the behavior of this malware.

First of all, the malware2 deletes itself like malware1. The malware requires administrator privileges to run, and while the prompt to be an administrator shows up, we can see some more information about the malware:



```
malware2.exe    15.01   1,184 K   2,596 K   3624 Glide              Swung
  explorer.exe   0.88   1,148 K   3,700 K   3660 Windows Explorer   Microsoft Corporation
```

Do you want to allow the following program fr
unknown publisher to make changes to this cor

Illustration 3: Malware2 starting up

It shows us that the supposed name of the company is Glide and and the description is Swung, which I think does not mean something.

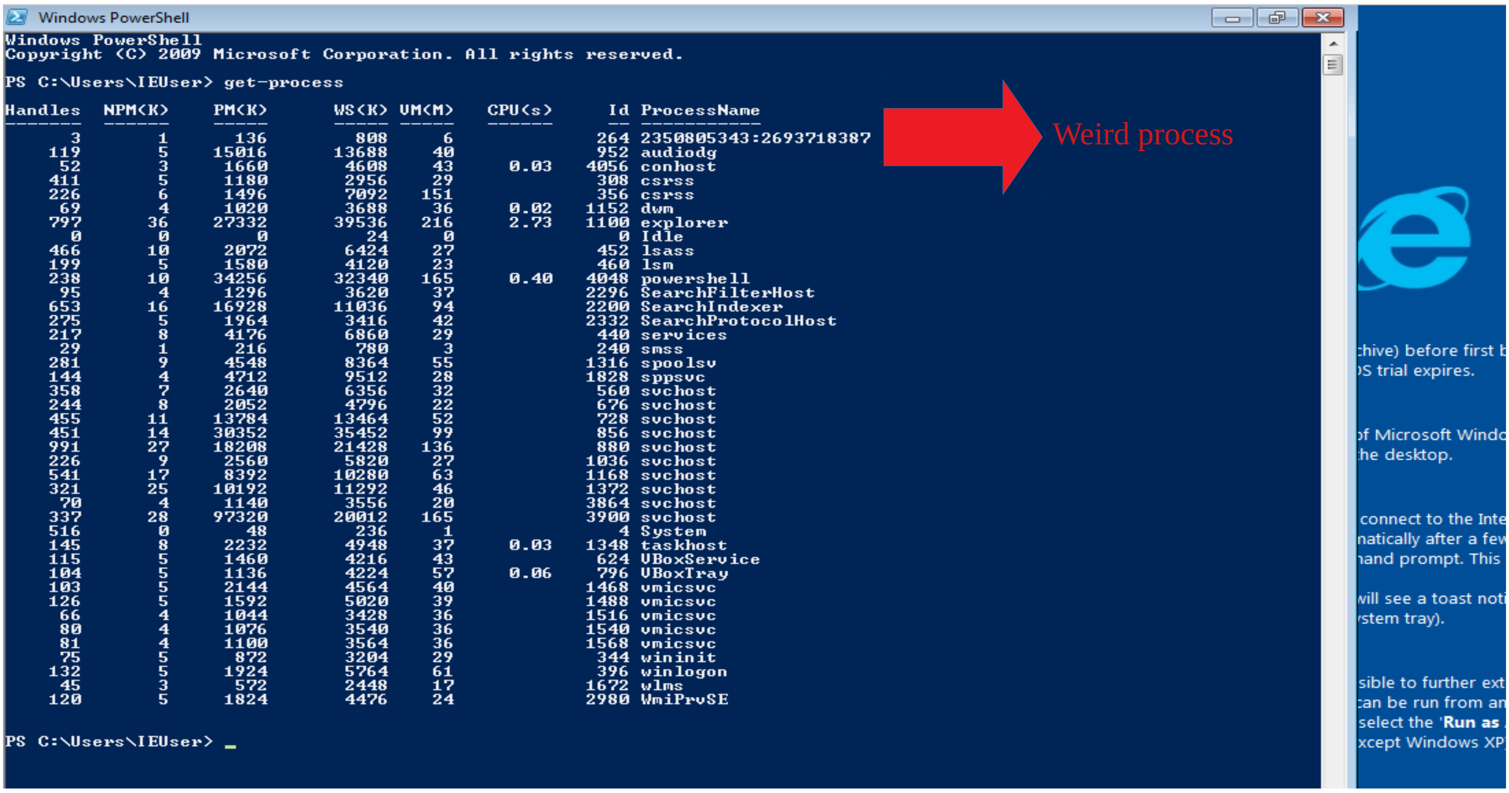
I did not find something useful about this malware at the first glance. Hence, I decided to simply restart the machine without restoring its non-infected state. When I tried to launch some analysis tools, I noticed that **I did not have permissions to do so**. This leaded me to try to understand what exactly the malware did to change my user permissions.

I decided to run a couple more times the malware with the initial state of the machine(restored snapshot), and I got some more information.

Right after the malware is launched there is a process **called "consent.exe"** that is also launched, but it closes very fast, up to the point that one could barely take a screenshot.

After making some investigation[2] about this process, I realized that it could only be consent.exe that was changing the permissions I have. The malware could have launched consent.exe to do whatever it wanted, because it had admin rights. I finally understood the reason why I did not have any permissions to those specific files that make forensics. The malware was protecting itself for those kind of tools.

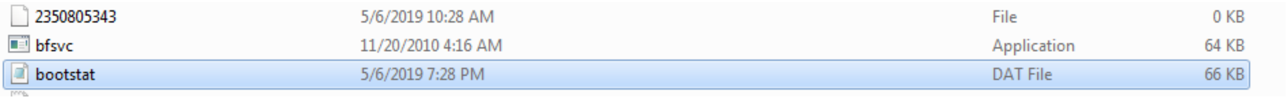
Also, I noticed the presence of some abnormal process. I could not use proexe, hence I used powershell with the following command: *get-process*



```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\IEUser> get-process

Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id ProcessName
-------  ------    -----      ----- -----   ------     -- -----------
      3       1      136        808     6              264 2350805343:2693718387
    119       5    15016      13688    40              952 audiodg
     52       3     1660       4608    43     0.03   4056 conhost
    411       5     1180       2956    29              308 csrss
    226       6     1496       7092   151              356 csrss
     69       4     1020       3688    36     0.02   1152 dwm
    797      36    27332      39536   216     2.73   1100 explorer
      0       0        0         24     0                0 Idle
    466      10     2072       6424    27              452 lsass
    199       5     1580       4120    23              460 lsm
    238      10    34256      32340   165     0.40   4048 powershell
     95       4     1296       3620    37             2296 SearchFilterHost
    653      16    16928      11036    94             2200 SearchIndexer
    275       5     1964       3416    42             2332 SearchProtocolHost
    217       8     4176       6860    29              440 services
     29       1      216        780     3              240 smss
    281       9     4548       8364    55             1316 spoolsv
    144       4     4712       9512    28             1828 sppsvc
    358       7     2640       6356    32              560 svchost
    244       8     2052       4796    22              676 svchost
    455      11    13784      13464    52              728 svchost
    451      14    30352      35452    99              856 svchost
    991      27    18208      21428   136              880 svchost
    226       9     2560       5820    27             1036 svchost
    541      17     8392      10280    63             1168 svchost
    321      25    10192      11292    46             1372 svchost
     70       4     1140       3556    20             3864 svchost
    337      28    97320      20012   165             3900 svchost
    516       0       48        236     1                4 System
    145       8     2232       4948    37     0.03   1348 taskhost
    115       5     1460       4216    43              624 VBoxService
    104       5     1136       4224    57     0.06    796 VBoxTray
    103       5     2144       4564    40             1468 vmicsvc
    126       5     1592       5020    39             1488 vmicsvc
     66       4     1044       3428    36             1516 vmicsvc
     80       4     1076       3540    36             1540 vmicsvc
     81       4     1100       3564    36             1568 vmicsvc
     75       5      872       3204    29              344 wininit
    132       5     1924       5764    61              396 winlogon
     45       3      572       2448    17             1672 wlms
    120       5     1824       4476    24             2980 WmiPrvSE

PS C:\Users\IEUser> _
```

I searched for the numbers that come before the semicolon, on the windows explorer and I was able to find the location of a file with 0 bytes.

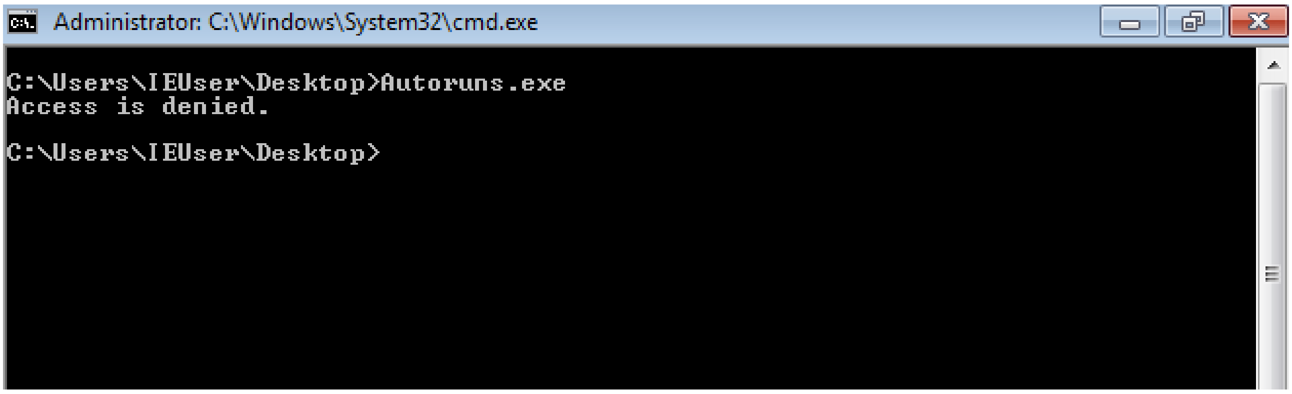
In this same location, I also found a dat file that has the word "boot" in its name:



2350805343	5/6/2019 10:28 AM	File	0 KB
bfsvc	11/20/2010 4:16 AM	Application	64 KB
bootstat	5/6/2019 7:28 PM	DAT File	66 KB

After searching, I concluded that bootstat is actually necessary to windows to analyze the boot state of the previous boot. However, I did not find any information about the file 2350805343.

I have also tried to open autoruns and some other program that I did not have permission to access with the cmd, opened with admin rights, but I was unsuccessful:



```
C:\Users\IEUser\Desktop>Autoruns.exe
Access is denied.

C:\Users\IEUser\Desktop>
```

References

[1] What smagent.exe is,
https://www.neuber.com/taskmanager/process/smagent.exe.html

[2] What is consent.exe, https://www.file.net/process/consent.exe.html