

DeepDive Challenge

Scenario

You have given a memory image for a compromised machine. As a security blue team analyst Analyze the image and figure out attack details.

Task 1:

What profile should you use for this memory sample?

I used the imageinfo plugin

```
remnux@remnux:~/volatility$ python2 vol.py -f '/home/remnux/Challenge 2/banking-malware.vmem' imageinfo
Volatility Foundation Volatility Framework 2.6.1
INFO : volatility.debug : Determining profile based on KDBG search...
Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64 24000, Win2008R2SP1x64 23418, Win2008R2SP1x64, Win7SP1x64 24000, Win7SP1x64 23418
AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
AS Layer2 : FileAddressSpace (/home/remnux/Challenge 2/banking-malware.vmem)
PAE type : No PAE
DTB : 0x187800L
KDBG : 0xf80002bef120L
Number of Processors : 1
Image Type (Service Pack) : 1
KPCR for CPU 0 : 0xfffff80002bf1000L
KUSER_SHARED_DATA : 0xfffff80000000000L
Image date and time : 2021-02-09 00:51:25 UTC+0000
Image local date and time : 2021-02-08 22:51:25 -0200
```

Answer: Win7SP1x64_24000

Task 2:

What is the KDBG virtual address of the memory sample?

I used the kdbgscan plugin

```
Instantiating KDBG using: Kernel AS Win7SP1x64 24000 (6.1.7601 64bit)
Offset (V) : 0xf80002bef120
Offset (P) : 0x2bef120
KDBG owner tag check : True
Profile suggestion (KDBGHeader): Win7SP0x64
Version64 : 0xf80002bef0e8 (Major: 15, Minor: 7601)
Service Pack (CmNtCSDVersion) : 1
Build string (NtBuildLab) : 7601.24214.amd64fre.win7sp1_ldr_
PsActiveProcessHead : 0xfffff80002c28940 (54 processes)
PsLoadedModuleList : 0xfffff80002c46c90 (147 modules)
KernelBase : 0xfffff80002a0c000 (Matches MZ: True)
Major (OptionalHeader) : 6
Minor (OptionalHeader) : 1
KPCR : 0xfffff80002bf1000 (CPU 0)
```

Answer: 0xf80002bef120

Task 3:

There is a malicious process running, but it's hidden. What's its name?

I checked the -help with volatility 2 and saw the plugin psxview

psxview Find hidden processes with various process listings

```
remnux@remnux:~/volatility$ python2 vol.py -f '/home/remnux/Challenge 2/banking-malware.vmem' -profile=Win7SP1x64_24000 psxview
Volatility Foundation Volatility Framework 2.6.1
Offset(P) Name PID pslist pscan thrdproc pspcid csrss session deskthrd ExitTime
-----
0x000000007e51b000 services.exe 500 True True True True True True False
0x000000007d9b4000 svchost.exe 1772 True True True True True True True
0x000000007d25b000 vsvch.exe 2184 True True True True True True True
0x000000007d9ab000 lsass.exe 516 True True True True True True True
0x000000007d38c000 vmtoolsdService.exe 2416 True True True True True True True
0x000000007d97f940 svchost.exe 356 True True True True True True True
0x000000007d27710 svchost.exe 636 True True True True True True True
0x000000007f647b00 CompatTelRunne 996 True True True True True True True
0x000000007c12f100 WinPrvSE.exe 3488 True True True True True True True
0x000000007d8ab000 winlogon.exe 480 True True True True True True True
0x000000007d273500 dlh0st.exe 2844 True True True True True True True
0x000000007fc70b00 spssvc.exe 3616 True True True True True True True
0x000000007d8ce920 winAggrv.exe 3692 True True True True True True True
0x000000007d84c000 officeClickToR 1232 True True True True True True True
0x000000007d353b00 dm.exe 2236 True True True True True True True
0x000000007e780b00 SDHelper.exe 2828 True True True True True True True
0x000000007d393300 lsass.exe 500 True True True True True True False
0x000000007d273500 dlh0st.exe 3188 True True True True True True True
0x000000007d5d7b00 WinPrvSE.exe 1812 True True True True True True True
0x000000007e482b00 taskhost.exe 2344 True True True True True True True
0x000000007fc8ab00 vmtoolsd.exe 1484 True True True True True True True
0x000000007d24d2f8 SDHelper.exe 3200 True True True True True True True
0x000000007e5f45c0 svchost.exe 704 True True True True True True True
0x000000007d8ab000 VolAuthService.exe 1432 True True True True True True True
0x000000007d6db000 esdte.exe 1968 True True True True True True True
0x000000007d6cf000 svchost.exe 916 True True True True True True True
0x000000007d365500 vds.ps.exe 2448 False False True True True True True
0x000000007d259d00 conhost.exe 3028 True True True True True True True
0x000000007d7f7d00 SearchIndexer.exe 2616 True True True True True True True
0x000000007d78b000 wmpnetwk.exe 856 True True True True True True True
0x000000007e306d00 wininit.exe 412 True True True True True True True
0x000000007d33e000 taskhost.exe 2192 True True True True True True True
0x000000007eae4000 CompatTelRunne 2984 True True True True True True True
0x000000007e1bf100 svchost.exe 3324 True True True True True True False
0x000000007d6db000 svchost.exe 1288 True True True True True True True
0x000000007d902600 svchost.exe 868 True True True True True True True
0x000000007d9ff618 svchost.exe 1132 True True True True True True True
0x000000007d3fcae0 vmtoolsd.exe 2424 True True True True True True True
0x000000007d35db00 explorer.exe 2280 True True True True True True True
0x000000007d203930 dlh0st.exe 1832 True True True True True True True
0x000000007d929370 audiodg.exe 1600 True True True True True True True
0x000000007d05db00 svchost.exe 3684 True True True True True True True
0x000000007e462b00 CompatTelRunne 2688 True True True True True True True
0x000000007d335500 taskeng.exe 2244 True True True True True True True
0x000000007d8a7b00 svchost.exe 756 True True True True True True True
0x000000007d24da38 SDHelper.exe 3196 True True True True True True True
0x000000007d9b9b00 spoolsv.exe 1696 True True True True True True True
0x000000007d6c4000 svchost.exe 960 True True True True True True True
0x000000007fd4ae00 ipconfig.exe 4008 True True False True True False 2021-02-09 00:51:25 UTC+0000
0x000000007d6e39a0 csrss.exe 360 True True True False True True 2021-02-09 00:51:25 UTC+0000
0x000000007fc46b00 cmd.exe 3732 True True False True True False 2021-02-09 00:51:25 UTC+0000
0x000000007e7d6c00 snus.exe 272 True True True False False False 2021-02-09 00:51:25 UTC+0000
0x000000007e3c0b00 csrss.exe 424 True True True False True True 2021-02-09 00:51:25 UTC+0000
0x000000007ffad800 System 4 True True True False False False 2021-02-09 00:51:25 UTC+0000
0x000000007fc32800 conhost.exe 3848 True True False True False True False 2021-02-09 00:51:25 UTC+0000
remnux@remnux:~/volatility$
```

Answer: vds.ps.exe

Task 4:

What is the physical offset of the malicious process?

0x000000007d336950	vds_ps.exe	2448	False	False	True	True	True	True	True
--------------------	------------	------	-------	-------	------	------	------	------	------

Task 5:
What is the full path (including executable name) of the hidden executable?

```
remnux@remnux:~/volatility$ python2 vls.py -f '/home/remnux/Challenge 2/banking-malware.vmem' --profile=Win7SP1x64_24000 filescan | grep -i vds_ps.exe
```

```
Volatility Foundation Volatility Framework 2.6.1
```

0x00000007d0035d0	2	R--r--d	\Device\HarddiskVolume1\Users\john\AppData\Local\api-ms-win-service-management-l2-1-0\	vds_ps.exe
0x00000007d0047e0	16	R--r--d	\Device\HarddiskVolume1\Users\john\AppData\Local\api-ms-win-service-management-l2-1-0\	vds_ps.exe
0x00000007dec4d20	16	R-----	\Device\HarddiskVolume1\Windows\Prefetch\	VDS_PS.EXE-69E862B4.pf

Task 6:
Which malware is this?

```
remnux@remnux:~/Volatility$ python2 vol.py -f ~/home/remnux/Challenge 2/banking-malware.vmem --profile=win7SP1x64_24000 dumpfiles -Q 0x000000007d0035d0 -D ~/home/remnux/Challenge 2/Malware/Volatility Foundation Volatility Framework 2.6.1
ImageSectionObject 0x7d0035d0 None \Device\HarddiskVolume1\Users\john\AppData\Local\api-ms-win-service-management-l2-1-0\vds.ps.exe
DataSectionObject 0x7d0035d0 None \Device\HarddiskVolume1\Users\john\AppData\Local\api-ms-win-service-management-l2-1-0\vds.ps.exe
```

Answer: emotet

Task 7:
The malicious process had two PEs injected into its memory. What's the size in bytes of the Vad that contains the largest injected PE? Answer in hex, like: 0xABC

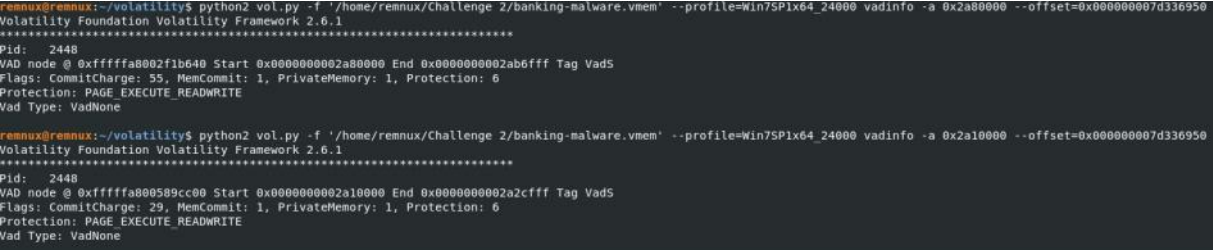
```
python2 vol.py -f '/home/remnux/Challenge 2/banking-malware.vmem' --profile=Win7SP1x64_24000
malfind --offset=0x000000007d336950
```

[illegible][illegible]

After that I asked the ChatGPT and he told me to use the plugin vadinfo with parameters of the address and offset so I used the command on both addresses.

```
python2 vol.py -f '/home/remnux/Challenge 2/banking-malware.vmem' --profile=Win7SP1x64_24000
vadinfo -a 0x2a80000 --offset=0x00000007d336950
```

```
python2 vol.py -f '/home/remnux/Challenge 2/banking-malware.vmem' --profile=Win7SP1x64_24000
vadinfo -a 0x2a10000 --offset=0x00000007d336950
```



Then I tried to calculate the Start and End with Chat but it didn't worked so I checked the write-up and they recommended on a HEX calculator website

<https://www.calculator.net/hex-calculator.html?number1=000000000002ab6fff&c2op=%2B&number2=000000000002a80000&calcType=op&x=Calculate>

Then I used the Start and End offsets and calculated it

Hex Calculator

Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

Result

Hex value:
000000000002ab6fff + 000000000002a80000 = 5536FFF

Decimal value:
44789759 + 44564480 = 89354239

00000000002

+

0000000002a

= ?

Calculate

Clear

At first I tried to entered the 5536FFF but it didn't worked so I checked the write-up again and the answer should be with 0x36fff

Answer: 0x36FFF

Task 8:
This process was unlinked from the ActiveProcessLinks list. Follow its forward link. Which process does it lead to? Answer with its name and extension

To be honest in this task I didn't understand shit, tried to read about the ActiveProcessLinks and even watched some of the write-ups but nothing really explained this task. (Looks like even the writeup owners didn't know it)

The only writeup which was good is [CyberDefenders - DeepDive](#)



Then I also asked chat but I still didn't understand but at the end he gives me the best explanation.

explain me like im dumb

Sure! Let's break it down into very simple terms:

What is `ActiveProcessLinks` ?

1. List of Processes:

- Imagine every program you open on your computer is like a person standing in a line. The `ActiveProcessLinks` is like a list that keeps track of everyone in that line (all the running programs).

2. Pointers:

- Each person in line has a little card that tells them who is in front of them and who is behind them. These cards are called `Flink` (next person) and `Blink` (previous person).

3. Why It's Important:

- This list helps the computer know what programs are running. If a program starts or stops, the computer adds or removes it from the line.
- If something bad happens (like a virus), it might try to hide itself by sneaking out of the line. Analyzing this list helps find those sneaky programs.

How It Works in Simple Steps

- When a program starts, it joins the line and is added to the `ActiveProcessLinks` list.
- When a program closes, it leaves the line and is removed from the list.
- You can follow the cards (pointers) to see `↓` is next to whom in the line.

Example

1. You open a game.

- The game gets added to the list of running programs.

2. The game closes.

- The game gets removed from the list.

3. A virus tries to hide.

- It removes its entry from the list to avoid being seen.

Summary

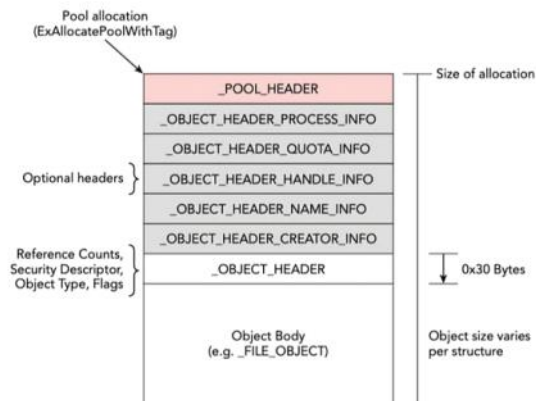
So, `ActiveProcessLinks` is like a line of people (programs) where each person knows who is next to them. It helps the computer keep track of what's running and can help find hidden bad stuff if you look closely!



In summary, the way to find the answer is to understand which PID are close to 2448 by the hidden malware

```
This picture is from pslist
remnux@remnux:~/volatility$ python2 vol.py -f ~/home/remnux/Challenge_2/banking-malware.vmem --profile=Win7SP1x64_24000 pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(s)      Name      PPID  Thds  Hnds  Sess  Wow64  Start              Exit
-----
0xffffffff8001804000 System      4      0    96    473  -----  0 2020-11-15 03:24:23 UTC+0000
0xffffffff800287d6c0 smss.exe    272     4     2     29  -----  0 2020-11-15 03:24:23 UTC+0000
0xffffffff8003e3e300 csrss.exe   268    352     9    740  -----  0 2020-11-15 03:24:27 UTC+0000
0xffffffff8002e986d0 wininit.exe 412    352     5     80  -----  0 2020-11-15 03:24:27 UTC+0000
0xffffffff8002fc0000 csrss.exe   424    404     9    294  -----  1 2020-11-15 03:24:27 UTC+0000
0xffffffff8003c0a000 winlogon.exe 480    404     5    117  -----  1 2020-11-15 03:24:27 UTC+0000
0xffffffff8003951b00 services.exe 500    412     3     294  -----  0 2020-11-15 03:24:27 UTC+0000
0xffffffff8003d38930 lsass.exe   500    412    16    860  -----  0 2020-11-15 03:24:27 UTC+0000
0xffffffff8003c4b000 lsm.exe    516    412    12    155  -----  0 2020-11-15 03:24:27 UTC+0000
0xffffffff8003e27710 svchost.exe 636    500    15    377  -----  0 2020-11-15 03:24:28 UTC+0000
0xffffffff80031f45c0 svchost.exe 704    500    12    327  -----  0 2020-11-15 03:24:28 UTC+0000
0xffffffff8003ea7b00 svchost.exe 756    500    32    592  -----  0 2020-11-15 03:24:28 UTC+0000
0xffffffff8003f02000 svchost.exe 868    500    25    420  -----  0 2020-11-15 03:24:28 UTC+0000
0xffffffff8003bcf000 svchost.exe 916    500    34    850  -----  0 2020-11-15 03:24:28 UTC+0000
0xffffffff8003c40000 svchost.exe 900    500    53    976  -----  0 2020-11-15 03:24:28 UTC+0000
0xffffffff8003f72970 audiodg.exe 1800   756     6    129  -----  0 2020-11-15 03:24:28 UTC+0000
0xffffffff8003f7f940 svchost.exe 356    500    28    539  -----  0 2020-11-15 03:24:28 UTC+0000
0xffffffff8003f89b00 spoolsv.exe 1096   500    18    282  -----  0 2020-11-15 03:24:28 UTC+0000
0xffffffff8003f7f610 svchost.exe 1132   500    23    331  -----  0 2020-11-15 03:24:28 UTC+0000
0xffffffff800408c000 OfficeClickToR 1232   500    25    579  -----  0 2020-11-15 03:24:28 UTC+0000
0xffffffff8004090b00 svchost.exe 1280   500    11    147  -----  0 2020-11-15 03:24:29 UTC+0000
0xffffffff800408ab00 VGAuthService 1432   500     4     87  -----  0 2020-11-15 03:24:29 UTC+0000
0xffffffff8003190f00 vmtoolsd.exe 1404   500    11    293  -----  0 2020-11-15 03:24:29 UTC+0000
0xffffffff800319ab00 svchost.exe 1772   500     6    186  -----  0 2020-11-15 03:24:29 UTC+0000
0xffffffff80043d7b00 WmiPrvSE.exe 1812   636    11    214  -----  0 2020-11-15 03:24:29 UTC+0000
0xffffffff8004403930 dlh0st.exe 1832   500    21    195  -----  0 2020-11-15 03:24:29 UTC+0000
0xffffffff8004473500 dlh0st.exe 2044   500    19    207  -----  0 2020-11-15 03:24:30 UTC+0000
0xffffffff80040db000 mdmfc.exe 1908   500    14    154  -----  0 2020-11-15 03:24:31 UTC+0000
0xffffffff80044c5000 VSSVC.exe 2104   500     6    119  -----  0 2020-11-15 03:24:31 UTC+0000
0xffffffff800453eb00 taskhost.exe 2192   500    12    210  -----  1 2020-11-15 03:24:33 UTC+0000
0xffffffff8004553b00 dm.exe     2236   868     6     75  -----  1 2020-11-15 03:24:33 UTC+0000
0xffffffff80045535b0 taskeng.exe 2244   960     7     80  -----  1 2020-11-15 03:24:33 UTC+0000
0xffffffff800455db00 explorer.exe 2260  2228    33    656  -----  1 2020-11-15 03:24:33 UTC+0000
0xffffffff800458cb00 vmtoolsd.exe 2416  2260     6     42  -----  1 2020-11-15 03:24:33 UTC+0000
0xffffffff80045fc3e0 vmtoolsd.exe 2424  2260    11    234  -----  1 2020-11-15 03:24:33 UTC+0000
0xffffffff800397db00 SearchIndexer.exe 2416   500    14    542  -----  0 2020-11-15 03:24:33 UTC+0000
0xffffffff8004078000 vmtoolsd.exe 856    500    17    453  -----  0 2020-11-15 03:24:40 UTC+0000
0xffffffff8003d26b00 svchost.exe 3084   500    31    316  -----  0 2020-11-15 03:24:40 UTC+0000
0xffffffff80037bf000 svchost.exe 3324   500    14    375  -----  0 2020-11-15 03:24:41 UTC+0000
0xffffffff8004021b00 WmiPrvSE.exe 3408   636    16    329  -----  0 2020-11-15 03:24:41 UTC+0000
0xffffffff8003ecce920 WmiPrvSE.exe 3692   500     8    119  -----  0 2020-11-15 03:24:50 UTC+0000
0xffffffff80044da300 SDRHelper.exe 3196   500    16    302  -----  1 2021-02-09 00:51:12 UTC+0000
0xffffffff800444d2f0 SDRHelper.exe 3200   500    16    302  -----  1 2021-02-09 00:51:12 UTC+0000
0xffffffff8002c40000 CompattelPumme 2904   500     3     52  -----  0 2021-02-09 00:51:13 UTC+0000
0xffffffff80045080c0 cmhost.exe 3028   360     2     33  -----  0 2021-02-09 00:51:13 UTC+0000
0xffffffff8003802b00 taskhost.exe 2344   500    10    167  -----  0 2021-02-09 00:51:13 UTC+0000
0xffffffff8003170000 SDRHelper.exe 2028   636    16    320  -----  1 2021-02-09 00:51:14 UTC+0000
0xffffffff8003d26b00 CompattelPumme 2608  2904    19    261  -----  0 2021-02-09 00:51:24 UTC+0000
0xffffffff800394a000 CompattelPumme 996  2608     6    141  -----  0 2021-02-09 00:51:26 UTC+0000
0xffffffff8001af0b00 spssvc.exe 3016   500     6    156  -----  0 2021-02-09 00:51:26 UTC+0000
0xffffffff80044f3580 dlh0st.exe 3188   636     7    138  -----  1 2021-02-09 00:51:25 UTC+0000
0xffffffff8001e4db00 cmd.exe     3732  1404     0  -----  0 2021-02-09 00:51:25 UTC+0000
0xffffffff8001e52000 cmhost.exe 3840   360     0  -----  0 2021-02-09 00:51:25 UTC+0000
0xffffffff8001b4a6e0 lpconfig.exe 4008  3732     0  -----  0 2021-02-09 00:51:25 UTC+0000
remnux@remnux:~/volatility$
```

This picture is from psxview



Next let us get the value of optional_header by subtracting 0x30 from the address

Command:

```
dt( "_POOL_HEADER", 0x000000007d336950-0x30,
space=addrspace().base)
```

```

[ 3 ]: dt( "_OBJECT_HEADER", 0x000000007d336950-0x30, space=addrspace().base)
[ OBJECT_HEADER OBJECT_HEADER ] @ 0x70336920
0x0 : PointerCount      149
0x8 : HandleCount       5
0x8 : NextToFree        5
0x10 : Lock             2100521264
0x10 : TypeIndex        7
0x19 : TraceFlags       0
0x1a : InfoMask         8
0x1b : Flags            0
0x20 : ObjectCreateInfo 18446738026461179264
0x20 : QuotaBlockCharged 18446738026461179264
0x28 : SecurityDescriptor 18446735964826813854
0x30 : Body             2100521296
```

InfoMask value corresponds to Optional Header value. Given it is 0x8, the optional header used is _OBJECT_HEADER_QUOTA_INFO and its size is 32 bytes based from the table lookup below (from the same link above);

Name	Structure	Bit Mask	Size (Bytes)	Description
Creator Info	_OBJECT_HEADER_CREATOR_INFO	0x1	32	Stores information on the creator of the object
Name Info	_OBJECT_HEADER_NAME_INFO	0x2	32	Stores the object's name
Handle Info	_OBJECT_HEADER_HANDLE_INFO	0x4	16	Maintains data about processes with open handles to the object
Quota Info	_OBJECT_HEADER_QUOTA_INFO	0x8	32	Tracks usage and resource stats
Process Info	_OBJECT_HEADER_PROCESS_INFO	0x10	16	Identifies the owning process

Let us combine all the data obtained above we can know that the initial pointer of pool_header is found by subtracting -0x60

Command:

```
vol.py -f banking-malware.vmem -- profile=Win7SP1x64_24000 volshell
dt( "_POOL_HEADER", 0x000000007d336950-0x60,
space=addrspace().base)
```

```

[ 5 ]: dt( "_POOL_HEADER", 0x000000007d336950-0x60, space=addrspace().base)
[ _POOL_HEADER _POOL_HEADER ] @ 0x703368F0
0x0 : BlockSize        86
0x0 : PoolIndex         0
0x0 : PoolType          2
0x0 : PreviousSize     10
0x0 : Ulong1            39190538
0x4 : PoolTag           1416573010
0x8 : AllocatorBackTraceIndex 0
0x8 : ProcessBilled     0
0xa : PoolTagHash       0
```

Convert the Pooltag value to hex which gives 546F3052. Then convert to ASCII but in reverse order as it is stored in memory.

52306f54 will be R0oT in ASCII when reversed

Format:

Answer: R0oT

#10 What is the physical address of the hidden executable's pooltag?
(HINT: use volshell)

We need to add 4 bytes to the pool_header physical address we found on question #9.

```

In [5]: dt( "POOL_HEADER", 0x000000007d3368f0-0x00, space=addrspace().base)
[ _POOL_HEADER _POOL_HEADER] @ 0x7D3368F0
0x0 : BlockSize      86
0x0 : PoolIndex     0
0x0 : PoolType      2
0x0 : PreviousSize  10
0x0 : ULong1        39190538
0x4 : PoolTag       1416573010
0x8 : AllocatorBackTraceIndex 0
0x8 : ProcessBilled 0
0xa : PoolTagHash   0

```

0x7D3368F0 + 0x4 = 0x7D3368F4

Answer: 0x7D3368F4

Here is the video as well please like and subscribe :)

[Cyberdefenders.org DeepDive CTF Walkthrough](https://www.cyberdefenders.org/DeepDive/CTF-Walkthrough)

