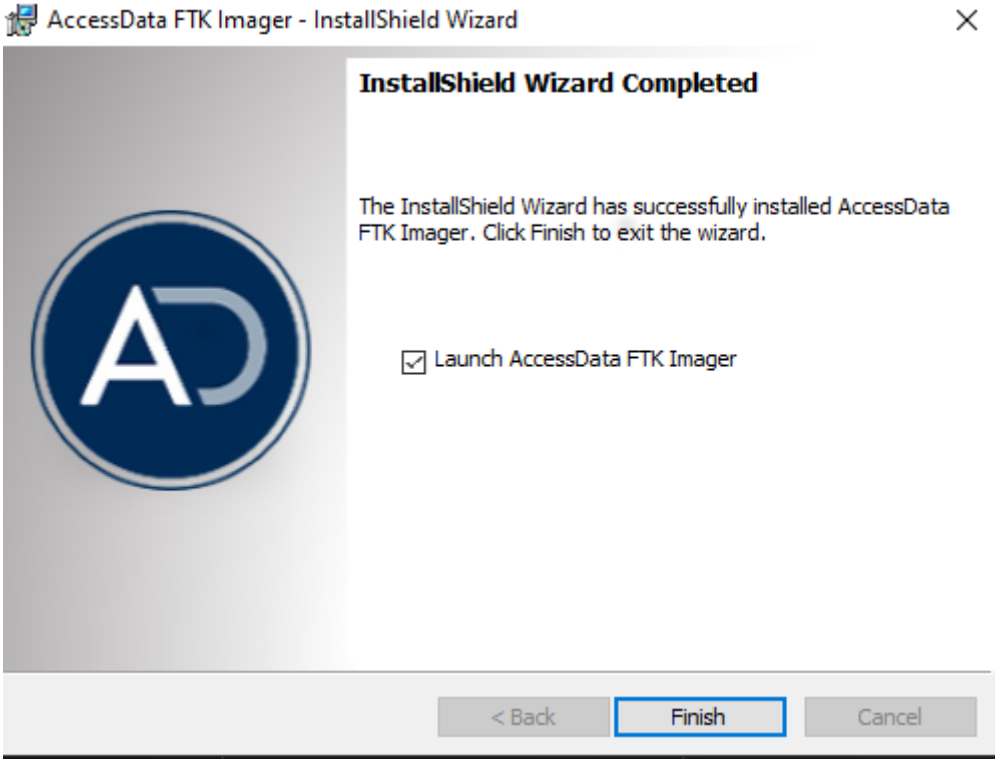
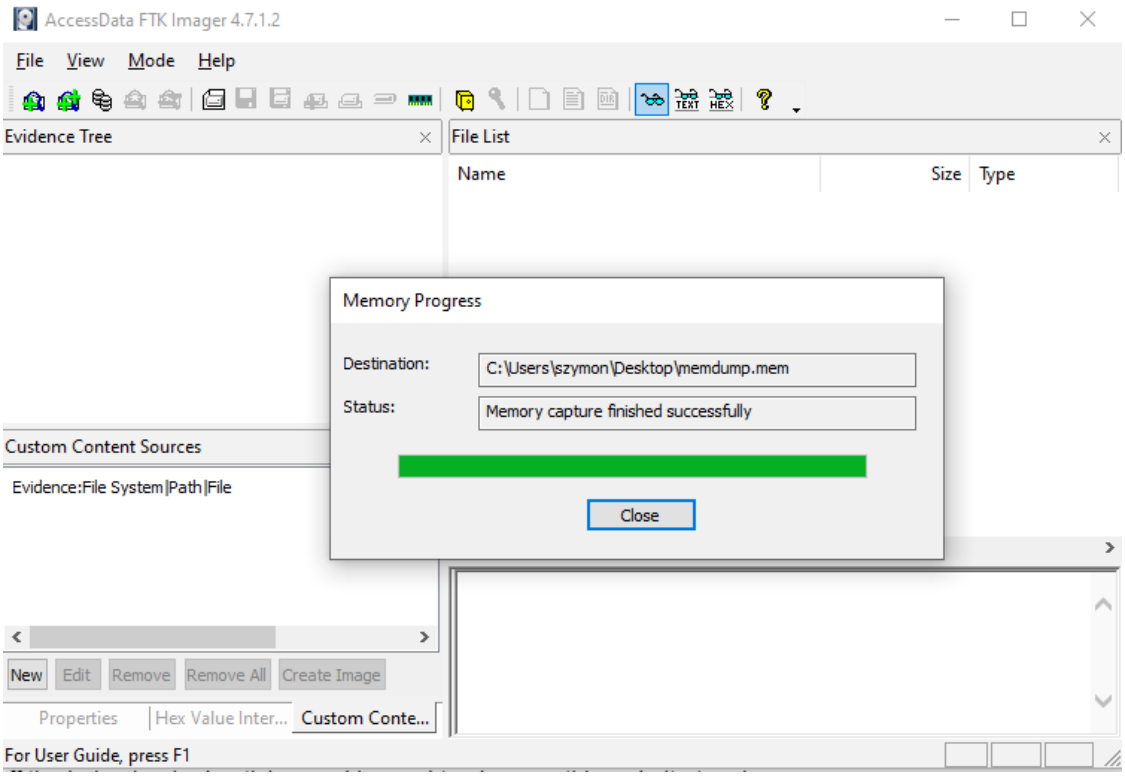
Analiza pamięci RAM – Szymon Szkarłat

Celem zadania 1 projektu jest analiza obrazu pamięci RAM (pamięci ulotnej/operacyjnej), w tym celu posłużyłem się zrzutem pamięci RAM, który utworzyłem na podstawie zasymulowania działań użytkownika na maszynie wirtualnej z z systemem Windows 10. Do wykonania zrzutu wykorzystałem program FTK Imager. Natomiast do przeprowadzania analizy wykorzystałem program Volatilty Framework zaimplementowany w Python.

Na początku wykonanie zrzutu pamięci RAM z maszyny wirtualnej Windowsa, przy wykorzystaniu FTK Imager, który zainstalowałem wcześniej.

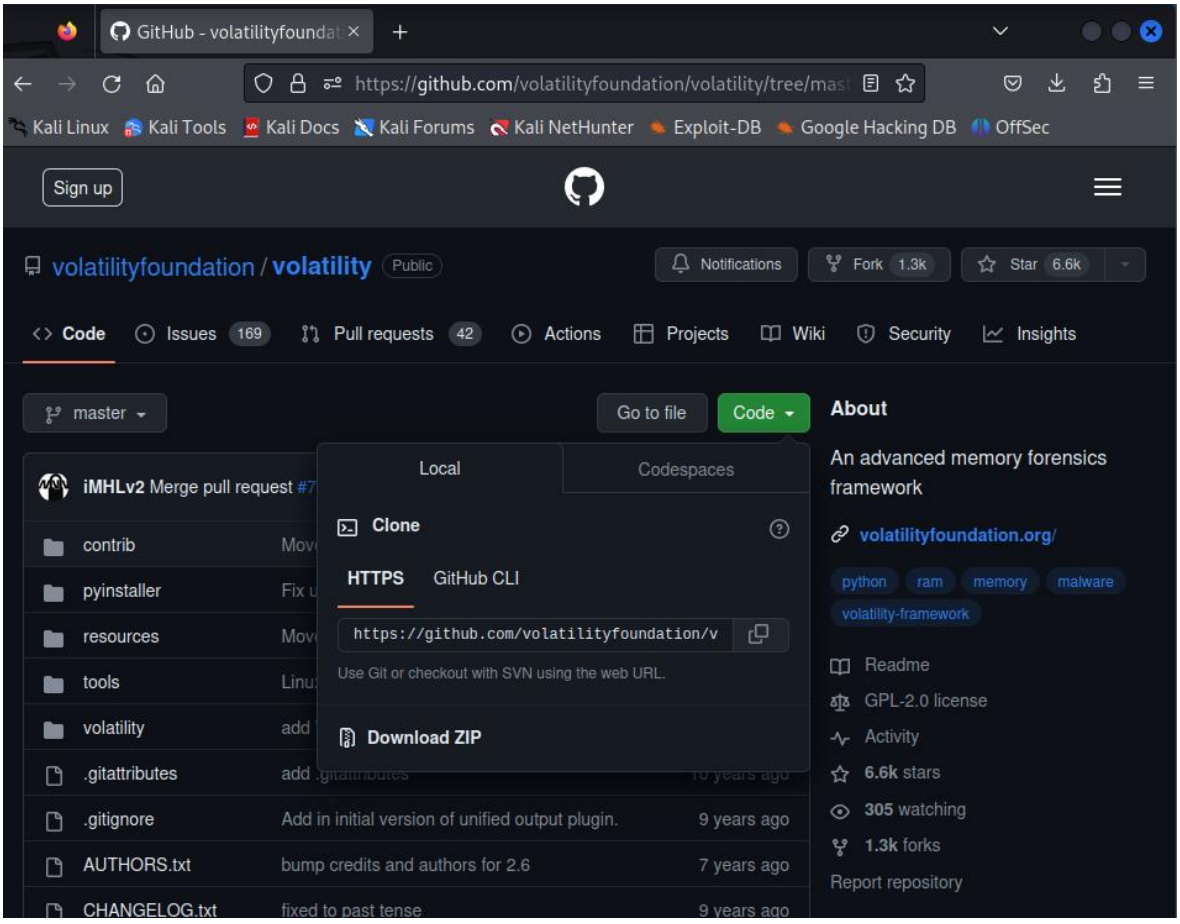


Tworzenie dumpa pamięci RAM

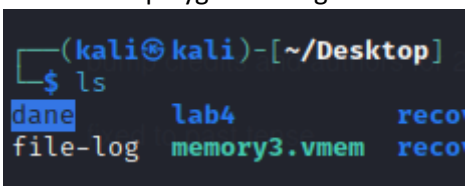


Analiza pamięci przy wykorzystaniu programu Volatility

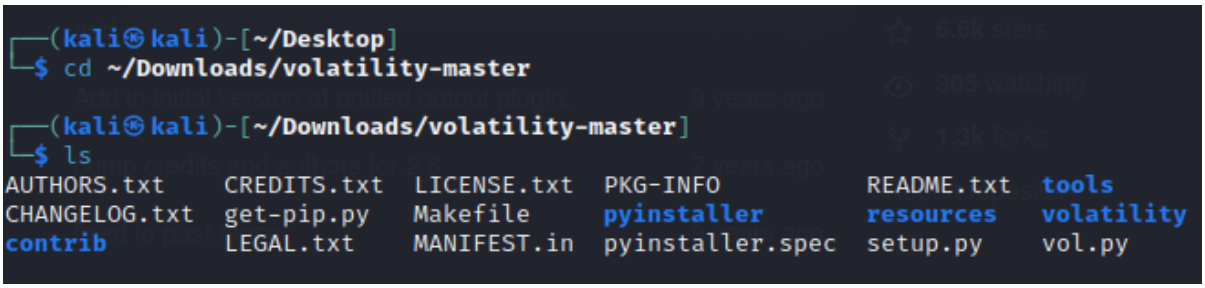
1. Pobranie programu Volatility przy wykorzystaniu wirtualnej maszyny z systemem Linux



2. Pobranie przygotowanego obrazu memory3.vmem

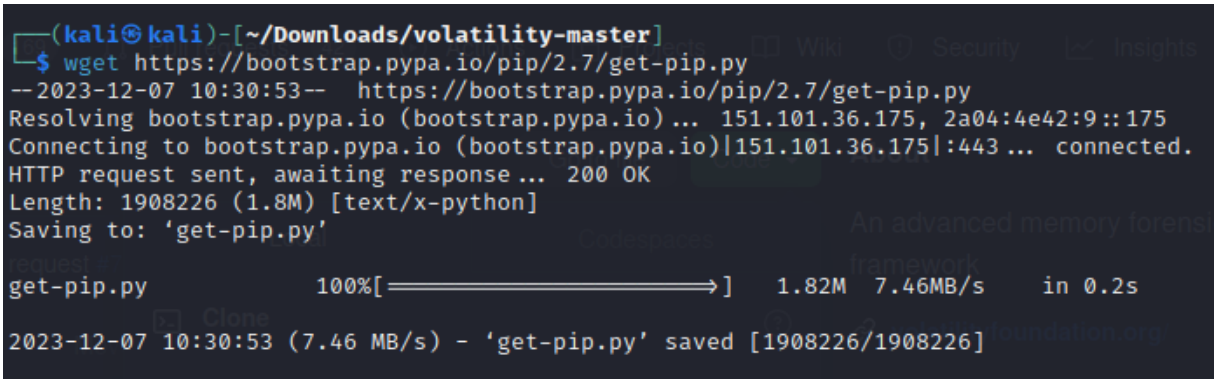


3. Przejście do pliku z pobranym frameworkiem Volatility.



4. Sprawdzenie czy program nie potrzebuje żadnych dodatkowych bibliotek

Okazało się, że dodatkowo należało zainstalować biblioteki pycrypto oraz distorm3.



```
┌──(kali㉿kali)-[~/Downloads/volatility-master]
└─$ sudo python2 get-pip.py
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgra
de your Python as Python 2.7 is no longer maintained. pip 21.0 will drop support for P
ython 2.7 in January 2021. More details about Python 2 support in pip can be found at
https://pip.pypa.io/en/latest/development/release-process/#python-2-support pip 21.0 w
ill remove support for this functionality.
Collecting pip<21.0
  Downloading pip-20.3.4-py2.py3-none-any.whl (1.5 MB)
     |████████████████████████████████| 1.5 MB 1.5 MB/s
Collecting wheel
  Downloading wheel-0.37.1-py2.py3-none-any.whl (35 kB)
Installing collected packages: pip, wheel
Successfully installed pip-20.3.4 wheel-0.37.1

┌──(kali㉿kali)-[~/Downloads/volatility-master]
└─$ pip2 install --upgrade setuptools
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgra
de your Python as Python 2.7 is no longer maintained. pip 21.0 will drop support for P
ython 2.7 in January 2021. More details about Python 2 support in pip can be found at
https://pip.pypa.io/en/latest/development/release-process/#python-2-support pip 21.0 w
ill remove support for this functionality.
Defaulting to user installation because normal site-packages is not writeable
Collecting setuptools
  Downloading setuptools-44.1.1-py2.py3-none-any.whl (583 kB)
     |████████████████████████████████| 583 kB 1.4 MB/s
Installing collected packages: setuptools
  WARNING: The scripts easy_install and easy_install-2.7 are installed in '/home/kali/
.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, u
se --no-warn-script-location.
Successfully installed setuptools-44.1.1
```

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install python2-dev
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  libpython2-dev libpython2.7 libpython2.7-dev python2.7-dev
The following NEW packages will be installed:
  libpython2-dev libpython2.7 libpython2.7-dev python2-dev python2.7-dev
0 upgraded, 5 newly installed, 0 to remove and 1212 not upgraded.
```

```
Processing triggers for man-db (2.11.2-3) ...
Processing triggers for kali-menu (2023.4.3) ...
Processing triggers for libc-bin (2.37-7) ...

┌──(kali㉿kali)-[~]
└─$ pip2 install pycrypto
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgra
de your Python as Python 2.7 is no longer maintained. pip 21.0 will drop support for P
ython 2.7 in January 2021. More details about Python 2 support in pip can be found at
https://pip.pypa.io/en/latest/development/release-process/#python-2-support pip 21.0 w
ill remove support for this functionality.
Defaulting to user installation because normal site-packages is not writeable
Collecting pycrypto
```

```
┌──(kali㉿kali)-[~]
└─$ pip2 install distorm3
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgra
de your Python as Python 2.7 is no longer maintained. pip 21.0 will drop support for P
ython 2.7 in January 2021. More details about Python 2 support in pip can be found at
https://pip.pypa.io/en/latest/development/release-process/#python-2-support pip 21.0 w
ill remove support for this functionality.
Defaulting to user installation because normal site-packages is not writeable
Collecting distorm3
  Downloading distorm3-3.5.2.tar.gz (138 kB)
     |████████████████████████████████| 138 kB 1.5 MB/s
```

Po zainstalowaniu dodatkowych bibliotek, nie wyświetlają się błędy, a jedynie „pomoc”.

```
┌──(kali㉿kali)-[~/Downloads/volatility-master]
└─$ python2 vol.py -h
Volatility Foundation Volatility Framework 2.6.1
Usage: Volatility - A memory forensics analysis platform.

Options:
  -h, --help            list all available options and their default values.
                        Default values may be set in the configuration file
                        (/etc/volatilityrc)
  --conf-file=/home/kali/.volatilityrc
                        User based configuration file
  -d, --debug           Debug volatility
  --plugins=PLUGINS     Additional plugin directories to use (colon separated)
  --info                Print information about all registered objects
  --cache-directory=/home/kali/.cache/volatility
                        Directory where cache files are stored
  --cache               Use caching
  --tz=TZ               Sets the (Olson) timezone for displaying timestamps
                        using pytz (if installed) or tzset
  -f FILENAME, --filename=FILENAME
                        Filename to use when opening an image
  --profile=WinXPSP2x86
                        Name of the profile to load (use --info to see a list
                        of supported profiles)
  -l LOCATION, --location=LOCATION
                        A URN location from which to load an address space
  -w, --write           Enable write support
  --dtb=DTB             DTB Address
  --shift=SHIFT         Mac KASLR shift address
  --output=text         Output in this format (support is module specific, see
                        the Module Output Options below)
  --output-file=OUTPUT_FILE
```

Przechodzę zatem do przeprowadzenia analizy pamięci operacyjnej (RAM)

```
┌──(kali㉿kali)-[~/Downloads/volatility-master]
└─$ python2 vol.py -f ~/Desktop/memdump.mem imageinfo
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win10x64_18362
                     AS Layer1 : SkipDuplicatesAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/kali/Desktop/memdump.mem)
                      PAE type : No PAE
                           DTB : 0x1ad002L
                          KDBG : 0xf80542e265e0L
          Number of Processors : 1
     Image Type (Service Pack) : 0
                KPCR for CPU 0 : 0xfffff8053fd1f000L
             KUSER_SHARED_DATA : 0xffffff78000000000L
           Image date and time : 2023-12-07 20:19:58 UTC+0000
     Image local date and time : 2023-12-07 21:19:58 +0100
```

Analizowany plik pamięci pochodzi z systemu operacyjnego Windows 10 (64-bit) w wersji 18362.
Podstawowe informacje:
a) profil systemu operacyjnego
Sugerowany profil to: Win10x64_18362
b) struktura pamięci:
- warstwa 1 (AS Layer1): SkipDuplicatesAMD64PagedMemory (Kernel AS).
- warstwa 2 (AS Layer2): FileAddressSpace (/home/kali/Desktop/memdump.mem).
- typ PAE: No PAE (brak rozszerzonej architektury stronicowej).
- adres DTB (Directory Table Base): 0x1ad002L.
- adres KDBG (Kernel Debugger Block): 0xf80542e265e0L.
c) inne cenne informacje:
- liczba procesorów: 1
- typ obrazu (Service Pack): 0
- KPCR (Kernel Processor Control Region) dla CPU 0: 0xfffff8053fd1f000L.
- KUSER_SHARED_DATA: 0xffffff78000000000L.

- data i czas utworzenia obrazu: 2023-12-07 20:19:58 UTC+0000.
- lokalna data i czas obrazu: 2023-12-07 21:19:58 +0100.

KDBG to struktura obsługiwana przez jądro systemu Windows do celów debugowania. Zawiera listę uruchomionych procesów i załadowanych modułów jądra. Zawiera także informacje o wersji, które pozwalają określić, czy zrzut pamięci pochodzi z systemu Windows XP czy Windows 7 oraz jaki dodatek Service Pack został zainstalowany.

```
┌──(kali㉿kali)-[~/Downloads/volatility-master]
└─$ python2 vol.py -f ~/Desktop/memdump.mem kdbgscan
Volatility Foundation Volatility Framework 2.6.1
**************************************************
Instantiating KDBG using: Unnamed AS Win10x64_18362 (6.4.18362 64bit)
Offset (V)                    : 0xf80542e265e0
Offset (P)                    : 0x20265e0
KdCopyDataBlock (V)           : 0xf80542ca2324
Block encoded                 : Yes
Wait never                    : 0xd3fb00034b5a653a
Wait always                   : 0x34b5b25ae40018
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win10x64_18362
Version64                     : 0xf80542e2a3c8 (Major: 15, Minor: 18362)
Service Pack (CmNtCSDVersion) : 0
Build string (NtBuildLab)     : 18362.1.amd64fre.19h1_release.19
PsActiveProcessHead           : 0xfffff80542e38b60 (161 processes)
PsLoadedModuleList            : 0xfffff80542e48150 (167 modules)
KernelBase                    : 0xfffff80542a00000 (Matches MZ: True)
Major (OptionalHeader)        : 10
Minor (OptionalHeader)        : 0
KPCR                          : 0xfffff8053fd1f000 (CPU 0)
```

Informacje KDBG:
- Offset (V): Adres wirtualny KDBG: 0xf80542e265e0.
- Offset (P): Adres fizyczny KDBG: 0x20265e0.
- KdCopyDataBlock (V): Adres funkcji KdCopyDataBlock: 0xf80542ca2324.
- Block encoded: KDBG jest zakodowany.

Informacje o Wersji Systemu Operacyjnego:
- Version64: Wersja systemu operacyjnego w formie liczby 64-bitowej: 0xf80542e2a3c8 (Major: 15, Minor: 18362).
- Service Pack (CmNtCSDVersion): Informacja o Service Pack: 0.
- Build string (NtBuildLab): Opis budowy systemu operacyjnego: 18362.1.amd64fre.19h1_release.19.

Informacje o Procesach i Modułach:
- psActiveProcessHead: Adres listy aktywnych procesów: 0xfffff80542e38b60 (161 procesów).
- psLoadedModuleList: Adres listy załadowanych modułów: 0xfffff80542e48150 (167 modułów).

Wywołanie funkcji w celu wyświetlenia listy procesów systemu

```
┌──(kali㉿kali)-[~/Downloads/volatility-master]
└─$ python2 vol.py -f ~/Desktop/memdump.mem --profile=Win10x64_18362 pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0xffff9e0d6066b300 System                    4      0     95        0 ------      0 2023-12-07 19:30:38 UTC+0000
0xffff9e0d607bd080 Registry                 68      4      4        0 ------      0 2023-12-07 19:30:33 UTC+0000
0xffff9e0d6340d400 smss.exe                516      4      2        0 ------      0 2023-12-07 19:30:38 UTC+0000
0xffff9e0d63322080 csrss.exe               624    608      9        0      0      0 2023-12-07 19:30:39 UTC+0000
0xffff9e0d63f73080 wininit.exe             692    608      1        0      0      0 2023-12-07 19:30:39 UTC+0000
0xffff9e0d63f72080 services.exe            812    692      9        0      0      0 2023-12-07 19:30:40 UTC+0000
0xffff9e0d63ee4140 lsass.exe               820    692      9        0      0      0 2023-12-07 19:30:40 UTC+0000
0xffff9e0d64768080 fontdrvhost.ex          924    692      5        0      0      0 2023-12-07 19:30:40 UTC+0000
0xffff9e0d64783240 svchost.exe             932    812      1        0      0      0 2023-12-07 19:30:40 UTC+0000
0xffff9e0d64790240 svchost.exe             992    812     14        0      0      0 2023-12-07 19:30:40 UTC+0000
0xffff9e0d6378d080 svchost.exe             548    812     12        0      0      0 2023-12-07 19:30:40 UTC+0000
0xffff9e0d64817240 svchost.exe             688    812      5        0      0      0 2023-12-07 19:30:40 UTC+0000
0xffff9e0d64837080 sppsvc.exe              880    812      0 ------      0      0 2023-12-07 19:30:40 UTC+0000   2023-12-07 19:34:23 UTC+0000
0xffff9e0d648b8280 svchost.exe            1044    812      1        0      0      0 2023-12-07 19:30:41 UTC+0000
0xffff9e0d648d22c0 svchost.exe            1084    812      2        0      0      0 2023-12-07 19:30:41 UTC+0000
0xffff9e0d64948300 svchost.exe            1168    812      1        0      0      0 2023-12-07 19:30:41 UTC+0000
0xffff9e0d6074f080 msoobe.exe             1288   1272      0 ------      1      0 2023-12-07 19:30:44 UTC+0000   2023-12-07 19:31:40 UTC+0000
0xffff9e0d6074b080 svchost.exe            1304    812      2        0      0      0 2023-12-07 19:30:44 UTC+0000
0xffff9e0d6073c480 svchost.exe            1332    812      8        0      0      0 2023-12-07 19:30:45 UTC+0000
0xffff9e0d60703080 svchost.exe            1412    812      6        0      0      0 2023-12-07 19:30:46 UTC+0000
0xffff9e0d64950240 svchost.exe            1480    812      9        0      0      0 2023-12-07 19:30:46 UTC+0000
0xffff9e0d607aa080 svchost.exe            1552    812      4        0      0      0 2023-12-07 19:30:46 UTC+0000
0xffff9e0d6079e080 svchost.exe            1580    812      6        0      0      0 2023-12-07 19:30:46 UTC+0000
0xffff9e0d64a23240 svchost.exe            1620    812      0 ------      0      0 2023-12-07 19:30:47 UTC+0000   2023-12-07 19:54:47 UTC+0000
0xffff9e0d64a81300 svchost.exe            1672    812      3        0      0      0 2023-12-07 19:30:47 UTC+0000
```

Procesów jest całkiem sporo. Poszczególne kolumny zawierają bardzo ważne informacje, tj. odpowiednio:
Offset(V) – adres wirtualny, na którym zaczyna się rekord informacji o procesie
PID – numer identyfikacyjny procesu (Process ID)
PPID – numer ID rodzica procesu (Parent Process ID)
Thds – liczba wątków, które są skojarzone z danycm procesem (Threads)
Hnds – liczba uchwytów, czyli otwartych obiektów przez proces (Handles)
Sess = numer sesji , w ramach której działa proces
Wow64 – informacja, czy proces działa w trybie WOW64 (Windows-on-Windows 64-bit)
Start – data i godzina uruchomienia procesu
Exit – data i godzina zakończenia procesu (jeśli proces został zakończony)

Znacznik (V) w rubryce Offset oznacza, że dane są w formacie wirtualnym (Virtual). Oznacza to, że wartości w tej kolumnie są adresami wirtualnymi, a nie fizycznymi.

Na powyżej prezentowanym screenie możemy dostrzec między innymi procesy, które się zakończyły. Kilka przykładów:
1) Proces o ID 880

```
0xffff9e0d64837080 sppsvc.exe             880    812      0 ------      0      0 2023-12-07 19:30:40 UTC+0000   2023-12-07 19:34:23 UTC+0000
```

Powyższa linia przedstawia informacje o procesie o nazwie „sppsvc.exe”, który miał PID 880, został on uruchomiony przez proces o identyfikatorze PPID równym 812. Miał priorytet równy 0, nie używał żadnej pamięci, został utworzony o 19:30:40 i zakończony o 19:34:23 w dniu 7 grudnia 2023 roku.
2) Proces o ID 2308 – używał on pamięć, w przeciwieństwie do procesu 880

```
0xffff9e0d651d84c0 svchost.exe            3208    812      0 ------      0      0 2023-12-07 19:30:52 UTC+0000   2023-12-07 20:15:58 UTC+0000
```

3) Na samym końcu jest proces FTK Imager.exe to narzędzie służące do tworzenia kopii obrazów dysków. Proces jest aktualnie używany. Został utworzony 2023-12-07 20:18:51. Liczba wątków, które są skojarzone z tym procesem to 2. PID = 2440.

```
0xffff9e0d64298080 svchost.exe            7768    812      5        0      0      0 2023-12-07 20:18:46 UTC+0000
0xffff9e0d656ec080 FTK Imager.exe         2440   2508     22        0      2      0 2023-12-07 20:18:51 UTC+0000
```

Procesy "System" i "smss.exe" nie posiadają informacji w rubryce "Sess" (sesja). To wynika z faktu, że te procesy są uruchamiane w kontekście systemowym, niezależnym od sesji użytkownika.

Podsumowując, Volatilty Framework było dla mnie bardzo przydatnym i pomocnym narzędziem podczas analizy pamięci operacyjnej (pamięci RAM). Jest to całkowicie opensource'owe narzędzie napisane przy pomocy języka Python. Narzędzie pozwoli mi zapoznać się z technikami oraz złożonością związaną z wydobywaniem cyfrowych artefaktów z pamięci ulotnej. Mogłem przeanalizować co użytkownik robił: jakie aplikacje miał uruchomione oraz jakie strony odwiedzał podczas korzystania z przeglądarki Internet Explorer.
Co więcej, analiza pamięci RAM za pomocą narzędzia Volatility jest często wykorzystywana w dziedzinie Cyberbezpieczeństwa.
Podstawowe cele analizy pamięci RAM:

- Identyfikacja podejrzanych procesów.
- Odkrywanie podejrzanych zachowań systemu.
- Odzyskiwanie informacji o procesach, połączeniach sieciowych, otwartych plikach itp.
- Wykrywanie i analiza szkodliwego oprogramowania.

Analiza pamięci RAM za pomocą Volatility jest skomplikowanym procesem, wymagającym zaawansowanej wiedzy z zakresu systemów operacyjnych, sieci i bezpieczeństwa komputerowego. Nie mniej sprawiła mi olbrzymią przyjemność oraz wiele mnie nauczyła.