INCIDENT RESPONSE AND FORENSICS

SUBMITTED TO: SMART INTERNZ

SUBMITTED BY:
K. ASHFAAQ AHMED 20BCN7025
MAGAM PAVAN ADITHYA 20BCN7106

Introduction:

- Incident response and computer forensics are closely related disciplines in the field of cybersecurity. While incident response focuses on the immediate response and management of security incidents, computer forensics deals with the collection, analysis, and preservation of digital evidence related to those incidents. Here's a brief overview of each:

- Incident Response:
- Incident response involves the structured and coordinated approach to managing and mitigating the impact of security incidents. The main objectives of incident response are to minimize damage, identify the source and nature of the incident, and restore normal operations as quickly as possible. Key activities in incident response include:

    - Preparation: Establishing incident response plans, defining roles and responsibilities, and implementing necessary technical controls.
    - Detection and Analysis: Monitoring systems and networks for signs of security incidents, analyzing indicators of compromise, and determining the scope and severity of the incident.
    - Containment and Eradication: Taking immediate actions to contain the incident, prevent further damage, and remove the cause from affected systems.
    - Recovery and Remediation: Restoring affected systems to their normal operational state, implementing additional security controls, and patching vulnerabilities to prevent similar incidents in the future.
    - Post-Incident Analysis: Conducting a thorough analysis of the incident, identifying lessons learned, and implementing improvements to enhance incident response capabilities.

- Computer Forensics:
- Computer forensics is the application of scientific methods and techniques to collect, analyze, and preserve digital evidence for legal or investigative purposes. It involves the systematic and meticulous examination of digital artifacts to reconstruct events, establish facts, and support legal proceedings. Key aspects of computer forensics include:

- Evidence Collection: Identifying and preserving digital evidence in a forensically sound manner to ensure its integrity and admissibility in legal proceedings.
- Analysis and Examination: Conducting in-depth analysis of digital artifacts, such as hard drives, memory dumps, network logs, and application data, to extract relevant information and uncover the sequence of events.
- Data Recovery: Employing specialized tools and techniques to recover deleted or hidden data that may be crucial to the investigation.
- Reporting and Presentation: Documenting findings in a clear and concise manner, presenting evidence in a format suitable for legal purposes, and providing expert testimony if required.

LITERATURE SURVEY AND THEORITICAL ANALYSIS

Volatility and voldiff are powerful tools that can be used to gather valuable information during incident response and forensics investigations. By using these tools, investigators can quickly identify the signs of a compromise and gather evidence that can be used to track down the attackers.

Here are some examples of how Volatility and voldiff can be used in incident response and forensics:

- Identifying malicious processes: Volatility can be used to list all of the processes that were running at the time of the memory dump. This information can be used to identify malicious processes, such as those that are known to be associated with malware.
- Tracking network activity: Volatility can be used to list all of the active and closed network connections at the time of the memory dump. This information can be used to track network activity that may have been associated with the compromise, such as connections to known malware command-and-control servers.
- Identifying malicious DLLs: Volatility can be used to list all of the DLLs that were loaded into memory at the time of the memory dump. This information can be used to identify malicious DLLs, such as those that are known to be used by malware.
- Comparing memory dumps: Voldiff can be used to compare two memory dumps to identify changes. This can be useful for identifying changes that were made to the system after the first memory dump was taken, such as changes to the list of processes, network connections, or DLLs.

Volatility and voldiff are powerful tools that can be used to gather valuable information during incident response and forensics investigations. By using these tools, investigators can quickly identify the signs of a compromise and gather evidence that can be used to track down the attackers.

What is Memory Forensics

- Memory forensics is the process of analyzing the volatile data in a computer's memory. This data can be used to identify malware, track down attackers, and reconstruct events that have taken place on asystem.
- Memory forensics is important because it can provide investigators with evidence that may not be available from other sources, such as disk images. For example, memory forensics can be used to identify malware that has been installed on a system, to track down the source of a network attack, orto identify who was using a system at a particular time.
- The contents of memory can change over time, so it is important to analyze memory as soon as possible after an incident. There are a number of tools that can be used for memory forensics, including Volatility, X-Ways Forensics, and The Sleuth Kit. These tools allow analysts to extract information from memory dumps, such as process information, network connections, and file systemaccess.
- Memory forensics is a complex and challenging field, but it can be a valuable tool for investigators. Byunderstanding volatility and using the right tools, analysts can gather valuable evidence that can be used to solve crimes and protect systems from attack.
- **Here are some of the benefits of using memory forensics:**
- It can be used to gather evidence that may not be available from other sources, such as disk images.
- It can be used to identify malware that has been installed on a system.
- It can be used to track down the source of a network attack.
- It can be used to identify who was using a system at a particular time.

What is volatility

- In memory forensics, volatility refers to the fact that the contents of memory can change over time. This is because memory is constantly being used by the operating system and applications. As a result, it is important to analyze memoryas soon as possible after an incident in order to preserve the evidence.
- There are a number of factors that can cause memory to become volatile, including:
- Power cycling the system: When a system is powered off, all of the data in memory is lost.
- Writing to disk: When data is written to disk, it is also copied to memory. This means that if data is written to disk, it may also be overwritten in memory.
- Running applications: When applications are run, they can change the contents of memory. This means that if an application is run after an incident, it may overwrite evidence.
- It is important to note that not all data in memory is volatile. Some data, such as the contents of the kernel memory, is not overwritten until the system is powered off. However, most of the data in memory is volatile and can be overwritten quickly.
- As a result of volatility, it is important to analyze memory as soon as possible after an incident. This will help to ensure thatthe evidence is not lost or overwritten. There are a number of tools that can be used to analyze memory, including Volatility, X-Ways Forensics, and The Sleuth Kit. These tools allow analysts to extract information from memory dumps, such as process information, network connections, and file system access.
- Memory forensics is a complex and challenging field, but it can be a valuable tool for investigators. By understanding volatility and using the right tools, analysts can gather valuable evidence that can be used to solve crimes and protect systems from attack.

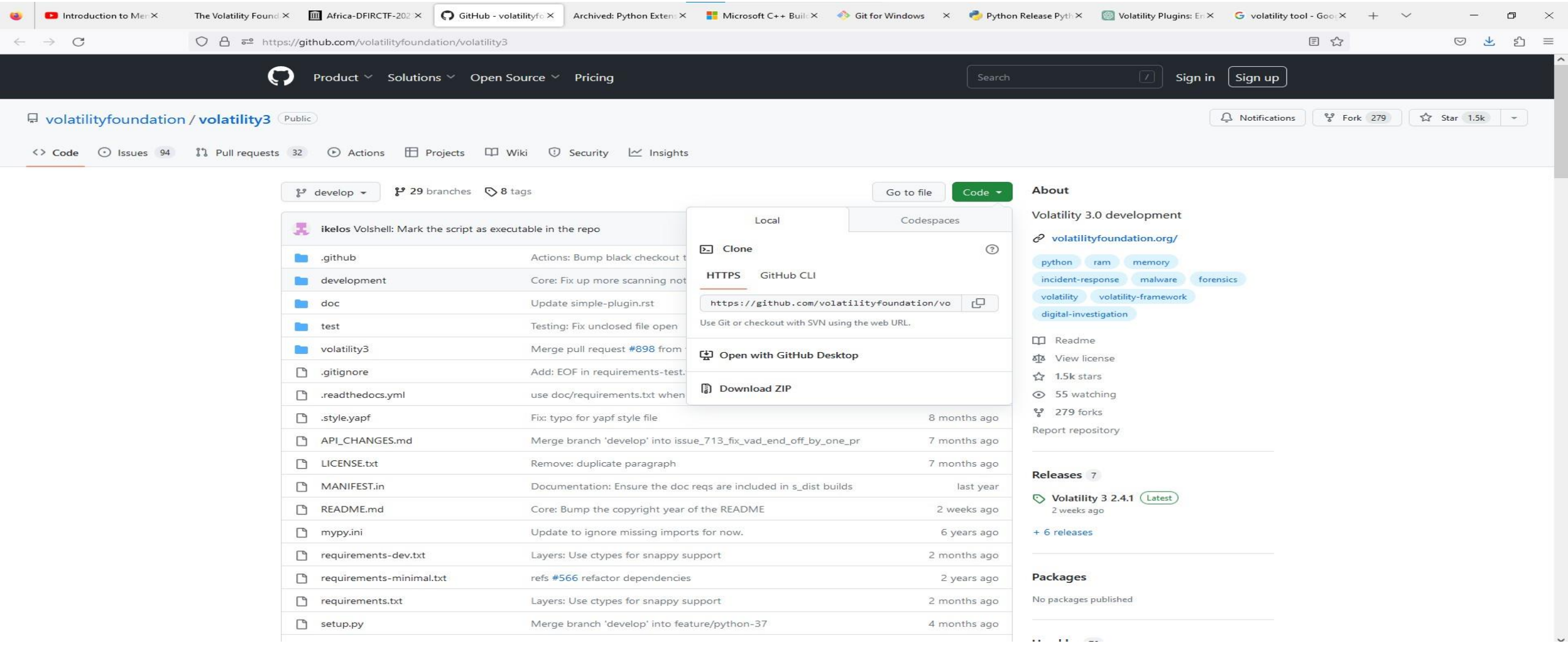
- Introduction to volatility 3
- Install volatility 3 on windows
- Find the path of target memory image
- Get RAM image information with windows.info
- Listing installed plugins
- Get process list from RAM with windows.pslist
- Filter Volatility output with powershell Select-String
- Find process handles with windows.handles
- Dump a specific file from RAM with windows .dumpfile
- Dump all files related to PID
- Find active network connections with netstat
- Find local user password hash with windows.hashdump
- Analyze user actions with windows.registry.userassist

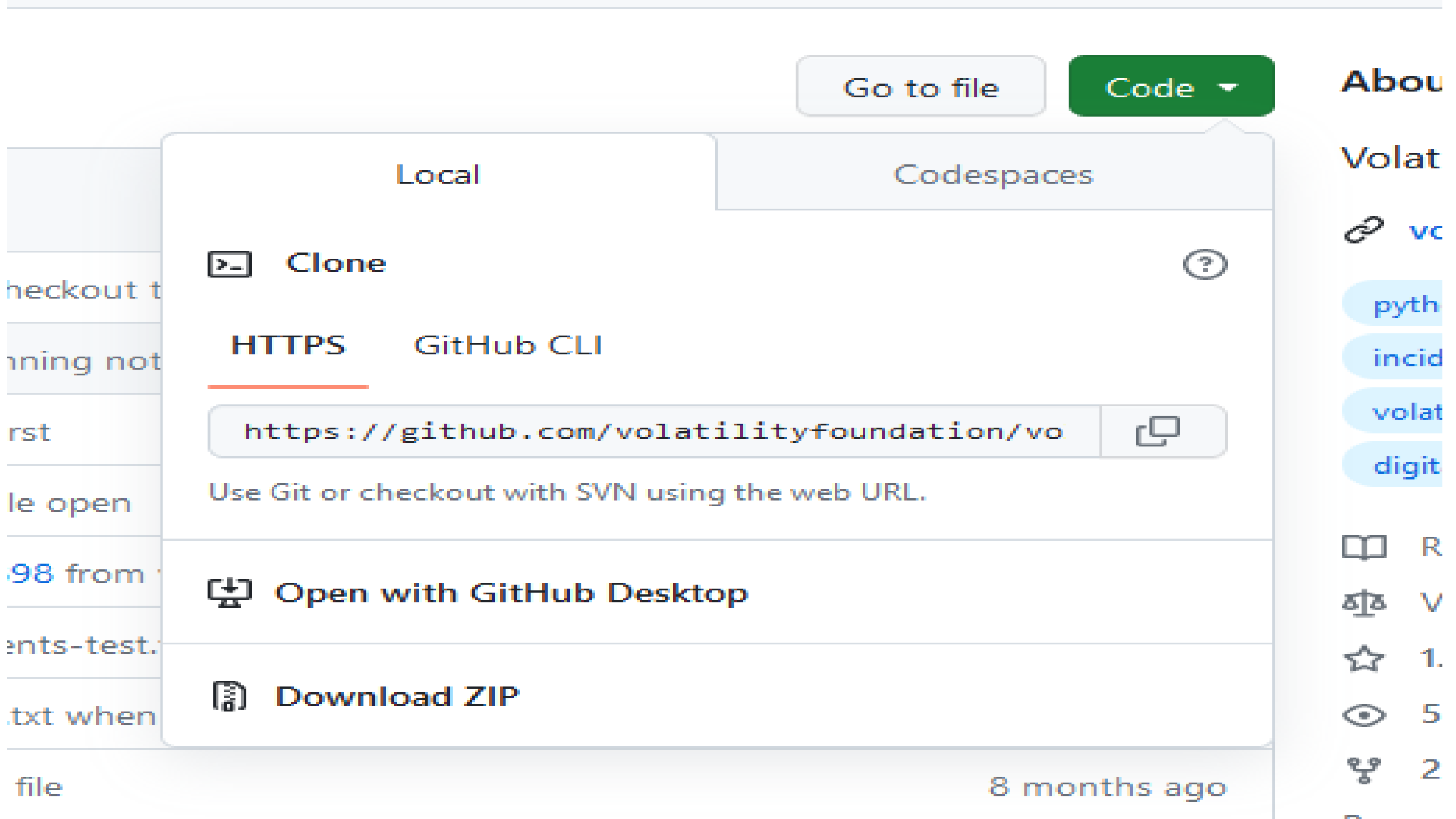
EXPERIMENTAL INVESTIGATIONS:

Installation

- Download new version of Python now as we dealing with volatility 3 itsupports python3.
- We need Git for Windows to download lot of forensic tools using git
- We need visual studio cpp build tools this is to build python modules thatway we can run volatility.
- Go to github release page and copy https url.
- Links: * Python: https://python.org (get version 3) * Git for Windows:https://gitforwindows.org/ * Microsoft C++ Build Tools: https://visualstudio.microsoft.com/vi... * Python Snappy: https://www.lfd.uci.edu/~gohlke/pytho... * Volatility 3: https://github.com/volatilityfoundati... * Practice memory image: https://archive.org/details/Africa-DF-...

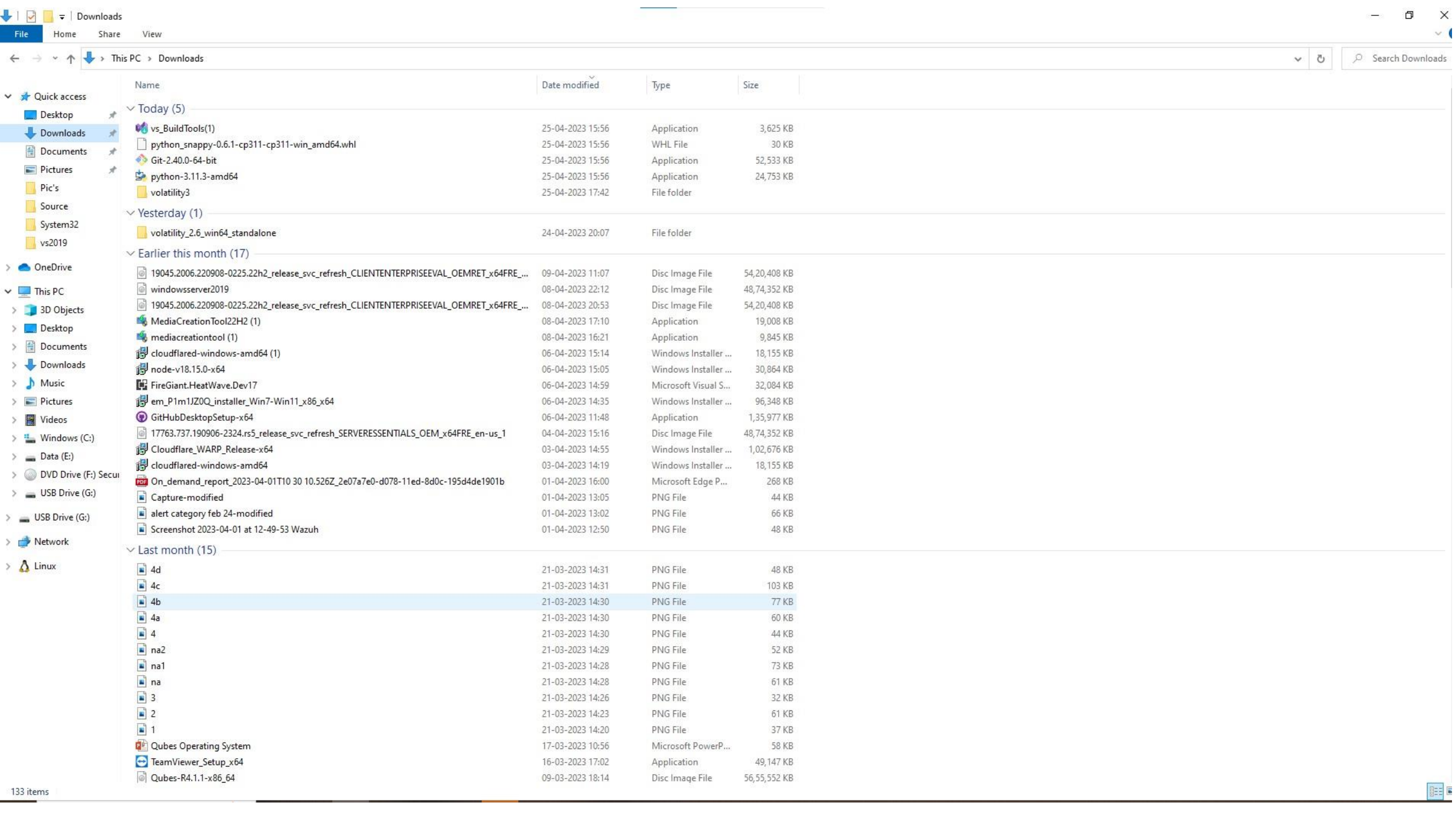
1.Visit this https://github.com/volatilityfoundation/volatility3and copy url.




Go to file
Code
Local
Codespaces
Clone
HTTPS
GitHub CLI
https://github.com/volatilityfoundation/vo
Use Git or checkout with SVN using the web URL.
Open with GitHub Desktop
Download ZIP
8 months ago
Abou
Volat
pyth
incid
volat
digit

1. Clone volatility code into my downloads

```
PS C:\Users\VIT-AP> cd Downloads
PS C:\Users\VIT-AP\Downloads> git clone https://github.com/volatilityfoundation/volatility3.git
Cloning into 'volatility3'...
remote: Enumerating objects: 30378, done.
remote: Counting objects: 100% (1222/1222), done.
remote: Compressing objects: 100% (543/543), done.
remote: Total 30378 (delta 795), reused 1036 (delta 672), pack-reused 29156
Receiving objects: 100% (30378/30378), 6.15 MiB | 2.47 MiB/s, done.
Resolving deltas: 100% (22976/22976), done.
```


Downloads
File
Home
Share
View
This PC › Downloads
Search Downloads
Quick access
Desktop
Downloads
Documents
Pictures
Pic's
Source
System32
vs2019
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
Data (E:)
DVD Drive (F:) Secui
USB Drive (G:)
USB Drive (G:)
Network
Linux
Name
Date modified
Type
Size
Today (5)
vs_BuildTools(1) 25-04-2023 15:56 Application 3,625 KB
python_snappy-0.6.1-cp311-cp311-win_amd64.whl 25-04-2023 15:56 WHL File 30 KB
Git-2.40.0-64-bit 25-04-2023 15:56 Application 52,533 KB
python-3.11.3-amd64 25-04-2023 15:56 Application 24,753 KB
volatility3 25-04-2023 17:42 File folder
Yesterday (1)
volatility_2.6_win64_standalone 24-04-2023 20:07 File folder
Earlier this month (17)
19045.2006.220908-0225.22h2_release_svc_refresh_CLIENTENTERPRISEEVAL_OEMRET_x64FRE_... 09-04-2023 11:07 Disc Image File 54,20,408 KB
windowsserver2019 08-04-2023 22:12 Disc Image File 48,74,352 KB
19045.2006.220908-0225.22h2_release_svc_refresh_CLIENTENTERPRISEEVAL_OEMRET_x64FRE_... 08-04-2023 20:53 Disc Image File 54,20,408 KB
MediaCreationTool22H2 (1) 08-04-2023 17:10 Application 19,008 KB
mediacreationtool (1) 08-04-2023 16:21 Application 9,845 KB
cloudflared-windows-amd64 (1) 06-04-2023 15:14 Windows Installer ... 18,155 KB
node-v18.15.0-x64 06-04-2023 15:05 Windows Installer ... 30,864 KB
FireGiant.HeatWave.Dev17 06-04-2023 14:59 Microsoft Visual S... 32,084 KB
em_P1m1JZ0Q_installer_Win7-Win11_x86_x64 06-04-2023 14:35 Windows Installer ... 96,348 KB
GitHubDesktopSetup-x64 06-04-2023 11:48 Application 1,35,977 KB
17763.737.190906-2324.rs5_release_svc_refresh_SERVERESSENTIALS_OEM_x64FRE_en-us_1 04-04-2023 15:16 Disc Image File 48,74,352 KB
Cloudflare_WARP_Release-x64 03-04-2023 14:55 Windows Installer ... 1,02,676 KB
cloudflared-windows-amd64 03-04-2023 14:19 Windows Installer ... 18,155 KB
On_demand_report_2023-04-01T10 30 10.526Z_2e07a7e0-d078-11ed-8d0c-195d4de1901b 01-04-2023 16:00 Microsoft Edge P... 268 KB
Capture-modified 01-04-2023 13:05 PNG File 44 KB
alert category feb 24-modified 01-04-2023 13:02 PNG File 66 KB
Screenshot 2023-04-01 at 12-49-53 Wazuh 01-04-2023 12:50 PNG File 48 KB
Last month (15)
4d 21-03-2023 14:31 PNG File 48 KB
4c 21-03-2023 14:31 PNG File 103 KB
4b 21-03-2023 14:30 PNG File 77 KB
4a 21-03-2023 14:30 PNG File 60 KB
4 21-03-2023 14:30 PNG File 44 KB
na2 21-03-2023 14:29 PNG File 52 KB
na1 21-03-2023 14:28 PNG File 73 KB
na 21-03-2023 14:28 PNG File 61 KB
3 21-03-2023 14:26 PNG File 32 KB
2 21-03-2023 14:23 PNG File 61 KB
1 21-03-2023 14:20 PNG File 37 KB
Qubes Operating System 17-03-2023 10:56 Microsoft PowerP... 58 KB
TeamViewer_Setup_x64 16-03-2023 17:02 Application 49,147 KB
Qubes-R4.1.1-x86_64 09-03-2023 18:14 Disc Image File 56,55,552 KB
133 items

2. Install python snappy extension

Python-snappy: wraps the snappy compression library.

python_snappy-0.6.1-pp38-pypy38_pp73-win_amd64.whl
python_snappy-0.6.1-cp311-cp311-win_amd64.whl
python_snappy-0.6.1-cp311-cp311-win32.whl [31 KB] [May 10, 2022]
python_snappy-0.6.1-cp310-cp310-win_amd64.whl
python_snappy-0.6.1-cp310-cp310-win32.whl
python_snappy-0.6.1-cp39-cp39-win_amd64.whl
python_snappy-0.6.1-cp39-cp39-win32.whl
python_snappy-0.6.1-cp38-cp38-win_amd64.whl
python_snappy-0.6.1-cp38-cp38-win32.whl
python_snappy-0.6.1-cp37-cp37m-win_amd64.whl
python_snappy-0.6.1-cp37-cp37m-win32.whl
python_snappy-0.5.4-cp36-cp36m-win_amd64.whl
python_snappy-0.5.4-cp36-cp36m-win32.whl
python_snappy-0.5.4-cp35-cp35m-win_amd64.whl
python_snappy-0.5.4-cp35-cp35m-win32.whl
python_snappy-0.5.4-cp27-cp27m-win_amd64.whl
python_snappy-0.5.4-cp27-cp27m-win32.whl
python_snappy-0.5.3-cp34-cp34m-win_amd64.whl
python_snappy-0.5.3-cp34-cp34m-win32.whl

https://visualstudio.microsoft.com/vi... * Python Snappy:

3. Check python version and Installing packages of python snappy in powershell

```
PS C:\Users\VIT-AP\Downloads> python -V
Python 3.11.3
PS C:\Users\VIT-AP\Downloads> pip install C:\Users\VIT-AP\Downloads\python_snappy-0.6.1-cp311-cp311-win_amd64.whl
Processing c:\users\vit-ap\downloads\python_snappy-0.6.1-cp311-cp311-win_amd64.whl
Installing collected packages: python-snappy
Successfully installed python-snappy-0.6.1
```

4. cd volatility 3 and see list of all files involatility 3 by ls command

```
PS C:\Users\VIT-AP\Downloads> cd .\volatility3\
```

```
PS C:\Users\VIT-AP\Downloads\volatility3> ls


    Directory: C:\Users\VIT-AP\Downloads\volatility3


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        25-04-2023     16:02                .github
d-----        25-04-2023     16:02                development
d-----        25-04-2023     16:02                doc
d-----        25-04-2023     17:42                dump
d-----        25-04-2023     16:02                test
d-----        25-04-2023     16:17                volatility3
-a----        25-04-2023     16:02            558 .gitignore
-a----        25-04-2023     16:02            520 .readthedocs.yml
-a----        25-04-2023     16:02           8200 .style.yapf
-a----        25-04-2023     16:02           1416 API_CHANGES.md
-a----        25-04-2023     16:02           3956 LICENSE.txt
-a----        25-04-2023     16:02            207 MANIFEST.in
-a----        25-04-2023     16:02             83 mypy.ini
-a----        25-04-2023     16:02           6094 README.md
-a----        25-04-2023     16:02            781 requirements-dev.txt
-a----        25-04-2023     16:02             76 requirements-minimal.txt
-a----        25-04-2023     16:02            639 requirements.txt
-a----        25-04-2023     16:02           1946 setup.py
-a----        25-04-2023     16:02            300 vol.py
-a----        25-04-2023     16:02           5560 vol.spec
-a----        25-04-2023     16:02            307 volshell.py
-a----        25-04-2023     16:02           3029 volshell.spec
```

5. We need to install all the requirements to run volatility

```
PS C:\Users\VIT-AP\Downloads\volatility3> pip install -r .\requirements.txt
Collecting pefile>=2017.8.1
  Downloading pefile-2023.2.7-py3-none-any.whl (71 kB)
     ---------------------------------------- 71.8/71.8 kB 131.2 kB/s eta 0:00:00
Collecting yara-python>=3.8.0
  Downloading yara_python-4.3.1-cp311-cp311-win_amd64.whl (1.1 MB)
     ---------------------------------------- 1.1/1.1 MB 1.1 MB/s eta 0:00:00
Collecting capstone>=3.0.5
  Downloading capstone-4.0.2-py2.py3-none-win_amd64.whl (896 kB)
     ---------------------------------------- 896.4/896.4 kB 5.7 MB/s eta 0:00:00
Collecting pycryptodome
  Downloading pycryptodome-3.17-cp35-abi3-win_amd64.whl (1.7 MB)
     ---------------------------------------- 1.7/1.7 MB 4.2 MB/s eta 0:00:00
Collecting leechcorepyc>=2.4.0
  Downloading leechcorepyc-2.14.3-cp36-abi3-win_amd64.whl (358 kB)
     ---------------------------------------- 358.4/358.4 kB 332.5 kB/s eta 0:00:00
Installing collected packages: yara-python, pycryptodome, pefile, leechcorepyc, capstone
Successfully installed capstone-4.0.2 leechcorepyc-2.14.3 pefile-2023.2.7 pycryptodome-3.17 yara-python-4.3.1
```

```
pip install -r .\requirements.txt
```

6. Checking version of volatility

```
PS C:\Users\VIT-AP\Downloads\volatility3> python vol.py -v
Volatility 3 Framework 2.4.2
INFO    volatility3.cli: Volatility plugins path: ['C:\\Users\\VIT-AP\\Downloads\\volatility3\\volatility3\\plugins', 'C:\\Users\\VIT-AP\\Downloads\\volatility3\\volatility3\\framework\\plugins']
INFO    volatility3.cli: Volatility symbols path: ['C:\\Users\\VIT-AP\\Downloads\\volatility3\\volatility3\\symbols', 'C:\\Users\\VIT-AP\\Downloads\\volatility3\\volatility3\\framework\\symbols']
usage: volatility [-h] [-c CONFIG] [--parallelism [{processes,threads,off}]] [-e EXTEND] [-p PLUGIN_DIRS] [-s SYMBOL_DIRS] [-v]
                  [-l LOG] [-o OUTPUT_DIR] [-q] [-r RENDERER] [-f FILE] [--write-config] [--save-config SAVE_CONFIG] [--clear-cache]
                  [--cache-path CACHE_PATH] [--offline] [--single-location SINGLE_LOCATION] [--stackers [STACKERS ...]]
                  [--single-swap-locations [SINGLE_SWAP_LOCATIONS ...]]
                  plugin ...
volatility: error: Please select a plugin to run
```

python vol.py -v

Plugins

- Plugins in Volatility are like add-on modules that you can use to get more information from a computer's memory (RAM) when investigating a cybersecurity incident or performing forensic analysis.They are like extra tools that extend the functionality of Volatility, which is a framework used for analyzing memory dumps.
- For example, you might have a plugin that helps you extract information about web browsing history, another plugin that helps you detect and analyze malware, or a plugin that generates visualizations to help you better understand the memory data. ed foranalyzing memory dumps.

Next thing is to get a memory image that we wantto analyse and get location of memory image

```
python vol.py -f C:\Users\VIT-AP\Downloads\volatility3\volatility3\Win10Home-20H2-64bit-memdump.mem
```

RESULTS:

First thing to do in analysing memory is to seea information of memory dump file:

```
PS C:\Users\VIT-AP\downloads\volatility3> python vol.py -f C:\Users\VIT-AP\Downloads\volatility3\volatility3\Win10Home-20H2-64bit-memdump.mem windows.info | more
Volatility 3 Framework 2.4.2

Variable        Value

Kernel Base     0xf8043cc00000
DTB     0x1aa000
Symbols file:///C:/Users/VIT-AP/Downloads/volatility3/volatility3/symbols/windows/ntkrnlmp.pdb/769C521E4833ECF72E21F02BF33691A5-1.json.xz
Is64Bit True
IsPAE   False
layer_name      0 WindowsIntel32e
memory_layer    1 FileLayer
KdVersionBlock  0xf8043d80f368
Major/Minor     15.19041
MachineType     34404
KeNumberProcessors      4
SystemTime      2021-04-30 17:52:19
NtSystemRoot    C:\Windows
NtProductType   NtProductWinNt
NtMajorVersion  10
NtMinorVersion  0
PE MajorOperatingSystemVersion  10
PE MinorOperatingSystemVersion  0
PE Machine      34404
PE TimeDateStamp        Tue Oct 11 07:04:26 1977
```

- **The command python vol.py -f C:\Users\VIT-AP\Downloads\volatility3\volatility3\Win10Home-20H2-64bit-memdump.mem windows.info is using Volatility to gather information about the Windows system captured in the memory dump file.**
- **Here are some of the information and variables provided by the command**:
- The output of the command you ran shows information about the Windows system that the memory dump file came from. The following is a breakdown of the information that is being displayed:
- Kernel Base: The address of the kernel in memory.
- DTB: The address of the DTB (Device Tree Base) in memory.
- Symbols: The path to the symbols file for the kernel.
- Is64Bit: A flag indicating whether the system is 64-bit.
- IsPAE: A flag indicating whether the system is PAE-enabled.
- layer_name: The name of the layer that the memory dump file came from.
- memory_layer: The name of the layer that the memory dump file was created on.
- KdVersionBlock: The address of the KdVersionBlock structure in memory.
- Major/Minor: The major and minor versions of the Windows kernel.
- MachineType: The machine type of the system.
- KeNumberProcessors: The number of processors in the system.
- SystemTime: The system time at the time of the crash.
- NtSystemRoot: The path to the Windows system root directory.
- NtProductType: The type of Windows product.
- NtMajorVersion: The major version of Windows.
- NtMinorVersion: The minor version of Windows.

- PE MajorOperatingSystemVersion: The major version of the operating system that the PE image was built for. PE MinorOperatingSystemVersion: The minor version of the operating system that the PE image was built for.
- PE Machine: The machine type of the PE image.
- PE TimeDateStamp: The time stamp of the PE image.
- This information can be used to help you investigate the memory dump file and determine what caused the crash.

Process list :

```
PS C:\Users\VIT-AP\downloads\volatility3> python vol.py -f C:\Users\VIT-AP\Downloads\volatility3\volatility3\Win10Home-20H2-64bit-memdump.mem windows.pslist | more
Volatility 3 Framework 2.4.2

PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

4       0       System  0xbf0f64a63080  132     -       N/A     False   2021-04-30 12:39:40.000000      N/A     Disabled
108     4       Registry        0xbf0f64bc6040  4       -       N/A     False   2021-04-30 12:39:38.000000      N/A     Disabled
396     4       smss.exe        0xbf0f66967040  2       -       N/A     False   2021-04-30 12:39:40.000000      N/A     Disabled
492     484     csrss.exe       0xbf0f6adb6080  13      -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
568     484     wininit.exe     0xbf0f6b67a080  1       -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
584     560     csrss.exe       0xbf0f6b681080  16      -       1       False   2021-04-30 12:39:44.000000      N/A     Disabled
668     560     winlogon.exe    0xbf0f6b6db080  5       -       1       False   2021-04-30 12:39:44.000000      N/A     Disabled
712     568     services.exe    0xbf0f6b6da080  11      -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
736     568     lsass.exe       0xbf0f6b6fb0c0  10      -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
856     712     svchost.exe     0xbf0f6b7042c0  20      -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
884     568     fontdrvhost.ex  0xbf0f6b70b1c0  5       -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
892     668     fontdrvhost.ex  0xbf0f6b7091c0  5       -       1       False   2021-04-30 12:39:44.000000      N/A     Disabled
976     712     svchost.exe     0xbf0f6b785340  12      -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
320     712     svchost.exe     0xbf0f6b78a2c0  6       -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
564     668     LogonUI.exe     0xbf0f6b7b7100  0       -       1       False   2021-04-30 12:39:44.000000      2021-04-30 17:39:58.000000      Disabled
560     668     dwm.exe 0xbf0f6b7ba080  21      -       1       False   2021-04-30 12:39:44.000000      N/A     Disabled
1080    712     svchost.exe     0xbf0f6be74340  2       -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
1088    712     svchost.exe     0xbf0f6be77080  3       -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
1164    712     svchost.exe     0xbf0f6becd300  5       -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
1172    712     svchost.exe     0xbf0f6becc080  3       -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
1204    712     svchost.exe     0xbf0f6bed1080  1       -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
1296    712     svchost.exe     0xbf0f6bf192c0  7       -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
1336    712     svchost.exe     0xbf0f6bf20080  8       -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
1376    712     svchost.exe     0xbf0f6be1c080  6       -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
1440    712     svchost.exe     0xbf0f6bf9a0c0  4       -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
1452    712     svchost.exe     0xbf0f6bfa52c0  8       -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
1556    712     svchost.exe     0xbf0f64a81080  8       -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
1660    712     VBoxService.ex  0xbf0f6bff4080  11      -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
1684    712     svchost.exe     0xbf0f6bff5080  5       -       0       False   2021-04-30 12:39:44.000000      N/A     Disabled
1784    712     svchost.exe     0xbf0f6c03d340  4       -       0       False   2021-04-30 17:39:46.000000      N/A     Disabled
1804    712     svchost.exe     0xbf0f6c03b080  5       -       0       False   2021-04-30 17:39:46.000000      N/A     Disabled
1824    712     svchost.exe     0xbf0f6bfee080  3       -       0       False   2021-04-30 17:39:46.000000      N/A     Disabled
1880    712     svchost.exe     0xbf0f6c0b1340  2       -       0       False   2021-04-30 17:39:46.000000      N/A     Disabled
```

- **The output of the command you ran shows a list of all the processes that were running on the system at the time of the crash.The following is a breakdown of the information that is being displayed:**
- PID: The process ID of the process.
- PPID: The parent process ID of the process.
- ImageFileName: The name of the process image.
- Offset(V): The virtual address of the process image in memory.
- Threads: The number of threads that are associated with the process.
- Handles: The number of handles that are open by the process.
- SessionId: The session ID of the process.
- Wow64: A flag indicating whether the process is running in 32-bit mode.
- CreateTime: The time at which the process was created.
- ExitTime: The time at which the process exited.
- File output: The path to the file where the process information was written.
- This information can be used to help you investigate the crash and determine what caused it. For example, you can look at the listof processes that were running at the time of the crash to see if any of them are known to be malicious.
- You can also look at the handles that are open by the process to see if any of them are to files that are known to be malicious.
- In addition to the information that is displayed by the windows.pslist plugin, you can also use the volatility framework to extract additional information about the processes that were running at the time of the crash.
- For example, you can use the windows.handles plugin to extract a list of all the handles that are open by a process. You can alsouse the windows.modules plugin to extract a list of all the modules that are loaded into a process.
- This additional information can be used to help you investigate the crash and determine what caused it.

The command python vol.py -f C:\Users\VIT-AP\Downloads\volatility3\volatility3\Win10Home-20H2-64bit-memdump.mem windows.pslist | Select-String chrome is using Volatility to list the running processes in the Windows system captured in the memory dump file and filtering the output to only show processes that have "chrome" in their name

```
PS C:\Users\VIT-AP\downloads\volatility3> python vol.py -f C:\Users\VIT-AP\Downloads\volatility3\volatility3\Win10Home-20H2-64bit-memdump.mem windows.pslist | Select-String
chrome | more

1328    4352    chrome.exe      0xbf0f6d53e080  26      -       1       False   2021-04-30 17:44:52.000000      N/A     Disabled
6764    1328    chrome.exe      0xbf0f6d748080  7       -       1       False   2021-04-30 17:44:52.000000      N/A     Disabled
4508    1328    chrome.exe      0xbf0f6be38340  10      -       1       False   2021-04-30 17:44:53.000000      N/A     Disabled
1840    1328    chrome.exe      0xbf0f6cbc5080  12      -       1       False   2021-04-30 17:44:53.000000      N/A     Disabled
5492    1328    chrome.exe      0xbf0f6a897080  7       -       1       False   2021-04-30 17:44:53.000000      N/A     Disabled
2432    1328    chrome.exe      0xbf0f6a9e7080  19      -       1       False   2021-04-30 17:44:53.000000      N/A     Disabled
5888    1328    chrome.exe      0xbf0f6d2ea340  13      -       1       False   2021-04-30 17:44:57.000000      N/A     Disabled
460     1328    chrome.exe      0xbf0f6d591300  12      -       1       False   2021-04-30 17:48:07.000000      N/A     Disabled
4108    1328    chrome.exe      0xbf0f667f1300  12      -       1       False   2021-04-30 17:48:07.000000      N/A     Disabled
3380    1328    chrome.exe      0xbf0f6d182080  6       -       1       False   2021-04-30 17:48:07.000000      N/A     Disabled
```

These are the processes with "chrome" in their name, likely referring to instances of the Google Chrome browser running in the captured system. The output provides information such as the PID (Process ID), PPID (Parent Process ID), ImageFileName (Name of the executable file), Offset(V) (Virtual address offset), Threads,Handles, SessionId, Wow64 (if the process is running in 32-bit compatibility mode on a 64-bit system), CreateTime, ExitTime (if applicable), and File output status.

The command python vol.py -f C:\Users\VIT-AP\Downloads\volatility3\volatility3\Win10Home-20H2-64bit-memdump.mem windows.handles is using Volatility to retrieve information about the handles (open resources) in the Windows system capturedin the memory dump file.

```
PS C:\Users\VIT-AP\downloads\volatility3> python vol.py -f C:\Users\VIT-AP\Downloads\volatility3\volatility3\Win10Home-20H2-64bit-memdump.mem windows.handles | more
Volatility 3 Framework 2.4.2

PID     Process Offset  HandleValue     Type    GrantedAccess   Name

4       System  0xbf0f64a63080  0x4     Process 0x1fffff        System Pid 4
4       System  0xbf0f64b4a140  0x8     Thread  0x1fffff        Tid 28 Pid 4
4       System  0xbf0f64aad0a0  0x10    Mutant  0x1f0001        BcdSyncMutant
4       System  0xa8032f041cd0  0x14    Directory       0xf000f GLOBAL??
4       System  0xa8032f032520  0x18    Directory       0xf000f
4       System  0xbf0f64a8ba40  0x1c    Partition       0x1f0003        MemoryPartition0
4       System  0xa8032f0326f0  0x20    Directory       0xf000f KernelObjects
4       System  0xbf0f64a74920  0x24    Event   0x1f0003        LowPagedPoolCondition
4       System  0xbf0f64a74ca0  0x28    Event   0x1f0003        HighPagedPoolCondition
4       System  0xbf0f64a74720  0x2c    Event   0x1f0003        LowNonPagedPoolCondition
4       System  0xbf0f64a74d20  0x30    Event   0x1f0003        HighNonPagedPoolCondition
4       System  0xbf0f64a747a0  0x34    Event   0x1f0003        LowMemoryCondition
4       System  0xbf0f64a74320  0x38    Event   0x1f0003        HighMemoryCondition
4       System  0xbf0f64a74a20  0x3c    Event   0x1f0003        LowCommitCondition
4       System  0xbf0f64a74da0  0x40    Event   0x1f0003        HighCommitCondition
4       System  0xbf0f64a74ea0  0x44    Event   0x1f0003        MaximumCommitCondition
4       System  0xbf0f64a748a0  0x48    Event   0x1f0003        MemoryErrors
4       System  0xbf0f64a746a0  0x4c    Event   0x1f0003        PhysicalMemoryChange
4       System  0xbf0f64bc1080  0x50    Thread  0x1fffff        Tid 92 Pid 4
4       System  0xbf0f64bc6040  0x54    Process 0x1fffff        Registry Pid 108
4       System  0xa8032f0f9e80  0x58    Key     0x2001f MACHINE\SYSTEM\CONTROLSET001\CONTROL\HIVELIST
```

- **The output of the command you ran shows a list of all the handles that are open by the system process. The following isa breakdown of the information that is being displayed:**
- PID: The process ID of the process that has the handle open.
- Process Offset: The virtual address of the process image in memory.
- HandleValue: The value of the handle.
- Type: The type of handle.
- GrantedAccess: The access that is granted to the handle.
- Name: The name of the object that the handle is open to.
- This information can be used to help you investigate the crash and determine what caused it.
- For example, you can look at the list of handles that are open by the process to see if any of them are to files that are known to be malicious. You can also look at the access that is granted to the handles to see if any of them are excessive.

```
PS C:\Users\VIT-AP\downloads\volatility3> python vol.py -f C:\Users\VIT-AP\Downloads\volatility3\volatility3\Win10Home-20H2-64bit-memdump.mem windows.handles | Select-String File | more

4       System  0xbf0f6660b1e0  0x64    File    0x12019f        \Device\HarddiskVolume2\$Extend\$RmMetadata\$TxfLog\$TxfLogContainer00000000000000000002
4       System  0xbf0f6a357a20  0xd0    File    0x150003        \Device\HarddiskVolume2\DumpStack.log.tmp
4       System  0xa8032f0b5db0  0xd4    Key     0xf003f MACHINE\SYSTEM\CONTROLSET001\HARDWARE PROFILES
4       System  0xbf0f66995ef0  0xe4    File    0x13008b        \Device\HarddiskVolume2\Windows\System32\LogFiles\WMI\RtBackup\EtwRTEventlog-Security.etl
4       System  0xbf0f668de470  0x144   File    0x12019f        \Device\Tcp
4       System  0xa803326964f0  0x170   Key     0x9     MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\IMAGE FILE EXECUTION OPTIONS
4       System  0xbf0f6a357ed0  0x174   File    0x140003        \Device\HarddiskVolume2\swapfile.sys
4       System  0xa8032f4fa3e0  0x190   Key     0x8     MACHINE\SYSTEM\CONTROLSET001\CONTROL\POWER\PROFILE\EVENTS\{54533251-82BE-4824-96C1-47B60B740D00}\{0DA965DC-8FCF-4C0B-8EFE-8DD5E7BC959A}\{7E01ADEF-81E6-4E1B
-8075-56F373584694}
4       System  0xa8032f4fbb40  0x194   Key     0x8     MACHINE\SYSTEM\CONTROLSET001\CONTROL\POWER\PROFILE\EVENTS\{54533251-82BE-4824-96C1-47B60B740D00}\{0AABB002-A307-447E-9B81-1D819DF6C6D0}\{CE74AA52-A71A-4036
-BEEF-B6C411010E28}
4       System  0xa8032f4f53d0  0x198   Key     0x8     MACHINE\SYSTEM\CONTROLSET001\CONTROL\POWER\PROFILE\EVENTS\{54533251-82BE-4824-96C1-47B60B740D00}\{8BC6262C-C026-411D-AE3B-7E2F70811A13}\{C072EEBB-1955-4FA9
-B4BA-421E96E1D674}
4       System  0xa8032f4fa600  0x19c   Key     0x8     MACHINE\SYSTEM\CONTROLSET001\CONTROL\POWER\PROFILE\EVENTS\{54533251-82BE-4824-96C1-47B60B740D00}\{D4140C81-EBBA-4E60-8561-6918290359CD}\{35037BB4-9528-481D
-8CB2-8FCC63A9DD81}
4       System  0xa8032f4fb920  0x1a0   Key     0x8     MACHINE\SYSTEM\CONTROLSET001\CONTROL\POWER\PROFILE\EVENTS\{54533251-82BE-4824-96C1-47B60B740D00}\{EE1E4F72-E368-46B1-B3C6-5048B11C2DBD}\{9C1F0DBA-33E9-43AF
-9EDA-A607AA5139DA}
4       System  0xa8032f4fab50  0x1ac   Key     0x10    MACHINE\SYSTEM\CONTROLSET001\CONTROL\FILESYSTEM
4       System  0xa8032f0e8360  0x1d8   Key     0x10    MACHINE\SYSTEM\CONTROLSET001\CONTROL\FILESYSTEM
4       System  0xbf0f668bfa90  0x214   File    0x12019f        \Device\HarddiskVolume1\$Extend\$RmMetadata\$TxfLog\$TxfLog.blf
4       System  0xbf0f6660a0a0  0x218   File    0x12019f        \Device\HarddiskVolume2\$Extend\$RmMetadata\$TxfLog\$TxfLog.blf
4       System  0xbf0f6660a380  0x220   File    0x12019f        \Device\HarddiskVolume2\$Extend\$RmMetadata\$TxfLog\$TxfLogContainer00000000000000000005
4       System  0xbf0f6660a210  0x228   File    0x1     \Device\HarddiskVolume2\$Extend\$RmMetadata\$Txf:$I30:$INDEX_ALLOCATION
4       System  0xbf0f6660c380  0x244   File    0x120089        \Device\HarddiskVolume2\Windows\System32\drivers\en-US\ntfs.sys.mui
```

- **The output of the command you ran shows a list of all the handles that are open to files by the system process.**
- **The following is a breakdown of the information that is being displayed:**
- PID: The process ID of the process that has the handle open.
- Process Offset: The virtual address of the process image in memory.
- HandleValue: The value of the handle. Type: The type of handle.
- GrantedAccess: The access that is granted to the handle.
- Name: The name of the object that the handle is open to.
- This information can be used to help you investigate the crash and determine what caused it.
- For example, you can look at the list of files that are open by the process to see if any of them are known to be malicious.You can also look at the access that is granted to the files to see if any of it is excessive.
- **Here are the some file-related handles obtained from the output:**
  \Device\HarddiskVolume2$Extend$RmMetadata$TxfLog$TxfLogContainer00000000000000000002
  \Device\HarddiskVolume2\DumpStack.log.tmp
  \Device\HarddiskVolume2\Windows\System32\LogFiles\WMI\RtBackup\EtwRTEventlog-Security.etl
  \Device\HarddiskVolume2\swapfile.sys \Device\HarddiskVolume1$Extend$RmMetadata$TxfLog$TxfLog.blf
  \Device\HarddiskVolume2$Extend$RmMetadata$TxfLog$TxfLog.blf

File handles related of any execution:

```
PS C:\Users\VIT-AP\downloads\volatility3> python vol.py -f C:\Users\VIT-AP\Downloads\volatility3\volatility3\Win10Home-20H2-64bit-memdump.mem windows.handles --pid 1328 | more
Volatility 3 Framework 2.4.2

PID     Process Offset  HandleValue     Type    GrantedAccess   Name

1328    chrome.exe      0xbf0f6cb02260  0x4     Event   0x1f0003
1328    chrome.exe      0xbf0f6cb03360  0x8     Event   0x1f0003
1328    chrome.exe      0xbf0f6d0547c0  0xc     WaitCompletionPacket    0x1
1328    chrome.exe      0xbf0f6ab07900  0x10    IoCompletion    0x1f0003
1328    chrome.exe      0xbf0f6d0057b0  0x14    TpWorkerFactory 0xf00ff
1328    chrome.exe      0xbf0f6ce2cb50  0x18    IRTimer 0x100002
1328    chrome.exe      0xbf0f6d0539f0  0x1c    WaitCompletionPacket    0x1
1328    chrome.exe      0xbf0f6ce2b830  0x20    IRTimer 0x100002
1328    chrome.exe      0xbf0f6d055320  0x24    WaitCompletionPacket    0x1
1328    chrome.exe      0xbf0f6ab07d50  0x28    EtwRegistration 0x804
1328    chrome.exe      0xbf0f6ab07b90  0x2c    EtwRegistration 0x804
1328    chrome.exe      0xbf0f6ab08290  0x30    EtwRegistration 0x804
1328    chrome.exe      0xa8032f8100c0  0x34    Directory       0x3     KnownDlls
1328    chrome.exe      0xbf0f6cb03960  0x38    Event   0x1f0003
1328    chrome.exe      0xbf0f6cb03b60  0x3c    Event   0x1f0003
1328    chrome.exe      0xbf0f6ab232d0  0x40    EtwRegistration 0x804
1328    chrome.exe      0xbf0f6ab07e30  0x44    EtwRegistration 0x804
1328    chrome.exe      0xbf0f6ab08530  0x48    EtwRegistration 0x804
1328    chrome.exe      0xbf0f6c567070  0x4c    ALPC Port       0x1f0001
```

- **Here are the some handles associated with the process ID (PID) 1328:**
- HandleValue: 0x4, Type: Event, GrantedAccess: 0x1f0003
- HandleValue: 0x8, Type: Event, GrantedAccess: 0x1f0003
- HandleValue: 0xc, Type: WaitCompletionPacket, GrantedAccess: 0x1
- HandleValue: 0x10, Type: IoCompletion, GrantedAccess: 0x1f0003
- HandleValue: 0x14, Type: TpWorkerFactory, GrantedAccess: 0xf00ff
- HandleValue: 0x18, Type: IRTimer, GrantedAccess: 0x100002
- HandleValue: 0x1c, Type: WaitCompletionPacket, GrantedAccess: 0x1
- HandleValue: 0x20, Type: IRTimer, GrantedAccess: 0x100002
- HandleValue: 0x24, Type: WaitCompletionPacket, GrantedAccess: 0x1
- This additional information can be used to help you investigate the crash and determine what caused it.
- The handles that are listed in the output of the command you ran are all associated with the chrome.exe process.
- The chrome.exe process is a web browser that is used to display web pages. The handles that are open to files are all related to the chrome.exe process and its operation.
- The fact that these handles are open to files that are related to the chrome.exe process suggests that the chrome.exe process is functioning normally. However, it is always a good idea to investigate any crash that occurs, even if it appears tobe caused by a normal process.

File handles of chrome execution:

```
PS C:\Users\VIT-AP\downloads\volatility3> python vol.py -f C:\Users\VIT-AP\Downloads\volatility3\volatility3\Win10Home-20H2-64bit-memdump.mem windows.handles --pid 1328 | Select-String File | more

1328    chrome.exe      0xbf0f6ac8c9f0  0x70    File    0x100020        \Device\HarddiskVolume2\Program Files\Google\Chrome\Application\90.0.4430.93
1328    chrome.exe      0xbf0f6ac8a470  0x8c    File    0x100001        \Device\KsecDD
1328    chrome.exe      0xbf0f6ac8bd70  0x9c    File    0x100001        \Device\CNG
1328    chrome.exe      0xa80337116510  0xac    Key     0x9     MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\IMAGE FILE EXECUTION OPTIONS
1328    chrome.exe      0xbf0f6abe2080  0x294   File    0x120089        \Device\HarddiskVolume2\Program Files\Google\Chrome\Application\90.0.4430.93\icudtl.dat
1328    chrome.exe      0xbf0f6abe2850  0x29c   File    0x120089        \Device\HarddiskVolume2\Program Files\Google\Chrome\Application\90.0.4430.93\v8_context_snapshot.bin
1328    chrome.exe      0xbf0f6abe3660  0x33c   File    0x12019f        \Device\HarddiskVolume2\Users\John Doe\AppData\Local\Google\Chrome\User Data\BrowserMetrics\BrowserMetrics-608C4214-530.pma
1328    chrome.exe      0xbf0f6abe3b10  0x344   File    0x120089        \Device\HarddiskVolume2\Program Files\Google\Chrome\Application\90.0.4430.93\chrome_100_percent.pak
1328    chrome.exe      0xbf0f6abe3e30  0x34c   File    0x120089        \Device\HarddiskVolume2\Program Files\Google\Chrome\Application\90.0.4430.93\chrome_200_percent.pak
1328    chrome.exe      0xbf0f6abe4150  0x354   File    0x120089        \Device\HarddiskVolume2\Program Files\Google\Chrome\Application\90.0.4430.93\Locales\en-US.pak
1328    chrome.exe      0xbf0f6abc6970  0x35c   File    0x120089        \Device\HarddiskVolume2\Program Files\Google\Chrome\Application\90.0.4430.93\resources.pak
1328    chrome.exe      0xbf0f6abe4600  0x444   File    0x100020        \Device\HarddiskVolume2\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.19041.844_none_ca00b6081b84eb1d
1328    chrome.exe      0xbf0f6c5bb2a0  0x474   File    0x100080        \Device\Nsi
1328    chrome.exe      0xbf0f6db9c240  0x490   File    0x100020        \Device\HarddiskVolume2\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.19041.844_none_ca00b6081b84eb1d
1328    chrome.exe      0xbf0f6ac8b280  0x5a4   File    0x100020        \Device\HarddiskVolume2\Program Files\Google\Chrome\Application\90.0.4430.93
1328    chrome.exe      0xbf0f6db9d1e0  0x5e4   File    0x12019f        \Device\HarddiskVolume2\Users\John Doe\AppData\Local\Microsoft\Windows\Explorer\iconcache_32.db
1328    chrome.exe      0xbf0f6db935a0  0x628   File    0x100001        \Device\HarddiskVolume2\Windows\System32\drivers\etc
1328    chrome.exe      0xa803377efc10  0x638   Key     0x20019 USER\S-1-5-21-3061953532-2461696977-1363062292-1001\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\EXPLORER\FILEEXTS
1328    chrome.exe      0xbf0f6abe2d00  0x698   File    0x120089        \Device\DeviceApi\CMApi
1328    chrome.exe      0xbf0f6abe8160  0x6b4   File    0x120089        \Device\DeviceApi\CMNotify
```

- The files that are open by the chrome.exe process are all related to the operation of the web browser.
- The files that are open to files that are related to the chrome.exe process suggests that the chrome.exeprocess is functioning normally.
- However, it is always a good idea to investigate any crash that occurs, even if it appears to be caused by a normal process.
- The files that are open by the chrome.exe process include: \Device\HarddiskVolume2\Program Files\Google\Chrome\Application\90.0.4430.93: This is the directory where the chrome.exe process islocated.
- \Device\HarddiskVolume2\Users\John Doe\AppData\Local\Google\Chrome\User Data\BrowserMetrics\BrowserMetrics-608C4214-530.
- pma: This is a file that contains performance metrics for the chrome.exe process. \Device\HarddiskVolume2\Program Files\Google\Chrome\Application\90.0.4430.93\icudtl.
- dat: This is a file that contains data for the International Components for Unicode (ICU) library.

List of all the open network connections onthe system

```
PS C:\Users\VIT-AP\downloads\volatility3> python vol.py -f C:\Users\VIT-AP\Downloads\volatility3\volatility3\Win10Home-20H2-64bit-memdump.mem windows.netstat | more
Volatility 3 Framework 2.4.2

Offset  Proto   LocalAddr       LocalPort       ForeignAddr     ForeignPort     State   PID     Owner   Created

0xbf0f6a535aa0  TCPv4   10.0.2.15       49846   96.90.32.107    7680    SYN_SENT        2116    svchost.exe     2021-04-30 17:52:01.000000
0xbf0f6d8a1010  TCPv4   10.0.2.15       49771   185.70.41.35    443     CLOSE_WAIT      1840    chrome.exe      2021-04-30 17:44:57.000000
0xbf0f6cbb9530  TCPv4   10.0.2.15       49772   185.70.41.35    443     FIN_WAIT2       1840    chrome.exe      2021-04-30 17:44:58.000000
0xbf0f6d0c64a0  TCPv4   10.0.2.15       49843   204.79.197.222  443     ESTABLISHED     5104    SearchApp.exe   2021-04-30 17:51:26.000000
0xbf0f6ad1fad0  TCPv4   10.0.2.15       49847   52.230.222.68   443     ESTABLISHED     2812    svchost.exe     2021-04-30 17:52:17.000000
0xbf0f6ca71a20  TCPv4   10.0.2.15       49769   142.250.190.14  443     CLOSE_WAIT      1840    chrome.exe      2021-04-30 17:44:55.000000
0xbf0f6cfd17f0  TCPv4   10.0.2.15       49777   35.186.220.63   443     CLOSE_WAIT      1840    chrome.exe      2021-04-30 17:44:58.000000
0xbf0f6ad16050  TCPv4   10.0.2.15       49829   142.250.191.208 443     ESTABLISHED     5624    svchost.exe     2021-04-30 17:49:58.000000
0xbf0f6c85bb20  TCPv4   10.0.2.15       49775   185.70.41.35    443     FIN_WAIT2       1840    chrome.exe      2021-04-30 17:44:58.000000
0xbf0f6d51c4a0  TCPv4   10.0.2.15       49838   13.107.3.254    443     ESTABLISHED     5104    SearchApp.exe   2021-04-30 17:51:23.000000
0xbf0f6d5c8a70  TCPv4   10.0.2.15       49797   172.217.4.74    443     CLOSE_WAIT      1840    chrome.exe      2021-04-30 17:48:27.000000
0xbf0f6d51c010  TCPv4   10.0.2.15       49763   172.217.4.35    443     CLOSE_WAIT      1840    chrome.exe      2021-04-30 17:44:53.000000
0xbf0f6c6352b0  TCPv4   10.0.2.15       49842   52.113.196.254  443     ESTABLISHED     5104    SearchApp.exe   2021-04-30 17:51:25.000000
0xbf0f6d525a20  TCPv4   10.0.2.15       49845   23.101.202.202  443     ESTABLISHED     1156    MsMpEng.exe     2021-04-30 17:51:36.000000
0xbf0f6a896ae0  TCPv4   10.0.2.15       49773   185.70.41.35    443     FIN_WAIT2       1840    chrome.exe      2021-04-30 17:44:58.000000
0xbf0f6d5d1ac0  TCPv4   10.0.2.15       49770   185.70.41.35    443     FIN_WAIT2       1840    chrome.exe      2021-04-30 17:44:57.000000
0xbf0f6cd4fa20  TCPv4   10.0.2.15       49837   204.79.197.200  443     ESTABLISHED     5104    SearchApp.exe   2021-04-30 17:51:18.000000
0xbf0f6c7104d0  TCPv4   10.0.2.15       49778   185.70.41.130   443     ESTABLISHED     1840    chrome.exe      2021-04-30 17:45:00.000000
```

- **The output of the command you ran shows a list of all the open network connections on the system.**
  The following is a breakdown of the information that is being displayed:
- Offset: The offset of the network connection in the memory dump.
- Proto: The protocol that is being used for the network connection. LocalAddr: The local address ofthe network connection.
- LocalPort: The local port of the network connection.
- ForeignAddr: The foreign address of the network connection.
- ForeignPort: The foreign port of the network connection.
- State: The state of the network connection.
- PID: The process ID of the process that owns the network connection.
- Owner: The name of the process that owns the network connection.
- Created: The time that the network connection was created.
- The network connections that are listed in the output of the command you ran are all essential for theoperation of the system.
- The fact that these connections are open suggests that the system is functioning normally.
- However, it is always a good idea to investigate any crash that occurs, even if it appears to be caused by a normal process.

Network connections associated with the"chrome.exe" process

```
PS C:\Users\VIT-AP\downloads\volatility3> python vol.py -f C:\Users\VIT-AP\Downloads\volatility3\volatility3\Win10Home-20H2-64bit-memdump.mem windows.netstat | Select-String chrome | more

0xbf0f6d8a1010  TCPv4   10.0.2.15       49771   185.70.41.35    443     CLOSE_WAIT      1840    chrome.exe      2021-04-30 17:44:57.000000
0xbf0f6cbb9530  TCPv4   10.0.2.15       49772   185.70.41.35    443     FIN_WAIT2       1840    chrome.exe      2021-04-30 17:44:58.000000
0xbf0f6ca71a20  TCPv4   10.0.2.15       49769   142.250.190.14  443     CLOSE_WAIT      1840    chrome.exe      2021-04-30 17:44:55.000000
0xbf0f6cfd17f0  TCPv4   10.0.2.15       49777   35.186.220.63   443     CLOSE_WAIT      1840    chrome.exe      2021-04-30 17:44:58.000000
0xbf0f6c85bb20  TCPv4   10.0.2.15       49775   185.70.41.35    443     FIN_WAIT2       1840    chrome.exe      2021-04-30 17:44:58.000000
0xbf0f6d5c8a70  TCPv4   10.0.2.15       49797   172.217.4.74    443     CLOSE_WAIT      1840    chrome.exe      2021-04-30 17:48:27.000000
0xbf0f6d51c010  TCPv4   10.0.2.15       49763   172.217.4.35    443     CLOSE_WAIT      1840    chrome.exe      2021-04-30 17:44:53.000000
0xbf0f6a896ae0  TCPv4   10.0.2.15       49773   185.70.41.35    443     FIN_WAIT2       1840    chrome.exe      2021-04-30 17:44:58.000000
0xbf0f6d5d1ac0  TCPv4   10.0.2.15       49770   185.70.41.35    443     FIN_WAIT2       1840    chrome.exe      2021-04-30 17:44:57.000000
0xbf0f6c7104d0  TCPv4   10.0.2.15       49778   185.70.41.130   443     ESTABLISHED     1840    chrome.exe      2021-04-30 17:45:00.000000
0xbf0f6aa4b320  UDPv4   0.0.0.0 5353    *       0               1328    chrome.exe      2021-04-30 17:45:04.000000
0xbf0f6aa4a6a0  UDPv4   0.0.0.0 5353    *       0               1328    chrome.exe      2021-04-30 17:45:04.000000
0xbf0f6aa4a6a0  UDPv6   ::      5353    *       0               1328    chrome.exe      2021-04-30 17:45:04.000000
0xbf0f6aa4f010  UDPv4   0.0.0.0 5353    *       0               1328    chrome.exe      2021-04-30 17:45:04.000000
0xbf0f6aa4f010  UDPv6   ::      5353    *       0               1328    chrome.exe      2021-04-30 17:45:04.000000
```

- There are several TCPv4 connections with the local IP address 10.0.2.15 and different local ports.The connections are in various states such as CLOSE_WAIT and FIN_WAIT2.
- They are established with remote IP addresses on port 443.
- There is an established TCPv4 connection between 10.0.2.15 (local IP) on port 49778 and the

  remote IP address 185.70.41.130 on port 443.
- This connection is associated with the "chrome.exe" process and was created at 2021-04-3017:45:00.000000.
- There are also some UDPv4 and UDPv6 connections associated with "chrome.exe" on port 5353.

VOLDIFF

VolDiff is a Python script that uses the Volatility memory analysis framework to analyze two memory dumps and output the differences between them.

When applied to memory analysis, this script will focus your attention on memory artifacts generated after, and possibly as a result of, code execution.

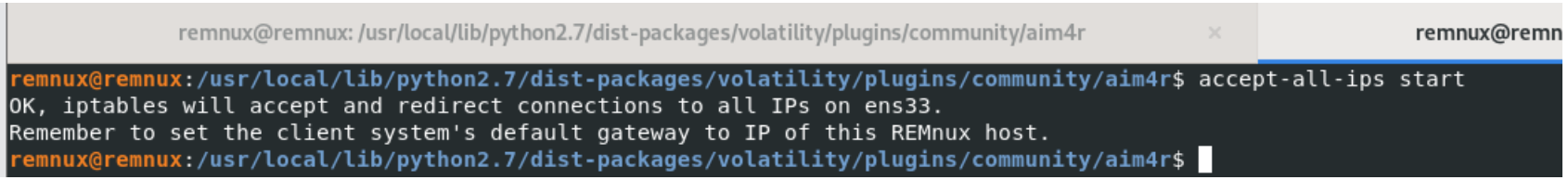
This can expedite your analysis of large memory dumps to detect activity such as code injection and provide visibility into packed or obfuscated code.

However, keep in mind that memory is in a state of flux, so changes included in the diff results are not necessarily caused by executing the suspect file.

I used a file named funfile.exe, you can download the sample (password: infected).

Some initial behavioral analysis indicated that this sample generated network traffic to an IP address. Since our goal is to assess memory artifacts, I chose to launch several “fake” services in REMnux to encourage activity. Specifically, I ran the following from a REMnux terminal:

accept-all-ips start: This bash shell script written by Lenny Zeltser redirects all network traffic destined for an IP address to the REMnux VM.



To compare memory dumps using VolDiff, we need to capture a memory image before and after infecting a sacrificial host. With VMware, one approach to obtaining a memory image is to use the snapshot feature. Whenever a snapshot is created, VMware saves a “.vmem” file that includes the contents of memory at the time the snapshot was created. This file can then be analyzed using a memory analysis tool like Volatility. To create the memory dumps VolDiff requires, I followed these steps:

- I copied funfile.exe to the Windows VM desktop.
- I created a VM snapshot and noted the new “.vmem” file name on my host.
- In the Windows VM, I right-clicked funfile.exe and selected “Run as administrator” to execute the sample with admin rights.
- After giving the sample a couple minutes to run, I created another VM snapshot and noted this second “.vmem” file name.

VolDiff against my two memory dumps, I ran the command shown below. Note that the command requires the correct OS profile for the memory images.

```
                _    ___ _  __  __
 /\   /\___ | |  /   (_)/ _|/ _|
 \ \ / / _ \| | / /\ / | |_| |_
  \ V / (_) | |/ /_//| |  _|  _|
   \_/ \___/|_/___,' |_|_| |_|

Volatility analysis report generated by VolDiff v2.1.4
Download the latest VolDiff version from https://github.com/aim4r/VolDiff/

Baseline memory image: /mnt/hgfs/windows7/baseline.vmem
Infected memory image: /mnt/hgfs/windows7/infected.vmem
Profile: Win7SP1x64
Date and time: 02/07/2023 08:16
```

New pslist entries.

```
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x874fecc0 svchost.exe            2976   2968      5        0      1      0 2015-06-14 16:12:10 UTC+0000
0x874d8cc0 svchost.exe            3000    548      3        0      0      0 2015-06-14 16:12:10 UTC+0000
```

New psscan entries.

```
Offset(P)          Name                PID   PPID PDB        Time created                   Time exited
------------------ ---------------- ------ ------ ---------- ------------------------------ ------------------------------
0x000000007cad8cc0 svchost.exe        3000    548 0x7e23a2c0 2015-06-14 16:12:10 UTC+0000
0x000000007cafecc0 svchost.exe        2976   2968 0x7e23a2a0 2015-06-14 16:12:10 UTC+0000
```

New psxview entries.

```
Offset(P)  Name                    PID pslist psscan thrdproc pspcid csrss session deskthrd ExitTime
---------- -------------------- ------ ------ ------ -------- ------ ----- ------- -------- --------
0x7dae7cc0 svchost.exe             904 True   True   True     False  True  True    False
0x7db67cc0 svchost.exe            1204 True   False  True     False  True  True    False
0x7cad8cc0 svchost.exe            3000 True   True   True     False  True  True    False
0x7cafecc0 svchost.exe            2976 True   True   True     False  True  True    False
```

New netscan entries.

```
Offset(P)          Proto    Local Address                  Foreign Address      State            Pid      Owner          Created
0x7c668470         UDPv4    0.0.0.0:0                      *:*                                   0        ?A????       2015-06-14 16:07:20 UTC+0000
0x7cadb858         TCPv4    172.16.240.128:49159           -:443                ESTABLISHED      2976     svchost.exe
```

```
New malfind entries.
================================================================================================================
Process: svchost.exe Pid: 2976 Address: 0xed0000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: PrivateMemory: 1, Protection: 6

0x00ed0000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x00ed0010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x00ed0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00ed0030  00 00 00 00 00 00 00 00 00 00 00 00 c8 00 00 00   ................

0xed0000 4d              DEC EBP
0xed0001 5a              POP EDX
0xed0002 90              NOP
0xed0003 0003            ADD [EBX], AL
0xed0005 0000            ADD [EAX], AL
0xed0007 000400          ADD [EAX+EAX], AL
0xed000a 0000            ADD [EAX], AL
0xed000c ff              DB 0xff
0xed000d ff00            INC DWORD [EAX]
0xed000f 00b800000000    ADD [EAX+0x0], BH
0xed0015 0000            ADD [EAX], AL
0xed0017 004000          ADD [EAX+0x0], AL
0xed001a 0000            ADD [EAX], AL
0xed001c 0000            ADD [EAX], AL
0xed001e 0000            ADD [EAX], AL
0xed0020 0000            ADD [EAX], AL
0xed0022 0000            ADD [EAX], AL
0xed0024 0000            ADD [EAX], AL
0xed0026 0000            ADD [EAX], AL
0xed0028 0000            ADD [EAX], AL
0xed002a 0000            ADD [EAX], AL
0xed002c 0000            ADD [EAX], AL
0xed002e 0000            ADD [EAX], AL
0xed0030 0000            ADD [EAX], AL
0xed0032 0000            ADD [EAX], AL
0xed0034 0000            ADD [EAX], AL
0xed0036 0000            ADD [EAX], AL
0xed0038 0000            ADD [EAX], AL
0xed003a 0000            ADD [EAX], AL
0xed003c c8000000        ENTER 0x0, 0x0

Process: svchost.exe Pid: 2976 Address: 0x12d0000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: PrivateMemory: 1, Protection: 6

0x012d0000  0f 01 0d 30 fa e9 00 c3 00 00 00 00 00 00 00 00   ...0............
0x012d0010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x012d0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x012d0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

New sessions entries.

```
 Process: 3000 svchost.exe 2015-06-14 16:12:10 UTC+0000
Session(V): 8dccc000 ID: 1 Processes: 7
 Process: 2976 svchost.exe 2015-06-14 16:12:10 UTC+0000
```

New messagehooks entries.

Offset(V)	Sess	Desktop	Thread	Filter	Flags	Function	Module
0x90998b38	0	Service-0...\Default	3008 (svchost.exe 3000)	WH_KEYBOARD	5570628	0x0042004b	0x2e0053L
0x90998b38	0	Service-0...\Default	2144 (svchost.exe 856)	WH_KEYBOARD	5570628	0x0042004b	0x2e0053L
0x90998b38	0	Service-0...\Default	1364 (svchost.exe 856)	WH_KEYBOARD	5570628	0x0042004b	0x2e0053L
0x90998b38	0	Service-0...\Default	2680 (vmtoolsd.exe 1828)	WH_KEYBOARD	5570628	0x0042004b	0x2e0053L
0x909c7c18	1	WinSta0\Default	2980 (svchost.exe 2976)	WH_SYSMSGFILTER	5439565	0x00000000	0x540043L
0x909c7c18	1	WinSta0\Default	2612 (vmtoolsd.exe 2600)	WH_SYSMSGFILTER	5439565	0x00000000	0x540043L
0x909c7c18	1	WinSta0\Default	2616 (vmtoolsd.exe 2600)	WH_SYSMSGFILTER	5439565	0x00000000	0x540043L

New cmdline entries.

```
**********************************************************************
svchost.exe pid:   2976
Command line : svchost.exe
**********************************************************************
svchost.exe pid:   3000
Command line : C:\Windows\System32\svchost.exe -k WerSvcGroup
```

New driverscan entries.

Offset(P)	#Ptr	#Hnd	Start	Size	Service Key	Name	Driver Name
0x000000007c40e5a8	3	0	0x805d3000	0x1a000		???	??
0x000000007c413790	7	0	0x805ed000	0x7000		???	
0x000000007c41b670	3	0	0xa1003000	0xa1000		?h???	(08X
0x000000007c41f818	4	0	0xa10a4000	0xa000			??
0x000000007c578520	3	0	0x80400000	0x25000			?????????
0x000000007e619840	13	0	0x821ba000	0x46000	FltMgr	FltMgr	\FileSystem\FltMgr
0x000000007e877d18	3	0	0xa19e4000	0xd000	condrv	condrv	\Driver\condrv

New driverscan entries.

Offset(P)	#Ptr	#Hnd	Start	Size	Service Key	Name	Driver Name
0x000000007c40e5a8	3	0	0x805d3000	0x1a000		???	??
0x000000007c413790	7	0	0x805ed000	0x7000		???	
0x000000007c41b670	3	0	0xa1003000	0xa1000		?h???	(08X
0x000000007c41f818	4	0	0xa10a4000	0xa000			??
0x000000007c578520	3	0	0x80400000	0x25000			?????????
0x000000007e619840	13	0	0x821ba000	0x46000	FltMgr	FltMgr	\FileSystem\FltMgr
0x000000007e877d18	3	0	0xa19e4000	0xd000	condrv	condrv	\Driver\condrv

New driverirp entries.

DriverName: condrv
DriverStart: 0xa19e4000
DriverSize: 0xd000
DriverStartIo: 0x0

0	IRP_MJ_CREATE	0xa19e8420	condrv.sys
1	IRP_MJ_CREATE_NAMED_PIPE	0x8192148e	ntoskrnl.exe
2	IRP_MJ_CLOSE	0xa19e82f0	condrv.sys
3	IRP_MJ_READ	0xa19ec778	condrv.sys
4	IRP_MJ_WRITE	0xa19e86b0	condrv.sys
5	IRP_MJ_QUERY_INFORMATION	0x8192148e	ntoskrnl.exe
6	IRP_MJ_SET_INFORMATION	0x8192148e	ntoskrnl.exe
7	IRP_MJ_QUERY_EA	0x8192148e	ntoskrnl.exe
8	IRP_MJ_SET_EA	0x8192148e	ntoskrnl.exe
9	IRP_MJ_FLUSH_BUFFERS	0xa19ec72a	condrv.sys
10	IRP_MJ_QUERY_VOLUME_INFORMATION	0x8192148e	ntoskrnl.exe
11	IRP_MJ_SET_VOLUME_INFORMATION	0x8192148e	ntoskrnl.exe
12	IRP_MJ_DIRECTORY_CONTROL	0x8192148e	ntoskrnl.exe
13	IRP_MJ_FILE_SYSTEM_CONTROL	0x8192148e	ntoskrnl.exe
14	IRP_MJ_DEVICE_CONTROL	0xa19e9610	condrv.sys
15	IRP_MJ_INTERNAL_DEVICE_CONTROL	0x8192148e	ntoskrnl.exe
16	IRP_MJ_SHUTDOWN	0x8192148e	ntoskrnl.exe
17	IRP_MJ_LOCK_CONTROL	0x8192148e	ntoskrnl.exe
18	IRP_MJ_CLEANUP	0xa19e8670	condrv.sys
19	IRP_MJ_CREATE_MAILSLOT	0x8192148e	ntoskrnl.exe
20	IRP_MJ_QUERY_SECURITY	0xa19e8010	condrv.sys
21	IRP_MJ_SET_SECURITY	0xa19ec7c6	condrv.sys
22	IRP_MJ_POWER	0x8192148e	ntoskrnl.exe
23	IRP_MJ_SYSTEM_CONTROL	0x8192148e	ntoskrnl.exe
24	IRP_MJ_DEVICE_CHANGE	0x8192148e	ntoskrnl.exe
25	IRP_MJ_QUERY_QUOTA	0x8192148e	ntoskrnl.exe
26	IRP_MJ_SET_QUOTA	0x8192148e	ntoskrnl.exe
27	IRP_MJ_PNP	0x8192148e	ntoskrnl.exe

```
New modules entries.
================================================================================================================================================
Offset(V)  Name                Base             Size File
---------- ------------------- ---------- ---------- ----
0x874ed4f0 condrv.sys          0xa19e4000     0xd000 \SystemRoot\System32\drivers\condrv.sys


New modscan entries.
================================================================================================================================================
Offset(P)          Name                Base             Size File
------------------ ------------------- ---------- ---------- ----
0x000000007c41b148 ??                  0xa10ae000    0x33000
0x000000007c577008                      0x80400000    0x25000
0x000000007caed4f0 condrv.sys          0xa19e4000     0xd000 \SystemRoot\System32\drivers\condrv.sys


New devicetree entries.
================================================================================================================================================
DRV 0x7c40e5a8 ??
DRV 0x7c413790
DRV 0x7c41b670 (08X
DRV 0x7c41f818 ??
DRV 0x7c578520 ?????????
---| DEV 0x80000070  UNKNOWN
------| ATT 0x85c48ee0  - \FileSystem\FltMgr FILE_DEVICE_MAILSLOT
------| ATT 0x86711a48  - \FileSystem\FltMgr FILE_DEVICE_NAMED_PIPE
---| DEV 0x87503680  FILE_DEVICE_DISK_FILE_SYSTEM
---| DEV 0x85c48ee0  FILE_DEVICE_MAILSLOT
---| DEV 0x86711a48  FILE_DEVICE_NAMED_PIPE
DRV 0x7e877d18 \Driver\condrv
---| DEV 0x85677bf8 ConDrv UNKNOWN
---| DEV 0x874a78e8  FILE_DEVICE_DISK_FILE_SYSTEM
------| ATT 0x87503680  - \FileSystem\FltMgr FILE_DEVICE_DISK_FILE_SYSTEM


New mutantscan entries.
================================================================================================================================================
.NET CLR Data_Perf_Library_Lock_PID_724
.NET CLR Networking 4.0.0.0_Perf_Library_Lock_PID_724
.NET CLR Networking_Perf_Library_Lock_PID_724
.NET Data Provider for Oracle_Perf_Library_Lock_PID_724
.NET Data Provider for SqlServer_Perf_Library_Lock_PID_724
.NET Memory Cache 4.0_Perf_Library_Lock_PID_724
.NETFramework_Perf_Library_Lock_PID_724
273...4:0
748:752
BITS_Perf_Library_Lock_PID_724
DBWinMutex
ESENT_Perf_Library_Lock_PID_724
LOADPERF_MUTEX
Lsa_Perf_Library_Lock_PID_724
MSDTC Bridge 3.0.0.0_Perf_Library_Lock_PID_724
MSDTC Bridge 4.0.0.0_Perf_Library_Lock_PID_724
MSDTC_Perf_Library_Lock_PID_724
MSSCNTRS_Perf_Library_Lock_PID_724
```

No notable changes to highlight from the following plugins.
==========
iehistory
privs
eventhooks
envars
shimcache
shellbags
consoles
hashdump
drivermodule
unloadedmodules
callbacks
threads
symlinkscan
ssdt

Plugins that were executed but are not included in the report above.
==========
filescan
handles
getsids
deskscan
dlllist
ldrmodules
atoms
svcscan
atomscan
idt
gdt
timers
gditimers

End of report.

This report provides an analysis of volatility, generated by VolDiff v2.1. The report includes information about the memory images, profile, and date and time of the analysis. It also presents various sections with new entries found during the analysis.

Here is a summary of the new entries found in each section:

New pslist entries:

svchost.exe with PID 2976, PPID 2968, 5 threads, and 0 handles.
svchost.exe with PID 3000, PPID 548, 3 threads, and 0 handles.
New psscan entries:

svchost.exe with PID 3000, PPID 548.
svchost.exe with PID 2976, PPID 2968.
New psxview entries:

svchost.exe with PID 904 and various attributes.
svchost.exe with PID 1204 and various attributes.
svchost.exe with PID 3000 and various attributes.
svchost.exe with PID 2976 and various attributes.
New netscan entries:

UDPv4 connection on 0.0.0.0:0.

TCPv4 connection on 172.16.240.128:49159 with an established state, owned by svchost.exe (PID 2976).

New malfind entries:

Process svchost.exe with PID 2976 at address 0xed0000, containing specific hexadecimal data.

Process svchost.exe with PID 2976 at address 0x12d0000, containing specific hexadecimal data.

New sessions entries:

Process svchost.exe with PID 3000, associated with session ID 1 and 7 processes.

Process svchost.exe with PID 2976, associated with session ID 1.

New messagehooks entries:

Entries with various offsets, sessions, desktops, threads, filters, flags, functions, and modules.

New cmdline entries:

svchost.exe with PID 2976 and command line "svchost.exe".

svchost.exe with PID 3000 and command line "C:\Windows\System32\svchost.exe -k WerSvcGroup".

New driverscan entries:

Various driver entries with offsets, pointers, handles, start addresses, sizes, service keys, and names.

New driverirp entries:

Information about the driver "condrv" with its start address, size, and associated IRP (I/O Request Packet) functions.

CONCLUSION :

By leveraging Volatility and VolDiff within an incident response process, organizations can gain several advantages. Here are a few key conclusions:

1. Memory Forensics Capabilities: Volatility's extensive set of plugins and its ability to analyze memory artifacts allow analysts to uncover hidden or malicious activity that might not be evident through traditional disk-based forensics. This provides a deeper understanding of the incident, including the execution of malware, presence of rootkits, or evidence of memory-based attacks.
2. Timeline Analysis: VolDiff complements Volatility by comparing memory images taken at different time points, highlighting differences and changes. This helps in tracking the evolution of an incident and understanding the persistence mechanisms employed by attackers. It can also assist in identifying stealthy or fileless malware that may not leave traditional artifacts on disk.
3. Malware Detection and Analysis: Volatility's memory analysis capabilities can aid in the identification and analysis of malware present in memory. This includes examining malicious processes, identifying injected code or hooks, and extracting malware samples for further analysis. VolDiff can assist in tracking changes made by malware across different memory images.
4. Enhanced Incident Response Capabilities: By incorporating Volatility and VolDiff into the incident response workflow, organizations can improve their ability to detect and respond to security incidents promptly. The ability to analyze live memory or memory images allows for a more comprehensive understanding of the incident, leading to faster containment, eradication, and recovery.

In conclusion, the integration of Volatility and VolDiff into the incident response process empowers organizations with advanced memory forensics capabilities. These tools provide deeper visibility into volatile data, facilitate the detection and analysis of malware, and contribute to a more effective and efficient incident response. By leveraging these tools, organizations can enhance their incident response capabilities and better protect their systems and data.