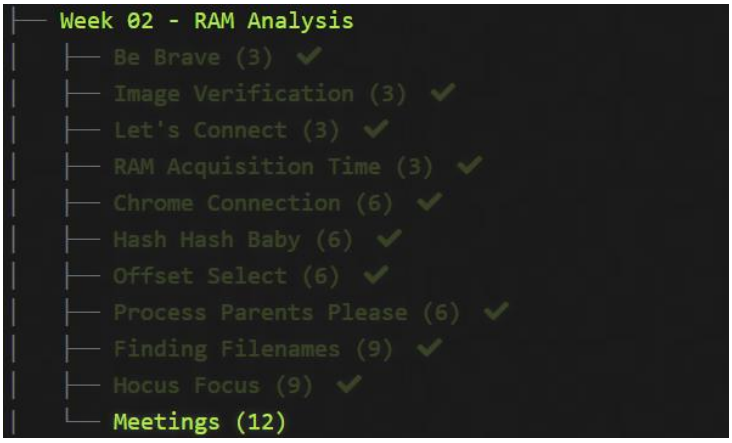
CYBERCRIME

Author: Kharim Mchatta

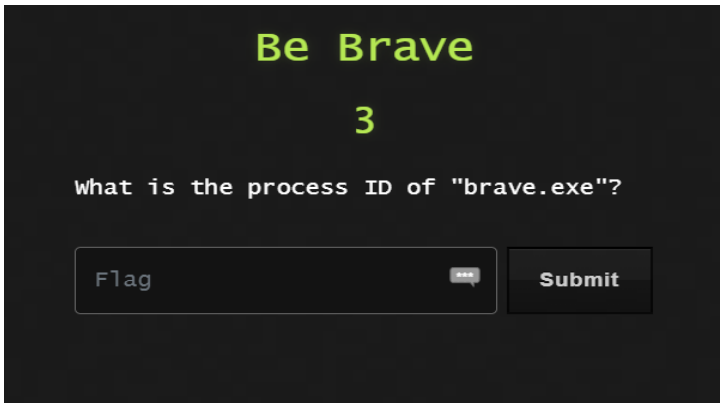
Title: Africa Digital Forensics CTF Competition

Date: 5/11/2021

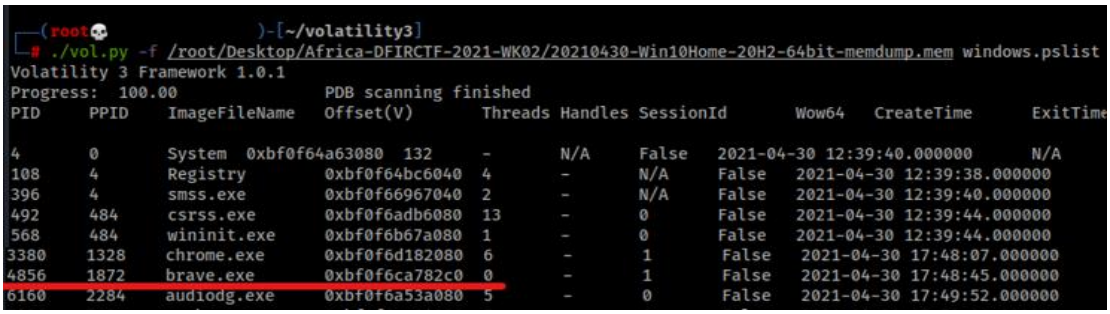
On 5/310/2021 the second week of the competition of the United Nation Office of Drugs and Cybercrime (UNODC) CTF had begun. The second week's challenge was based on memory analysis. This was a very interesting challenge to me due to the way the questions were set. Some questions were very trick but doable. In this week we were faced with 11 questions which were supposed to be answered based on the memory dump which was provided.



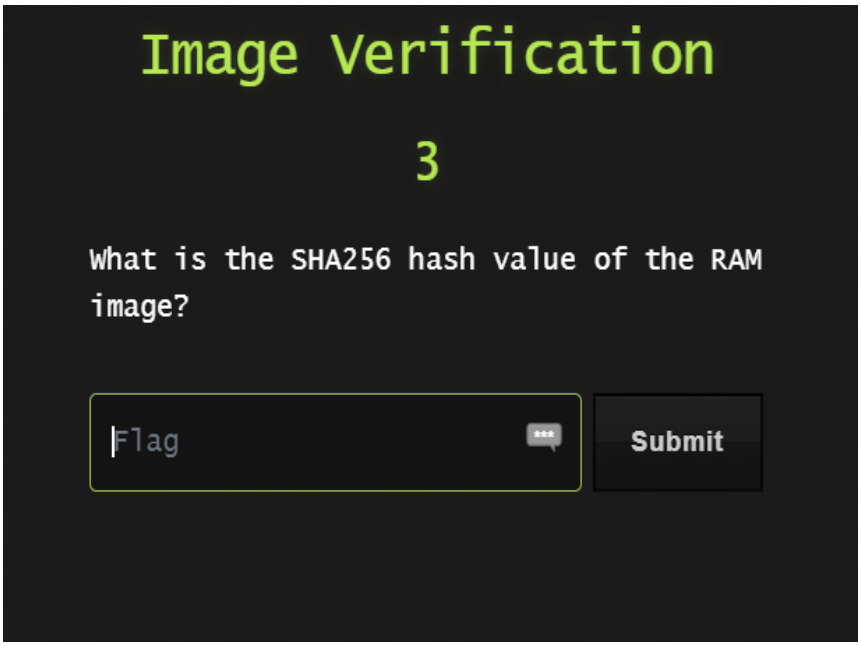
Starting off with the first question which was called Brave the question was to find the process ID of brave.exe. This challenge has 3 points to it, and this is how I solved it.



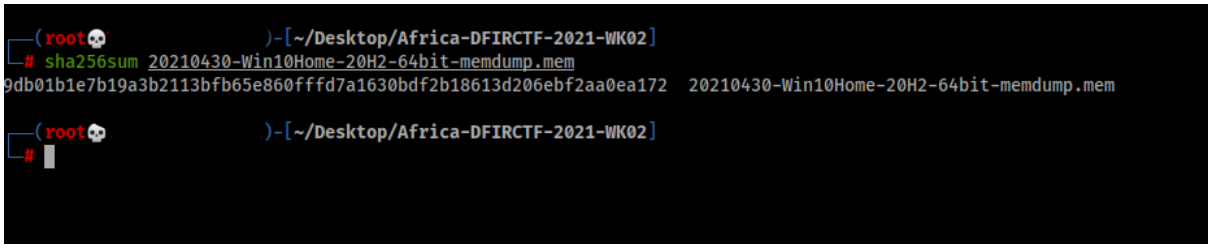
Solving this you needed a tool called volatility 3, if you used the previous version then you would face some few challenges in getting the profile of the image. In Volatility 3 I used the command **windows.pslist** to list all the running process with their id. After pressing enter I got the process id of brave. This was a straightforward challenge.



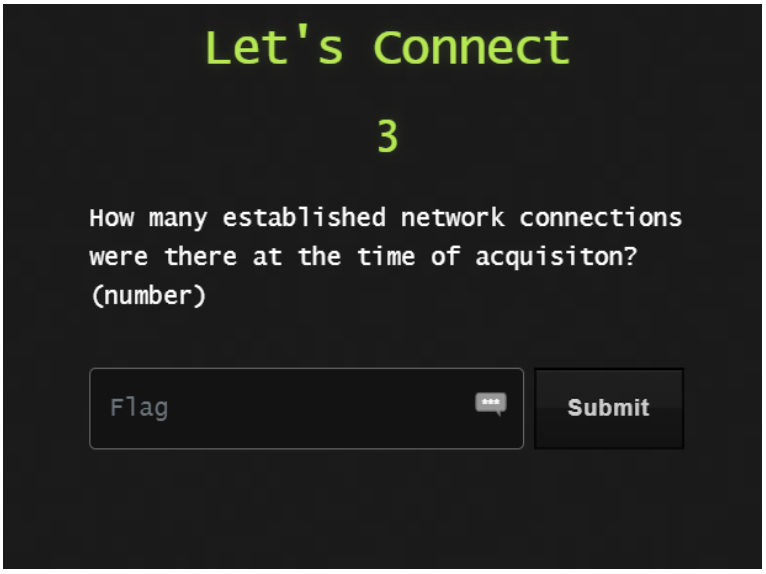
Heading off to the second question which was called Image verification the question was to find the SHA256 of the RAM. This challenge has 3 points to it, and this is how I solved it.



This was also a straightforward challenge which had no complications, we made use of a tool called sha256sum which helped us to get the hash of the memory dump, doing so we submitted the hash and got the answer.



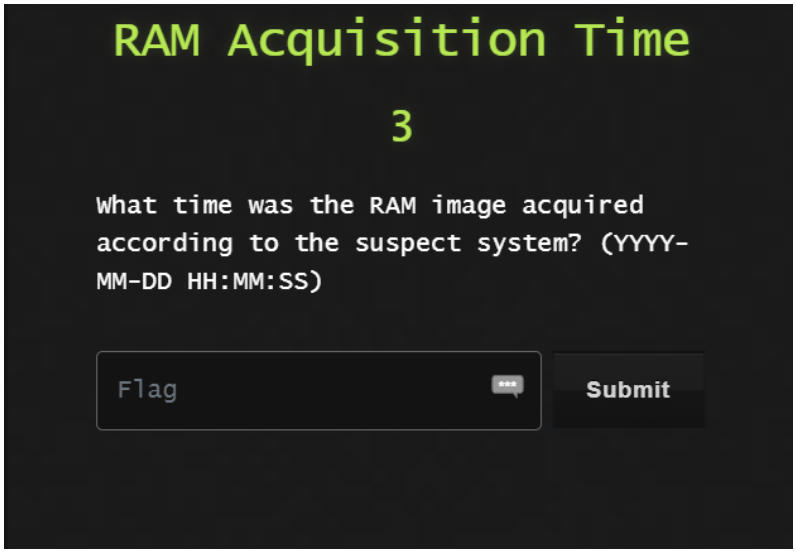
The third question was called Lets connect. the question was to how many established network connections there at the time of acquisition were. This challenge has 3 points to it, and this is how I solved it.



This was another very straight forward challenge. Here I had to make use of the command **windows.netscan** which allowed a user to view all the network activities that were running during the period the acquisition was taking place. To make my work simple I had to filter out unwanted information using **grep** from the results in order to get only the established connections. After running the command, we got our answer.

```
┌──(root💀                )-[~/volatility3]
└─# ./vol.py -t /root/Desktop/Africa-DFIRCTF-2021-WK02/20210430-Win10Home-20H2-64bit-memdump.mem windows.netscan | grep -vi closed | egrep 'ESTABLISHED'
0xbf0f6a53ca20.0TCPv4    10.0.2.15DB scan49833fin52.230.222.68    443     ESTABLISHED     2812    svchost.exe     2021-04-30 17:50:07.000000
0xbf0f6ad16050  TCPv4    10.0.2.15       49829   142.250.191.208 443     ESTABLISHED     5624    svchost.exe     2021-04-30 17:49:58.000000
0xbf0f6ad1fad0  TCPv4    10.0.2.15       49847   52.230.222.68   443     ESTABLISHED     2812    svchost.exe     2021-04-30 17:52:17.000000
0xbf0f6c6352b0  TCPv4    10.0.2.15       49842   52.113.196.254  443     ESTABLISHED     5104    SearchApp.exe   2021-04-30 17:51:25.000000
0xbf0f6c7104d0  TCPv4    10.0.2.15       49778   185.70.41.130   443     ESTABLISHED     1840    chrome.exe      2021-04-30 17:45:00.000000
0xbf0f6cd4fa20  TCPv4    10.0.2.15       49837   204.79.197.200  443     ESTABLISHED     5104    SearchApp.exe   2021-04-30 17:51:18.000000
0xbf0f6d0c64a0  TCPv4    10.0.2.15       49843   204.79.197.222  443     ESTABLISHED     5104    SearchApp.exe   2021-04-30 17:51:26.000000
0xbf0f6d51c4a0  TCPv4    10.0.2.15       49838   13.107.3.254    443     ESTABLISHED     5104    SearchApp.exe   2021-04-30 17:51:23.000000
0xbf0f6d525a20  TCPv4    10.0.2.15       49845   23.101.202.202  443     ESTABLISHED     1156    MsMpEng.exe     2021-04-30 17:51:36.000000
0xe80000193a20  TCPv4    10.0.2.15       49845   23.101.202.202  443     ESTABLISHED     1156    MsMpEng.exe     2021-04-30 17:51:36.000000

┌──(root💀                )-[~/volatility3]
└─#
```

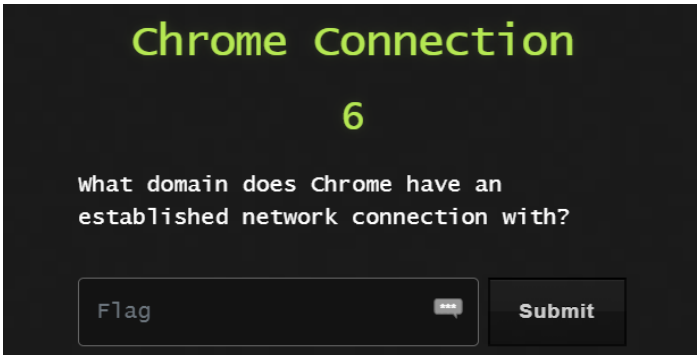
The fourth question was called Lets RAM Acquisition. the question was what time the RAM image was acquired according to the suspect system. This challenge has 3 points to it, and this is how I solved it.



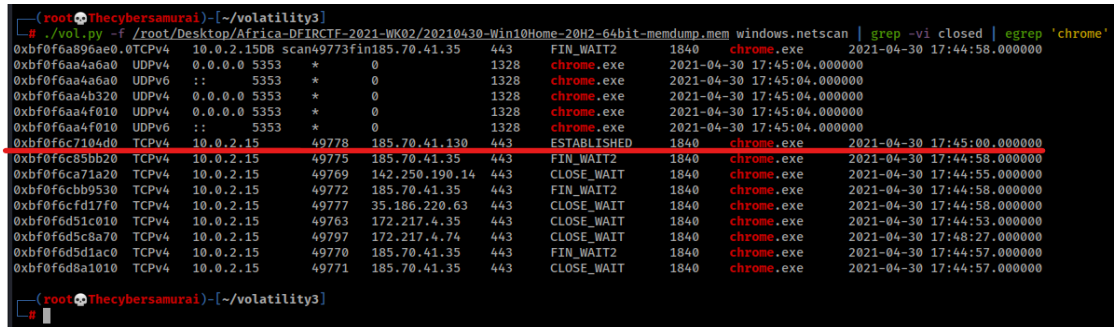
This was the simplest of them all. I had run the command **windows.info** which gave me information about the whole memory dump. After running the command, I had obtained the answer to the question.

```
┌──(root💀Thecybersamurai)-[~/volatility3]
└─# python3 vol.py -f /root/Desktop/Africa-DFIRCTF-2021-WK02/20210430-Win10Home-20H2-64bit-memdump.mem windows.info
Volatility 3 Framework 1.0.1
Progress:  100.00               PDB scanning finished
Variable        Value

Kernel Base     0xf8043cc00000
DTB     0x1aa000
Symbols file:///root/volatility3/volatility3/symbols/windows/ntkrnlmp.pdb/769C521E4833ECF72E21F02BF33691A5-1.json.xz
Is64Bit True
IsPAE   False
primary 0 WindowsIntel32e
memory_layer    1 FileLayer
KdVersionBlock  0xf8043d80f368
Major/Minor     15.19041
MachineType     34404
KeNumberProcessors      4
SystemTime      2021-04-30 17:52:19
```

The fifth question was called Lets Chrome Connection. the question was what domain chrome has an established network connection with. This challenge has 6 points to it, and this is how I solved it.

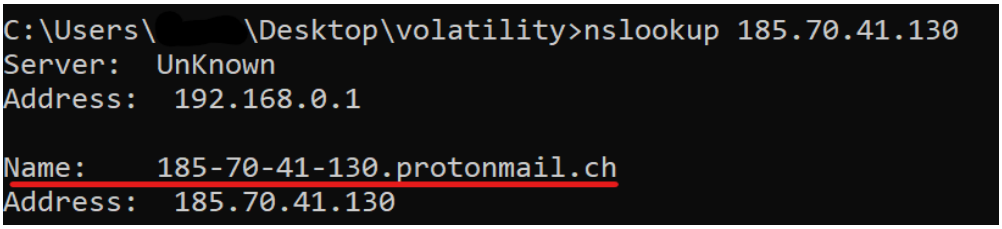


This was a very straight forward question but a tricky one on my side. I had run the command **windows.netscan** so as to see the network activities then I had filtered the network activities to only show chrome as results using **grep** command.

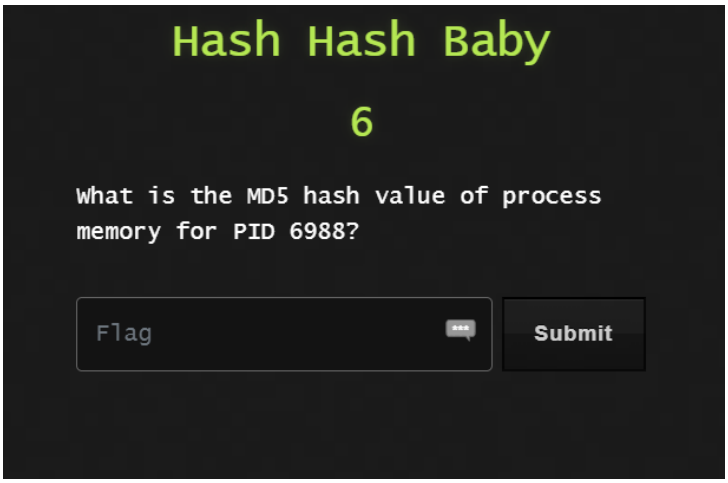


As you can see there is the established connection of chrome, but we are not finished, based on the question they wanted to know the domain name of the established connection. Here is where I had run into rabbit hole which I was not supposed to thanks to the power of overthinking, I had initially thought I should extract PCAP file from the memory dump using a tool called bulk_extractor64 which is a tool for windows. After extracting the PCAP file I had started analysing the file using wireshark which as I said initially was a rabbit hole and that was me overthinking things.

The correct way to have solved this challenge was to take the obtained IP address and perform a nslookup of the IP and after doing so I had got the domain name.



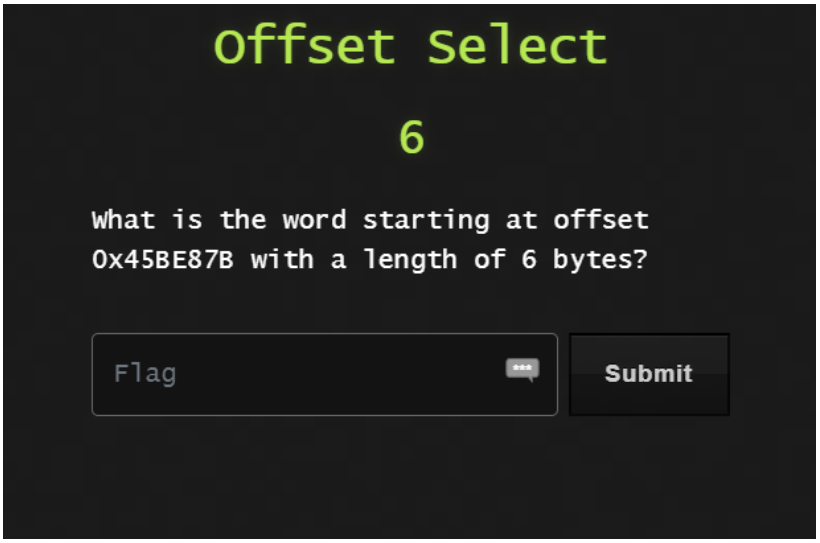
The sixth question was called hash hash baby. the question was what is the MD5 value of the process memory for PID 6988. This challenge has 6 points to it, and this is how I solved it.



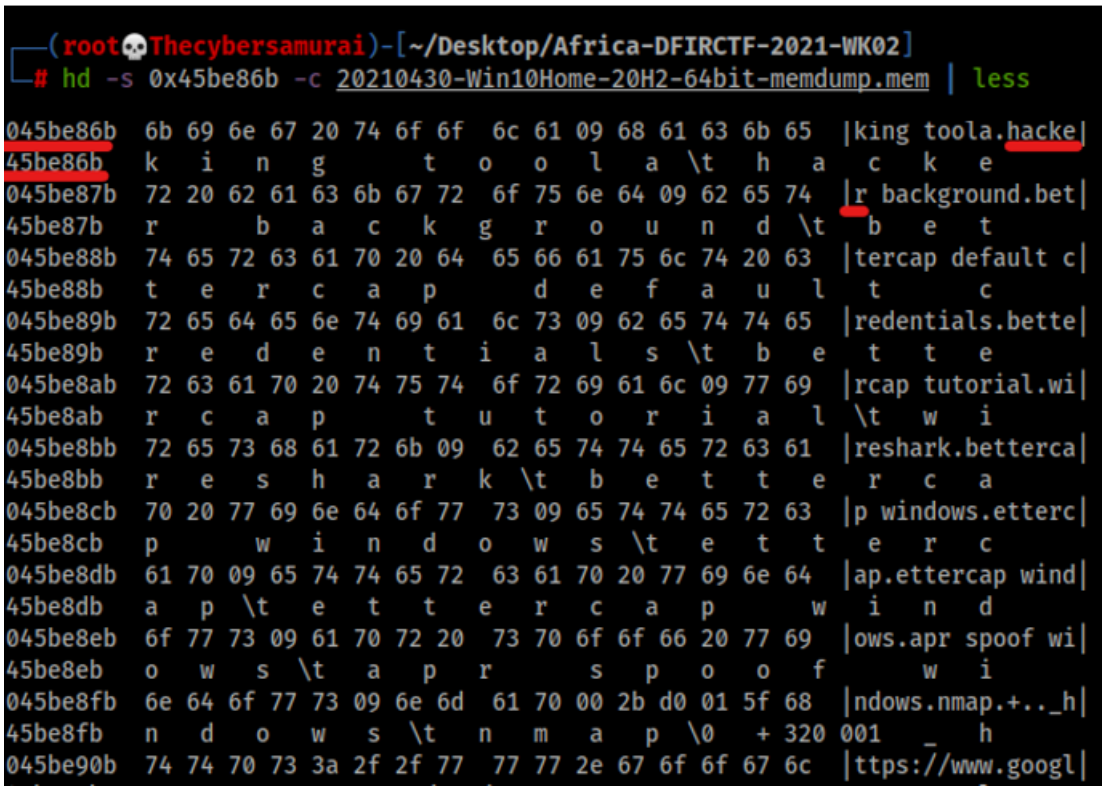
This was a very straight forward challenge which required me to dump the process first on to my local machine then be in a position to hash the process. For dumping a process in volatility, I used the command **windows.pslist** and then I filtered all the process to output only the process with the PID 6988, after that I used the command **- - pid** and **- -dump** which when run it extracted the process to my local machine, once that was complete, I hashed the process using the commands **md5sum** and got the answer.

```
┌──(root💀                )-[~/volatility3]
└─# ./vol.py -f /root/Desktop/Africa-DFIRCTF-2021-WK02/20210430-Win10Home-20H2-64bit-memdump.mem windows.pslist | grep 6988
6988    4352    OneDrive.exe    0xbf0f6d4262c0  26      -       1       True    2021-04-30 17:40:01.000000      N/A     Disabled

┌──(root💀                )-[~/volatility3]
└─# ./vol.py -f /root/Desktop/Africa-DFIRCTF-2021-WK02/20210430-Win10Home-20H2-64bit-memdump.mem windows.pslist --pid 6988 --dump
Volatility 3 Framework 1.0.1
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

6988    4352    OneDrive.exe    0xbf0f6d4262c0  26      -       1       True    2021-04-30 17:40:01.000000      N/A     pid.6988.0x1c0000.dmp

┌──(root💀                )-[~/volatility3]
└─# md5sum pid.6988.0x1c0000.dmp
0b493d8e26f03ccd2060e0be85f430af  pid.6988.0x1c0000.dmp

┌──(root💀                )-[~/volatility3]
└─#
```

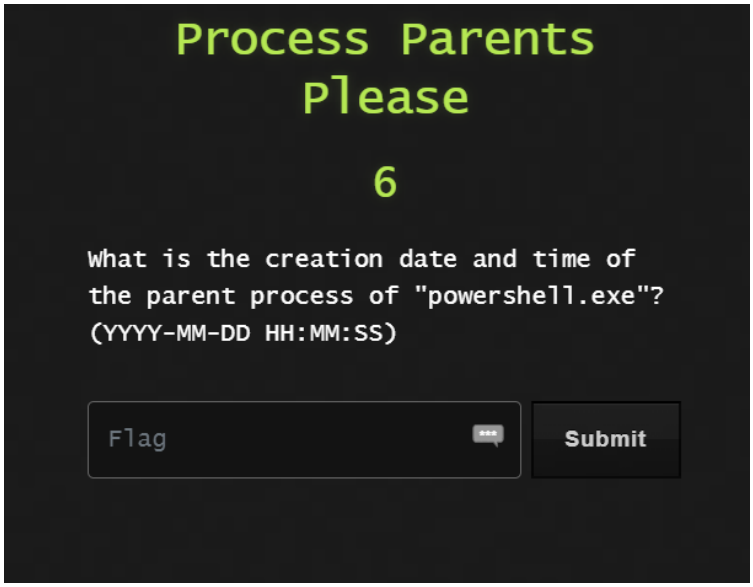
The seventh question was called offset select. the question was what the word is starting at offset 0x45BE87B with a length of 6 bytes. This challenge has 6 points to it, and this is how I solved it.



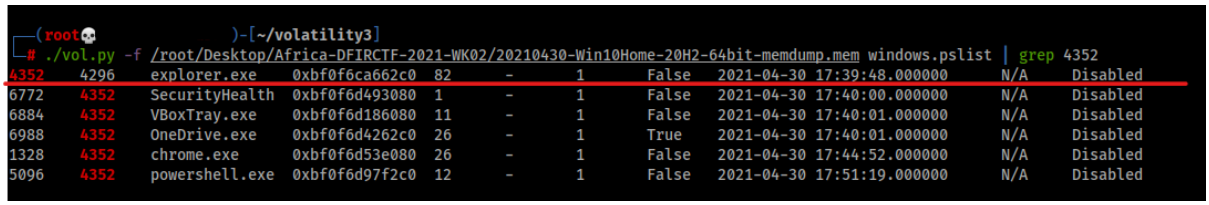
Here I used a program called **hd** which would display the hex value of a file. Another similar tool is called **hexdump** which does the same thing as **hd**. Running the command gave me the hex of the file analysing the result, I got the answer.



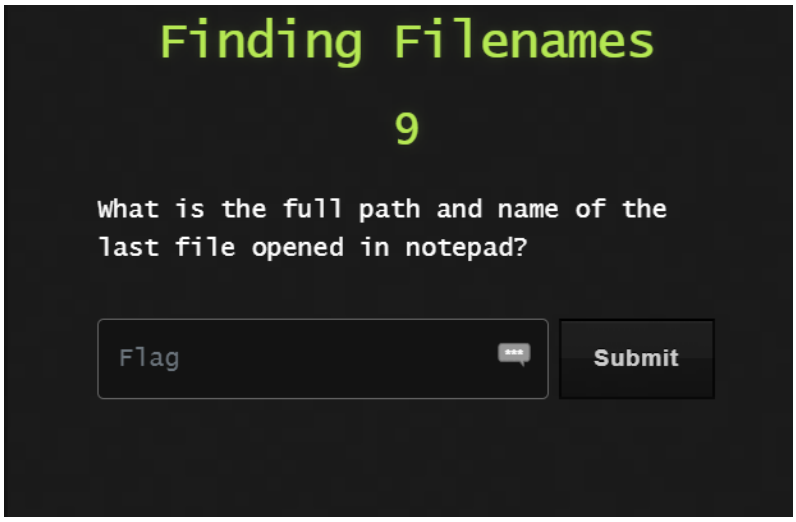
The eighth question was called process parent please. the question was what the creation date and time of the parent process of is 'powershell.exe'. This challenge has 6 points to it, and this is how I solved it.



PowerShell is a process that was running on the suspect device so logically we would use the command **windows.pslist** and filtered the parent process to match all the process. After analysing the result, I had found the parent process

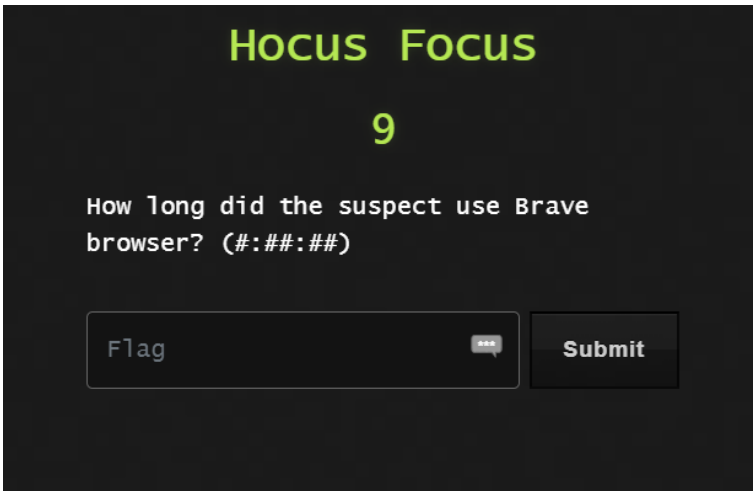


The nineth question was called Finding Filenames. the question was the full path and name of the file last opened in notepad. This challenge has 9 points to it, and this is how I solved it.

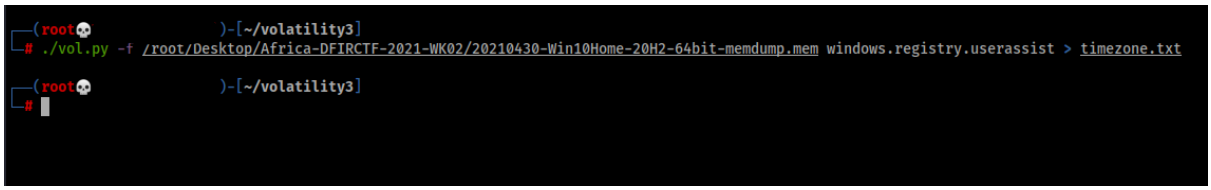


Here I had to make use of the command **windows.cmdline** this allows you to see what the last command was run from the terminal and what was the result. Running the command, I had gotten the last file that was run and there was the answer.

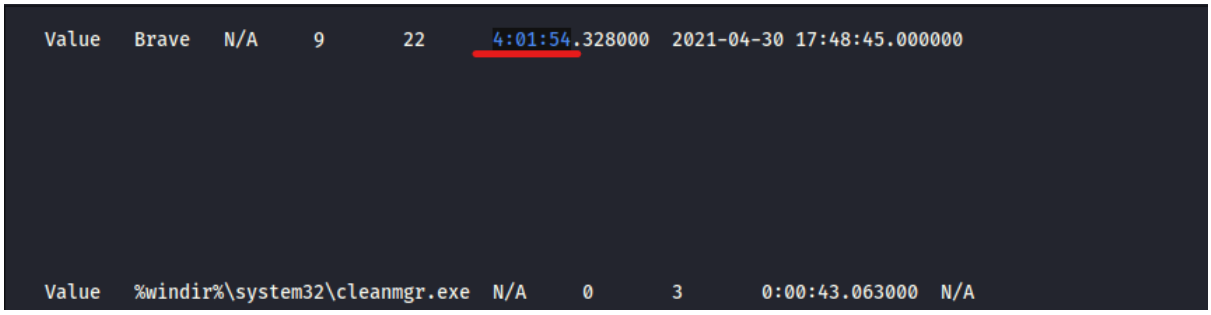
The tenth question was called Hocus Focus. The question was how long did the suspect use brave browser. This challenge has 9 points to it, and this is how I solved it.



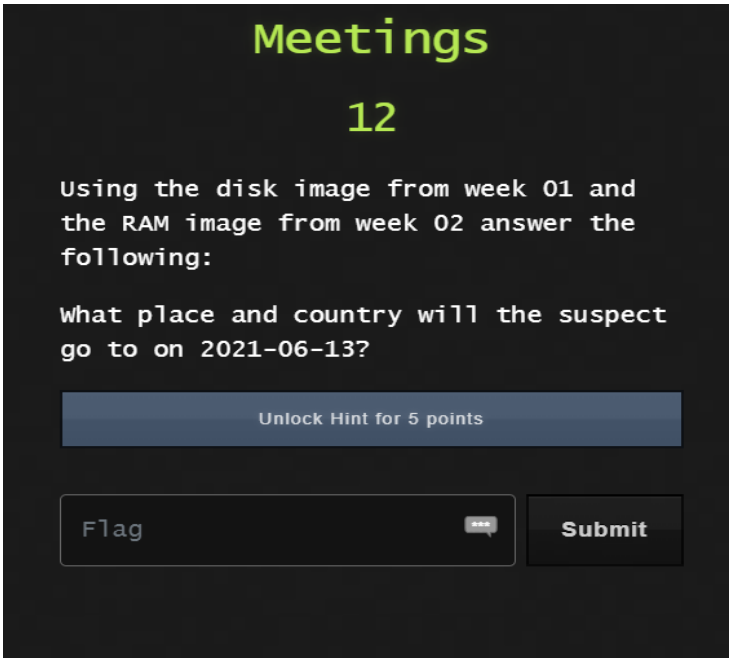
This was one of the toughest questions which really got me struggling. I used the command **windows.resgistry.userassist** command to get my answer. I had saved the result in a txt file simply because it was readable that way and I was able to see all the result.



Running through the txt file I had searched for the word brave and after going through the index word I came across the process which had the answer, there was no need for any calculation cause the answer was already there showing the time the browser had run.



The eleventh question was called Meetings. The question was using the disk image from week 1 and ram image from week2 what place and country the suspect will go to on 2021-06-13. This challenge has 12 points to it, and this is how I solved it.



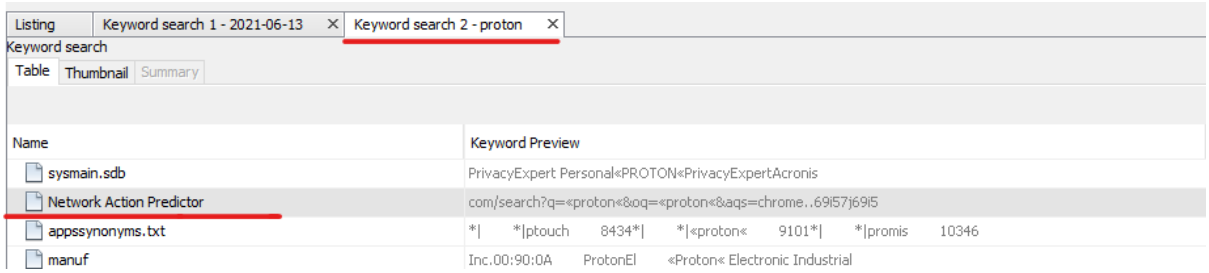
This was one of the most frustrating challenges I have ever met simply because of how it was setup in a way which required some proper critical thinking in order to solve it. The challenge was tricky in all sorts of ways which made me fall in plenty of rabbit holes before solving it.

The first tricky part was the aspect of the challenge that says you require both images from week 1 and 2 in order to solve the challenge which for some reason you actually did not. the second tricky part was where the challenge gave a date which was not located in none of the files, nor process in the images which made it more difficult to search for the date as we were used to in order to filter out results and narrow down your search scope.

The first rabbit hole I got into was trying to search for the string 2021-06-13 in both the images which for those who did the challenge know that did not bare any fruits. After hours of struggling, I was given a hint which said **from the suspect email**. Which this could mean anything to me anyway unless if I were overthinking the whole situation.

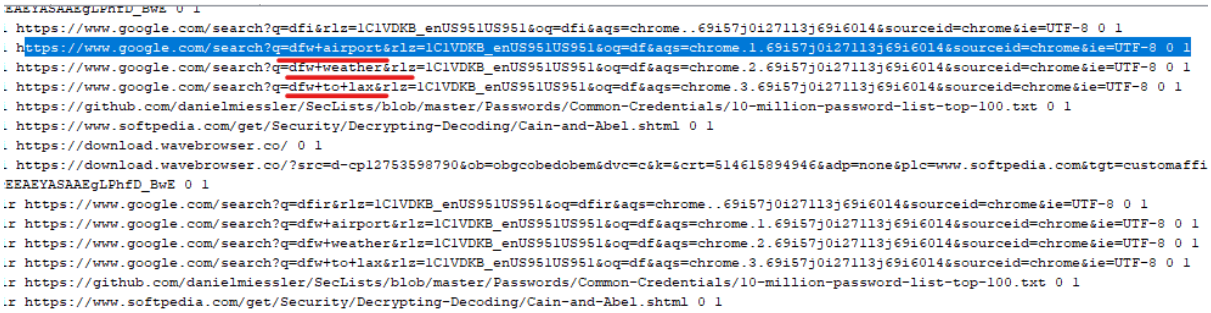
When I say overthink, this is how I was thinking email could be the email address or could be an email service provider.

I did a key search on the email service provider which was proton, and I got some hits on some files associated with the proton.



After going through the first file there was nothing interesting, going on to the second file there were various browser activity history information I had found and going through the information I had found that the user had search for the following things

1. Dallas international airport
2. Dallas to los Angeles
3. Weather in Dallas



Thinking that these were the answer. I went to insert the place and country I had found as the answer of my challenge and a pop up appeared which said incorrect

I was very baffled how those two were not the answer to the question. So, the hunt still continues

After digging and searching around I realized that the user john doe had downloaded a pdf file called almanac-start-a-garden.pdf file. So, I had to extract it and see the entails.

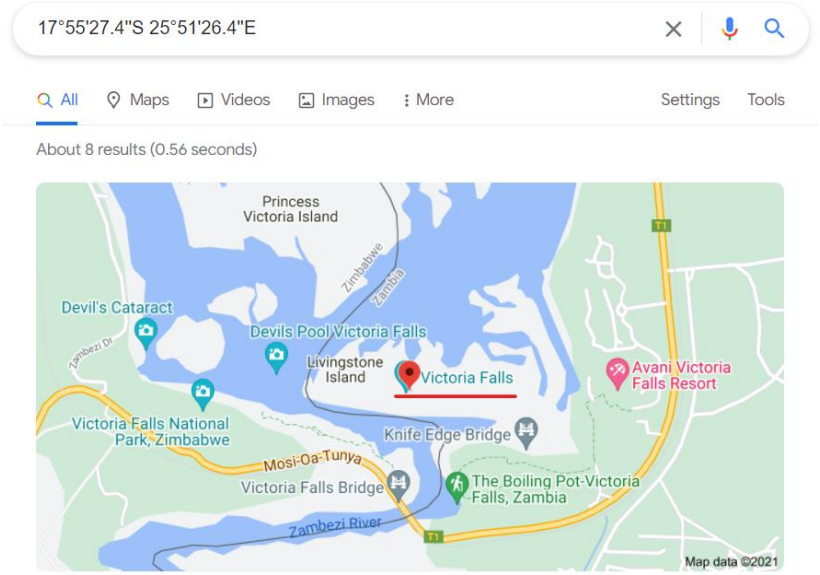
Name	S	O	Modified Time
payload.pdf		5	2021-04-28 20:23:12 EAT
payload.pdf		5	2021-04-28 20:23:12 EAT
payload.pdf		5	2021-04-28 20:23:12 EAT
payload.pdf		5	2021-04-28 20:23:12 EAT
payload.pdf		5	2021-04-28 20:23:12 EAT
payload.pdf		5	2021-04-28 20:23:12 EAT
payload.pdf		5	2021-04-28 20:23:12 EAT
almanac-start-a-garden.pdf		4	2021-04-30 04:05:53 EAT
almanac-start-a-garden.pdf:Zone.Identifier		4	2021-04-30 04:05:53 EAT

After downloading the file, it was a book about gardening, scrolling down through the pages I came across a page which has the date which was in one of the pages with coordinates.

Tip: The best times to spray are early morning and early evening, when the liquids will be absorbed most quickly and won't burn foliage. Choose a day when no rain is forecast and temperatures aren't extreme.

2021-06-13 17°55'27.4"S 25°51'26.4"E

Pasting the coordinates into google it pointed to the area which the suspect will go to.



Map for 17°55'27.4"S 25°51'26.4"E