

Process analysis

1. There are a total of 2 infected processes. For each process:

a. Process 1:

- i. Give the name and PID. (0.5 mark)

Name: **windows-update**

PID: **3240**

```
sansforensics@siftworkstation: ~/Desktop/CEC24069_Assignment_2
$ vol.py pslist -p 3240,3496
Volatility Foundation Volatility Framework 2.6.1
Offset(V)           Name          PID  PPID  Thds   Hnds  Sess  Wow64 Start
-----
0xfffffa8001e512d0 windows-update    3240  2808   10    240   1    1 2022-09-01 09:18:48 UTC+0000
0xfffffa8001dc3060 rundll32.exe     3496  3468    6    204   0    1 2022-09-01 09:19:53 UTC+0000
sansforensics@siftworkstation: ~/Desktop/CEC24069_Assignment_2
$
```

- ii. Give the PID of the parent process. (0.5 mark)

PID of the parent process (PPID): **2808**

```
sansforensics@siftworkstation: ~/Desktop/CEC24069_Assignment_2
$ vol.py pslist -p 3240,3496
Volatility Foundation Volatility Framework 2.6.1
Offset(V)           Name          PID  PPID  Thds   Hnds  Sess  Wow64 Start
-----
0xfffffa8001e512d0 windows-update    3240  2808   10    240   1    1 2022-09-01 09:18:48 UTC+0000
0xfffffa8001dc3060 rundll32.exe     3496  3468    6    204   0    1 2022-09-01 09:19:53 UTC+0000
sansforensics@siftworkstation: ~/Desktop/CEC24069_Assignment_2
$
```

- iii. Give one reason why the process is suspicious. (1 mark)

For me, the single strongest reason was that this process had multiple hits when I ran yarascan using rules.yar (obtained from <https://github.com/Te-k/cobaltstrike>), which contained some rules for detecting cobalt strike.

However, I did not conclude this process to be suspicious solely based on that single reason. If I may add, here are some of the other reasons:

- Output of cmdline for this process and its parent; windows-update running from a temp folder seems suspicious, and its parent process seems to have originated from a file in the Downloads folder.

windows-update pid: 3240

Command line : "C:\Users\User\AppData\Local\Temp\radA64C2.tmp\windows-update.exe"

mshta.exe pid: 2808

Command line : "C:\Windows\SysWOW64\mshta.exe" "C:\Users\User\Downloads\windows-update.hta"

- Appeared in malfind output with MZ header indicating a PE file / executable, may be a possible process injection

```

Assignment2 - VMware Workstation
File Edit View VM Tabs Help | └─ Assignment1 └─ Assignment2 └─ FlareVM
Activities Terminal Sep 26 11:12
sansforensics@siftworkstation: ~/Desktop/CEC24069_Assignment_2
$ vol.py malfind -p 3240,3496
Volatility Foundation Volatility Framework 2.6.1
Process: windows-update Pid: 3240 Address: 0x2080000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 62, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x00000000002080000 4d 5a 52 45 e8 00 00 00 5b 89 df 55 89 e5 81 MZRE.....[...U...
0x00000000002080010 c3 45 7d 00 00 ff d3 68 f0 b5 a2 56 68 04 00 .E}....h..Vh...
0x00000000002080020 00 57 ff d0 00 00 00 00 00 00 00 00 00 00 00 00 .W.....[...].
0x00000000002080030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 e8 00 00 00 ..[...].
0x00000000002080000 4d DEC EBP
0x00000000002080001 5a POP EDX
0x00000000002080002 52 PUSH EDX
0x00000000002080003 45 INC EBP
0x00000000002080004 e800000009 CALL 0x2080009
0x00000000002080009 5b POP EBX
0x00000000002080000 89df MOV EDI, EBX
0x00000000002080004 55 PUSH EBP
0x0000000000208000d 89e5 MOV EBP, ESP
0x0000000000208000f 81c3457d0000 ADD EBX, 0x7d45
0x00000000002080015 ffd3 CALL EBX
0x00000000002080017 68f0b5a256 PUSH DWORD 0x56a2b5f0
0x0000000000208001c 6804000000 PUSH DWORD 0x4
0x00000000002080021 57 PUSH EDI
0x00000000002080022 ffd0 CALL EAX
0x00000000002080024 0000 ADD [EAX], AL
0x00000000002080026 0000 ADD [EAX], AL
0x00000000002080028 0000 ADD [EAX], AL
0x0000000000208002a 0000 ADD [EAX], AL
0x0000000000208002c 0000 ADD [EAX], AL
0x0000000000208002e 0000 ADD [EAX], AL
0x00000000002080030 0000 ADD [EAX], AL
0x00000000002080032 0000 ADD [EAX], AL
0x00000000002080034 0000 ADD [EAX], AL
0x00000000002080036 0000 ADD [EAX], AL

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
83°F Mostly cloudy 7:12 pm 26/9/2022 ENG US
```

b. Process 2:

- i. Give the name and PID. (0.5 mark)

Name: **rundll32.exe**

PID: **3496**

```

sansforensics@siftworkstation: ~/Desktop/CEC24069_Assignment_2
$ vol.py pslist -p 3240,3496
Volatility Foundation Volatility Framework 2.6.1
Offset(V) Name PID PPID Thds Hnds Sess Wow64 Start Exit
0xfffffffffa8001e512d0 windows-update 3240 2808 10 240 1 1 2022-09-01 09:18:48 UTC+0000
0xfffffffffa8001dc3060 rundll32.exe 3496 3468 6 204 0 1 2022-09-01 09:19:53 UTC+0000
sansforensics@siftworkstation: ~/Desktop/CEC24069_Assignment_2
$
```

- ii. Give the PID of the parent process. (0.5 mark)

PID of the parent process (PPID): **3468**

- iii. Give one reason why the process is suspicious. (1 mark)

For me, the single strongest reason was that this process had multiple hits when I ran yarascan using rules.yar (obtained from <https://github.com/Te-k/cobaltstrike>), which contained some rules for detecting cobalt strike.

- It also appeared in malfind output with MZ header indicating a PE file / executable, may be a possible process injection

Assignment2 - VMware Workstation

File Edit View VM Tabs Help || □ ☰ 🔍 🌐 📁 🗃 🗃 🗃

Activities Terminal Sep 26 11:20

sansforensics@siftworkstation: ~/Desktop/CECZ4069_Assignment_2

```
$ vol.py malfind -p 3496
Volatility Foundation Volatility Framework 2.6.1
Process: rundll32.exe Pid: 3496 Address: 0x1800000
Vad Tag: VadTags Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 62, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x0000000000180000 4d 5a 52 45 e8 00 00 00 00 5b 89 df 55 89 e5 81 MZRE....[.U...
0x0000000000180010 c3 45 7d 00 00 ff d3 68 b5 a2 56 68 04 00 00 .E)...h...Vh...
0x0000000000180020 00 57 ff d0 00 00 00 00 00 00 00 00 00 00 00 00 .W...
0x0000000000180030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....[.U...

0x0000000000180000 4d DEC EBP
0x0000000000180001 5a POP EDX
0x0000000000180002 5f PUSH EDX
0x0000000000180003 45 INC EBP
0x0000000000180004 e800000000 CALL 0x1800009
0x0000000000180005 5b POP EBX
0x0000000000180006 89df MOV EDI, EBX
0x000000000018000c 55 PUSH EBP
0x000000000018000d 89e5 MOV EBP, ESP
0x0000000000180007 81c3457d0000 ADD EBX, 0x7d45
0x0000000000180015 f703 CALL EBX
0x0000000000180017 6870b5a256 PUSH DWORD 0x56a2b5f0
0x000000000018001c 6804600000 PUSH DWORD 0x4
0x0000000000180021 57 PUSH EDI
0x0000000000180022 f748 CALL EAX
0x0000000000180024 0000 ADD [EAX], AL
0x0000000000180026 0000 ADD [EAX], AL
0x0000000000180029 0000 ADD [EAX], AL
0x000000000018002d 0000 ADD [EAX], AL
0x000000000018002e 0000 ADD [EAX], AL
0x0000000000180030 0000 ADD [EAX], AL
0x0000000000180032 0000 ADD [EAX], AL
0x0000000000180034 0000 ADD [EAX], AL
0x0000000000180036 0000 ADD [EAX], AL

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
```

2. These 2 suspicious processes have different start times. Dump the process executable of the infected process that started earlier. What is the MD5 hash of the dumped process? (1 mark)

```

sansforensics@siftworkstation: ~/Desktop/CEC24069_Assignment_2
$ vol.py pslist -p 3240,3496
Volatility Foundation Volatility Framework 2.6.1
Offset(V)      Name          PID  PPID  Thds  Hnds  Sess  Wow64 Start
-----
0xfffffa8001e512d0 windows-update    3240   2808   10    240     1   1 2022-09-01 09:18:48 UTC+0000
0xfffffa8001dc3060 rundll32.exe    3496   3468     6    204     0   1 2022-09-01 09:19:53 UTC+0000
sansforensics@siftworkstation: ~/Desktop/CEC24069_Assignment_2
$ 

```

As seen above, windows-update process with PID 3240 started earlier, so we can dump using procdump and get the file's MD5 hash which is **fd75bd3918ba2b957d2cdf8ccdb37645**

```

sansforensics@siftworkstation: ~/Desktop/CEC24069_Assignment_2
$ vol.py procdump -p 3240 -D ~/Desktop/CEC24069_Assignment_2
Volatility Foundation Volatility Framework 2.6.1
Process(V)      ImageBase      Name          Result
-----
0xfffffa8001e512d0 0x0000000000400000 windows-update      OK: executable.3240.exe
sansforensics@siftworkstation: ~/Desktop/CEC24069_Assignment_2
$ md5sum executable.3240.exe
fd75bd3918ba2b957d2cdf8ccdb37645  executable.3240.exe
sansforensics@siftworkstation: ~/Desktop/CEC24069_Assignment_2
$ 

```

Registry analysis:

3. The threat actor has created a persistence for the malware in the registry. Please give the full registry key, registry value, and last write time/last modified time (UTC+0000) of the registry key

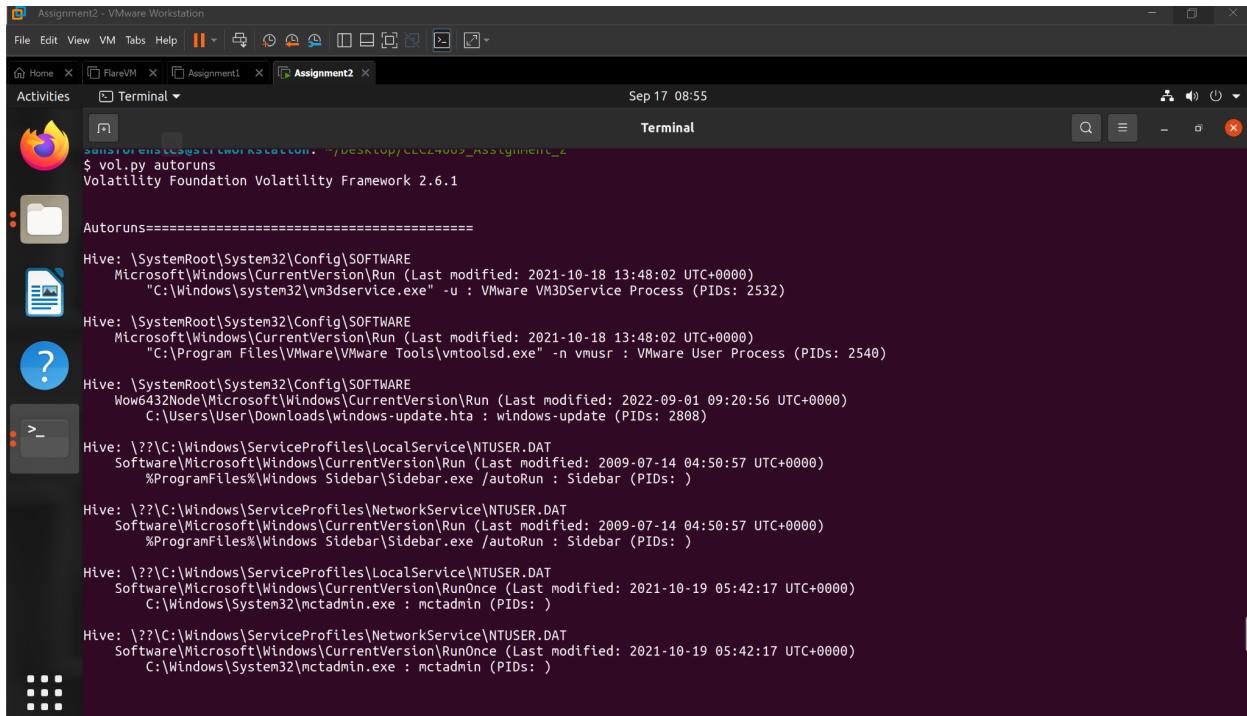
a. E.g.:

- i. Registry Key: SOFTWARE\Microsoft\Cryptography\MachineGuid
- ii. Registry Value: 2475afdc-1a0f-4383-ab3f-f9d24af02d4
- iii. Last Write Time: 2021-10-18 13:48:02 UTC+0000

(3 marks)

- i. Registry Key: **Wow6432Node\Microsoft\Windows\CurrentVersion\Run**
- ii. Registry Value:
windows-update : (S) **C:\Users\User\Downloads\windows-update.hta**
- iii. Last Write Time: **2022-09-01 09:20:56 UTC+0000**

The autoruns plugin was used, and the 3rd one from the top displayed the PID of 2808, which was the parent PID of 1 of the suspicious processes in question 1. Also, it was modified relatively recently.



```
sansforensics@sfifworkstation: ~/Desktop/CECZ4069_Assignment_2
$ vol.py autoruns
Volatility Foundation Volatility Framework 2.6.1

Autoruns=====

Hive: \SystemRoot\System32\Config\SOFTWARE
    Microsoft\Windows\CurrentVersion\Run (Last modified: 2021-10-18 13:48:02 UTC+0000)
        "C:\Windows\system32\m3dservice.exe" -u : VMware VM3DService Process (PIDs: 2532)

Hive: \SystemRoot\System32\Config\SOFTWARE
    Microsoft\Windows\CurrentVersion\Run (Last modified: 2021-10-18 13:48:02 UTC+0000)
        "C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmusr : VMware User Process (PIDs: 2540)

Hive: \SystemRoot\System32\Config\SOFTWARE
    Wow6432Node\Microsoft\Windows\CurrentVersion\Run (Last modified: 2022-09-01 09:20:56 UTC+0000)
        C:\Users\User\Downloads\windows-update.hta : windows-update (PIDs: 2808)

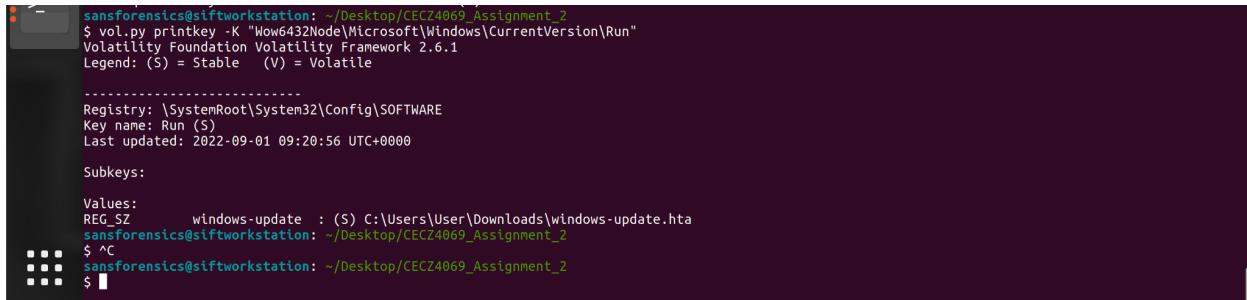
Hive: \?\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
    Software\Microsoft\Windows\CurrentVersion\Run (Last modified: 2009-07-14 04:50:57 UTC+0000)
        %ProgramFiles%\Windows Sidebar\Sidebar.exe /autoRun : Sidebar (PIDs: )

Hive: \?\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
    Software\Microsoft\Windows\CurrentVersion\Run (Last modified: 2009-07-14 04:50:57 UTC+0000)
        %ProgramFiles%\Windows Sidebar\Sidebar.exe /autoRun : Sidebar (PIDs: )

Hive: \?\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
    Software\Microsoft\Windows\CurrentVersion\RunOnce (Last modified: 2021-10-19 05:42:17 UTC+0000)
        C:\Windows\System32\mctadmin.exe : mctadmin (PIDs: )

Hive: \?\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
    Software\Microsoft\Windows\CurrentVersion\RunOnce (Last modified: 2021-10-19 05:42:17 UTC+0000)
        C:\Windows\System32\mctadmin.exe : mctadmin (PIDs: )
```

To confirm, we can print the registry key to see:



```
sansforensics@sfifworkstation: ~/Desktop/CECZ4069_Assignment_2
$ vol.py printkey -K "Wow6432Node\Microsoft\Windows\CurrentVersion\Run"
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable   (V) = Volatile

-----
Registry: \SystemRoot\System32\Config\SOFTWARE
Key name: Run (S)
Last updated: 2022-09-01 09:20:56 UTC+0000

Subkeys:

Values:
REG_SZ      windows-update : (S) C:\Users\User\Downloads\windows-update.hta
sansforensics@sfifworkstation: ~/Desktop/CECZ4069_Assignment_2
$ ^C
sansforensics@sfifworkstation: ~/Desktop/CECZ4069_Assignment_2
$
```

Network analysis:

4. What is the IP address and port of the malware C2 server? (1 mark)

IP address: **192.168.26.128**

Port: **80**

The image shows two side-by-side terminal windows within a VMware Workstation interface. Both terminals are running on a Linux host named 'sansforensics@siftworkstation'.

Terminal 1 (Top):

```

$ cd CobaltStrikeParser-master/
$ python3 parse_beacon_config.py ..\3496.dmp
BeaconType
Port
SleepTime
MaxGetSize
Jitter
MaxDNS
PublicKey_MDS
C2Server
UserAgent
HttpPostUrl
Malleable_C2_Instructions
HttpGet_Metadata
HttpPost_Metadata
HttpGet_Verb
HttpPost_Verb
HttpPostChunk
Spawnto_x86
Spawnto_x64
CrvoToScheme

```

Terminal 2 (Bottom):

```

$ cd CobaltStrikeParser-master/
$ python3 parse_beacon_config.py ..\3240.dmp
BeaconType
Port
SleepTime
MaxGetSize
Jitter
MaxDNS
PublicKey_MDS
C2Server
UserAgent
HttpPostUrl
Malleable_C2_Instructions
HttpGet_Metadata
HttpPost_Metadata
HttpGet_Verb
HttpPost_Verb
HttpPostChunk
Spawnto_x86
Spawnto_x64
CrvoToScheme

```

In both terminals, the command run is `python3 parse_beacon_config.py ..\[process_id].dmp`. The output shows various beacon configuration parameters such as BeaconType (HTTP), Port (80), SleepTime (60000), MaxGetSize (1048576), Jitter (0), MaxDNS (Not Found), PublicKey_MDS (defb5d95ce99e1ebbf421a1a38d9cb64), C2Server (192.168.26.128), UserAgent (Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; QQDownload 733; InfoPath.2)), HttpPostUrl (/submit.php), Malleable_C2_Instructions (Empty), HttpGet_Metadata (Metadata), HttpPost_Metadata (Content-Type: application/octet-stream, SessionId parameter "id"), HttpGet_Verb (GET, POST, 0), HttpPost_Verb (POST, 0), HttpPostChunk (%windir%\syswow64\rundll32.exe), Spawnto_x86 (Not Found), Spawnto_x64 (Not Found), and CrvoToScheme (0).

I extracted the memdump from both those processes identified in Q1 and ran a `parse_beacon_config.py` script on both to extract the cobalt strike configuration. (Source: <https://github.com/Sentinel-One/CobaltStrikeParser>) C2 server extracted was 192.168.26.128 and Beacon type is HTTP, which is over port 80.

To confirm, we can use netscan:

```

0x7fb0f03b0 UDPv4 0.0.0.0:59438 *:* 336 svchost.exe 2022-09-01 09:25:30 UTC+000
0x7fc4f7a0 UDPv4 0.0.0.0:60150 *:* 336 svchost.exe 2022-09-01 09:25:34 UTC+000
0x7fc5d9c0 UDPv4 0.0.0.0:52489 *:* 336 svchost.exe 2022-09-01 09:19:36 UTC+000
0x7fc5e230 UDPv4 0.0.0.0:57636 *:* 336 svchost.exe 2022-09-01 09:18:57 UTC+000
0x7fd372a0 UDPv4 0.0.0.0:63454 *:* 336 svchost.exe 2022-09-01 09:22:01 UTC+000
0x7fd4e2c0 UDPv4 0.0.0.0:56411 *:* 336 svchost.exe 2022-09-01 09:25:30 UTC+000
0x7fd71d70 UDPv4 0.0.0.0:61843 *:* 336 svchost.exe 2022-09-01 09:19:09 UTC+000
0x7fdb650 UDPv6 fe80::3056:129f:b222:12cb:546 *:* 740 svchost.exe 2022-09-01 09:17:04 UTC+000
0x7fa28520 TCPv4 0.0.0.0:445 255.255.255.255:49162 CLOSED 4 System
0x7fa84cf0 TCPv4 0.0.0.0:49163 255.255.255.255:445 CLOSED 4 System
0x7facf3b0 TCPv4 -:49176 192.168.26.128:80 CLOSED 3496 rundll32.exe
0x7facfcf0 TCPv4 0.0.0.0:49162 255.255.255.255:445 CLOSED 4 System
sansforensics@siftworkstation: ~/Desktop/CECZ4069_Assignment_2
$ vol.py netscan -p 3240,3496
Volatility Foundation Volatility Framework 2.6.1
Usage: Volatility - A memory forensics analysis platform.

vol.py: error: no such option: -p
sansforensics@siftworkstation: ~/Desktop/CECZ4069_Assignment_2
$ vol.py netscan | grep 3240
Volatility Foundation Volatility Framework 2.6.1
sansforensics@siftworkstation: ~/Desktop/CECZ4069_Assignment_2
$ vol.py netscan | grep 3496
Volatility Foundation Volatility Framework 2.6.1
0x7facf3b0 TCPv4 -:49176 192.168.26.128:80 CLOSED 3496 rundll32.exe
sansforensics@siftworkstation: ~/Desktop/CECZ4069_Assignment_2
$ 

```

Strings analysis:

5. Find the hidden flag in the memory image. The format of the flag is CECZ4069{<FLAG>}. Provide your answer in the same format. E.g. 'CECZ4069{I_<3_malware}' (1 mark)

```

CECZ4069_Assignment_2.vmem
This file has been changed on disk. You may choose to ignore the changes but reloading is the only safe option.
File Edit View Search Tools Help
File: /home/sansforensics/Desktop/CECZ4069_Assignment_2/CECZ4069_Assignment_2.vmem
Search for: CECZ4069
Search 8 bit: 67
Signed 8 bit: 67
Unsigned 8 bit: 67
Signed 16 bit: 17221
Unsigned 16 bit: 17221
Show little endian decoding
Signed 32 bit: 1128612698
Unsigned 32 bit: 1128612698
Float 32 bit: 197.2631
Float 64 bit: 1.19700584252325E+16
Show unsigned as hexadecimal
as Text Find Next Find Previous
Hexadecimal: 43 45 43 5A
Decimal: 067 069 067 090
Octal: 103 105 103 132
Binary: 01000011 01000101 01000011 01011010
ASCII Text: CECZ
Offset: 0x183a5d... Selection: 0x183a...

```

Using a hex editor, I used its in-built function to search for 'CECZ4069{' and got the flag:

CECZ4069{Memory_Forensics_Is_Fun}

6. Provide the process name and PID of the userland process that contains the flag (in question 5) in their memory space. (2 marks)

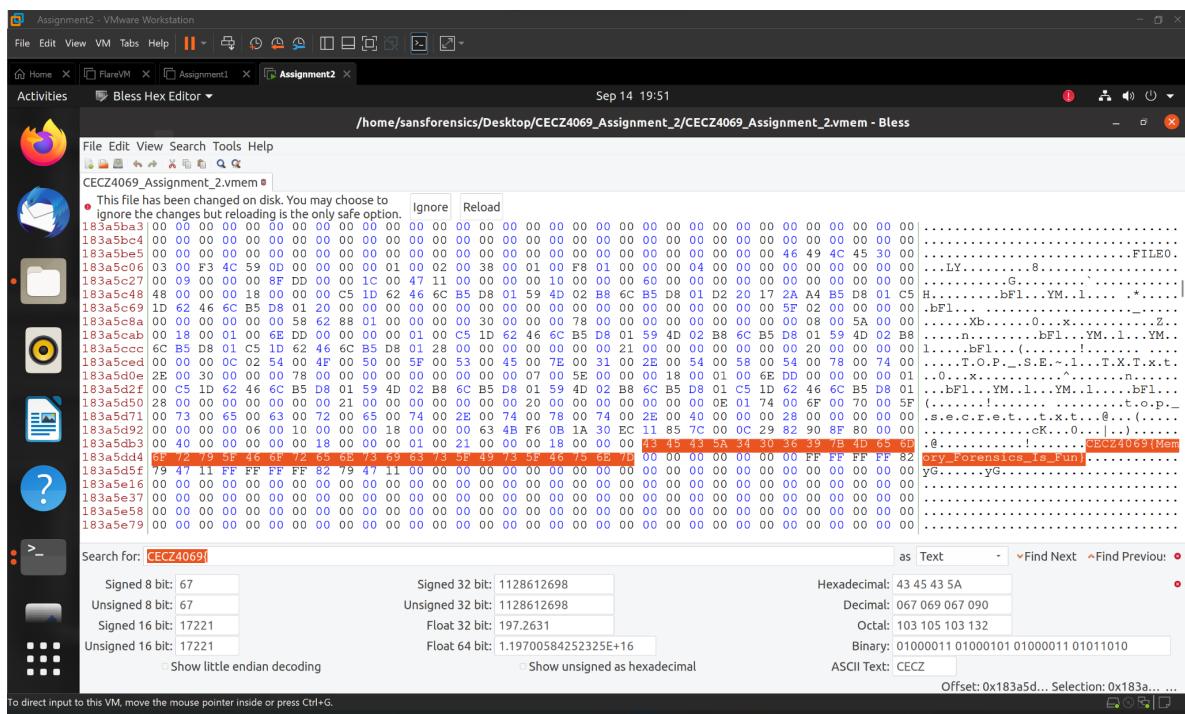
process name: **notepad.exe**

PID: **1212**

When I first ran vol.py cmdline, I spotted this command line argument which seemed a little out of place.

```
sansforensics@siftworkstation: ~/Desktop/CECZ4069_Assignment_2
$ vol.py cmdline -p 1212
Volatility Foundation Volatility Framework 2.6.1
*****
notepad.exe pid: 1212
Command line : "C:\Windows\system32\NOTEPAD.EXE" C:\Users\User\Documents\Work\top_secret.txt
sansforensics@siftworkstation: ~/Desktop/CECZ4069_Assignment_2
$
```

Also, in the hex editor, the flag was in close proximity to top_secret.txt seen above. Hence I thought the flag might be the content of that txt file, which could be related to the notepad.exe process seen earlier.



To prove that the flag was in the memory space of notepad.exe (PID 1212), I dumped the memory for the process before using grep to search for the flag string within the extracted strings from memdump. Sure enough, it was there.

```

sansforensics@siftworkstation: ~/Desktop/CECZ4069_Assignment_2
$ strings 1212.dmp -n 10 > strings_dmp.txt
sansforensics@siftworkstation: ~/Desktop/CECZ4069_Assignment_2
$ grep 'CECZ4069{Memory_Forensics_Is_Fun}' strings_dmp.txt
CECZ4069{Memory_Forensics_Is_Fun}

```

Threat analysis:

7. What version of cobalt strike was used to compromise this system? (Hint: volatility may not be able to help you here, use an external tool) (3 marks)

Version 4.4

Explanation:

Thanks to this guide at

<https://www.elastic.co/security-labs/extracting-cobalt-strike-beacon-configurations>, was able to correctly determine the version. Following the steps, I first used csce (<https://github.com/strozfriedberg/cobaltstrike-config-extractor>) to extract config info.

```

sansforensics@siftworkstation: ~/Desktop/CECZ4069_Assignment_2
$ csce -pretty 3240.dmp -v 4 >| csce_3240.txt
Could not parse source as PE file (DOS Header magic not found.)
sansforensics@siftworkstation: ~/Desktop/CECZ4069_Assignment_2
$ csce -pretty 3496.dmp -v 4 >| csce_3496.txt
Could not parse source as PE file (DOS Header magic not found.)

```

In the output, specifically the process-inject.stub field was of use. The value was "liuPJ9vfuo3dVZ7son6mSA==", the same for both process dumps.

```

Assignment2 - VMware Workstation
File Edit View VM Tabs Help || Activities Terminal
Activities Terminal Sep 17 16:44
sansforensics@siftworkstation: ~/Desktop/CECZ4069_Assignment_2
$ sed -n 5.69p csce_3240.txt
"process-inject": {
  "allocator": "VirtualAllocEx",
  "execute": [
    "CreateThread",
    "SetThreadContext",
    "CreateRemoteThread",
    "RtlCreateUserThread"
  ],
  "min_alloc": 0,
  "startrwx": true,
  "stub": "liuPJ9vfuo3dVZ7son6mSA==",
  "transform-x86": null,
  "transform-x64": null,
  "userwx": true
},
sansforensics@siftworkstation: ~/Desktop/CECZ4069_Assignment_2
$ sed -n 5.69p csce_3496.txt
"process-inject": {
  "allocator": "VirtualAllocEx",
  "execute": [
    "CreateThread",
    "SetThreadContext",
    "CreateRemoteThread",
    "RtlCreateUserThread"
  ],
  "min_alloc": 0,
  "startrwx": true,
  "stub": "liuPJ9vfuo3dVZ7son6mSA==",
  "transform-x86": null,
  "transform-x64": null,
  "userwx": true
},
sansforensics@siftworkstation: ~/Desktop/CECZ4069_Assignment_2

```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

82°F Mostly cloudy

ENG US 12:44 am 18/9/2022

Using that value, it was decoded from base64 and then converted to hex to get the MD5 file hash of the Cobalt Strike Java archive:

Inputting that MD5 hash into VirusTotal, we get its SHA256 hash (<https://www.virustotal.com/gui/file/7af9c759ac78da920395debb443b9007fdf51fa66a48f0fbdaafb30b00a8a858>). Then going to <https://verify.cobaltstrike.com/>, we can find the hash here:

Cobalt Strike 4.4 (August 04, 2021)
 7af9c759ac78da920395debb443b9007fdf51fa66a48f0fbdaafb30b00a8a858 Cobalt Strike 4.4
 Licensed (cobaltstrike.jar)

```
# Cobalt Strike 4.6.1 (May 23, 2022)
09e30bde7602cfa3358c0b1c9124079c77181c81c4ef0ef4f6789e24a3f07d5b Cobalt Strike 4.6.1 Licensed
(cobaltstrike.jar)

# Cobalt Strike 4.6 (April 12, 2022)
939aa731685ac5c2632e4790daf034110ae4aa7237a6db72c7bba219bd450727 Cobalt Strike 4.6.0 Licensed
(cobaltstrike.jar)

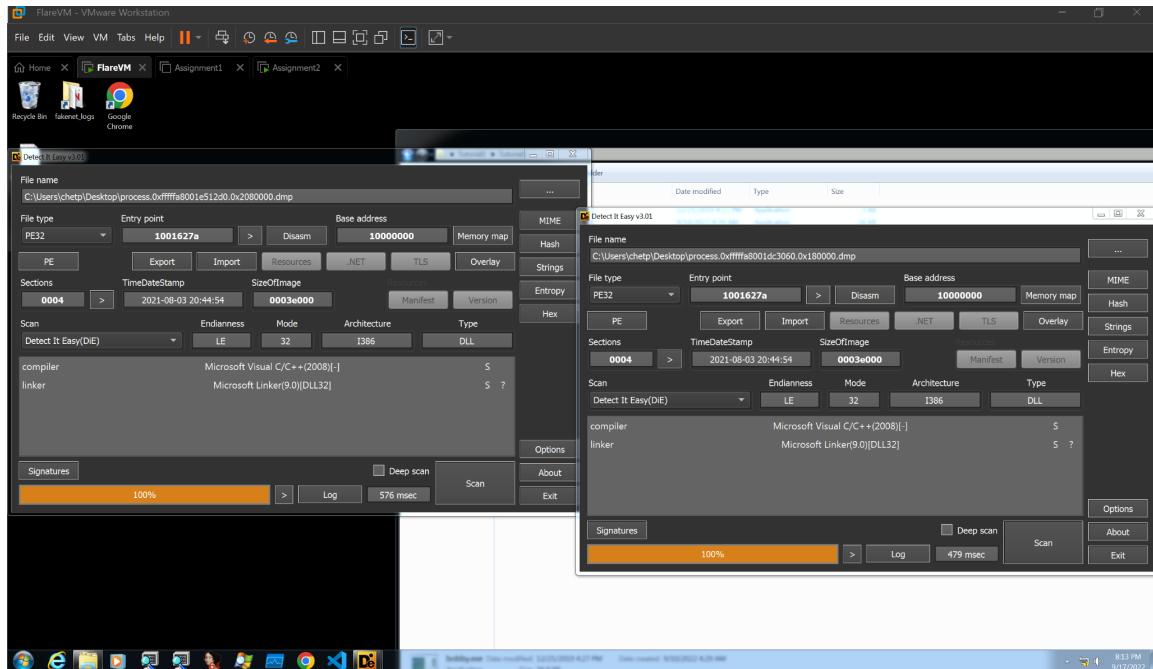
# Distribution Packages (released with Cobalt Strike 4.6)
d1f7eb1a34e17bb945f9b6bd656eac9d7837a00b2fadbc7d4d3e2621d2123a39 Cobalt Strike MacOSX Distribution
Package (cobaltstrike-dist.dmg 20220412)
f2d4a2312abc4357c19a876d335d1da41ca01e4d4c8c056890ed32c4a16055b4 Cobalt Strike Linux Distributions
Package (cobaltstrike-dist.tgz 20220412)
f696215f7fdf07a6525e91f864b39afc23e4e761cbc97c6ecb6c4ae0ffa8967 Cobalt Strike Windows Distribution
Package (cobaltstrike-dist.zip 20220412)

# Cobalt Strike 4.5 (December 14, 2021)
a5e980aac32d9c7af1d2326008537c66d55d7d9ccfc777eb732b2a31f4f7ee523 Cobalt Strike 4.5 Licensed
(cobaltstrike.jar)

# Cobalt Strike 4.4 (August 04, 2021)
7af9c759ac78da920395debb443b9007fdf51fa66a48f0fbdaafb30b00a8a858 Cobalt Strike 4.4 Licensed
(cobaltstrike.jar)

# Distribution Packages (released with Cobalt Strike 4.4)
5adf9d086a2f59be9095458f207de9e947a05696e63365a4da02acd17caa130 Cobalt Strike MacOSX Distribution
```

Also, the compile date is 2021-08-03 20:44:54 for the PE dumps (possibly beacon DLLs) obtained via malfind plugin. Version 4.4 was released on August 4 2021.



6 September 2018	3.12	2018-09-05T21:53:17 2018-09-05T21:54:00
4 May 2019	3.14	2019-04-18T23:51:29 2019-04-18T23:53:25
5 December 2019	4.0	2019-12-05T12:00:49 2019-12-05T12:01:49
22 February 2020	4.0	2020-02-21T04:55:08
25 June 2020	4.1	2020-06-23T19:17:48 2020-06-23T19:18:44 2020-06-23T19:21:26
6 November 2020	4.2	2020-11-03T01:27:30 2020-11-03T01:31:35
3 March 2021	4.3	2021-03-02T08:03:15 2021-03-09T16:42:17
17 March 2021	4.3	2021-03-16T17:37:35 2021-03-16T06:10:34

Table 9 – Cobalt Strike Beacon compilation timestamps

Based on this table, past versions typically had compilation timestamps in close proximity to the release date. Table was obtained from

<https://i.crn.com/sites/default/files/ckfinderimages/userfiles/images/crn/custom/2022/Blackberry/bb-e-book-finding-beacons-in-the-dark.pdf>