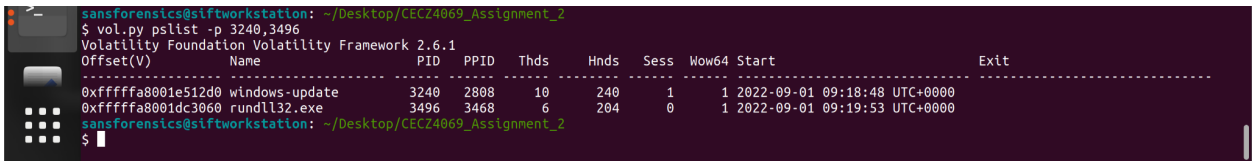
Process analysis

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

a. Process 1:

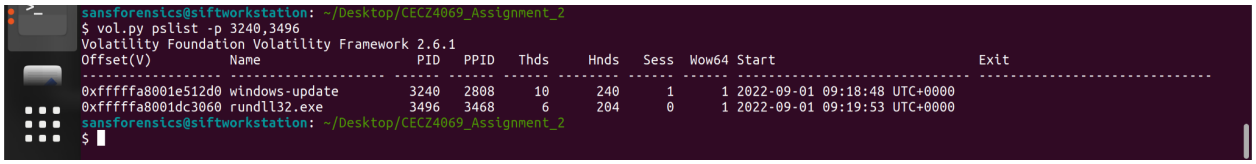
i. Give the name and PID. (0.5 mark)

Name: **windows-update**
PID: **3240**



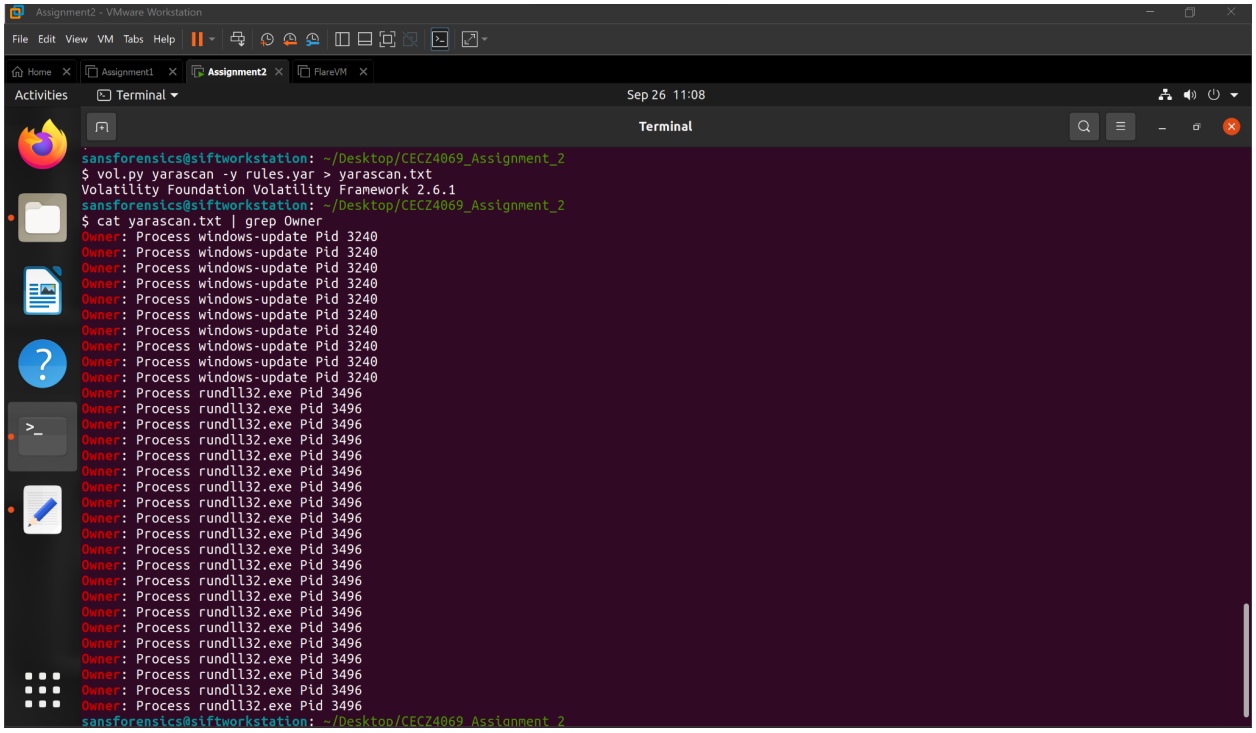
ii. Give the PID of the parent process. (0.5 mark)

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



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.



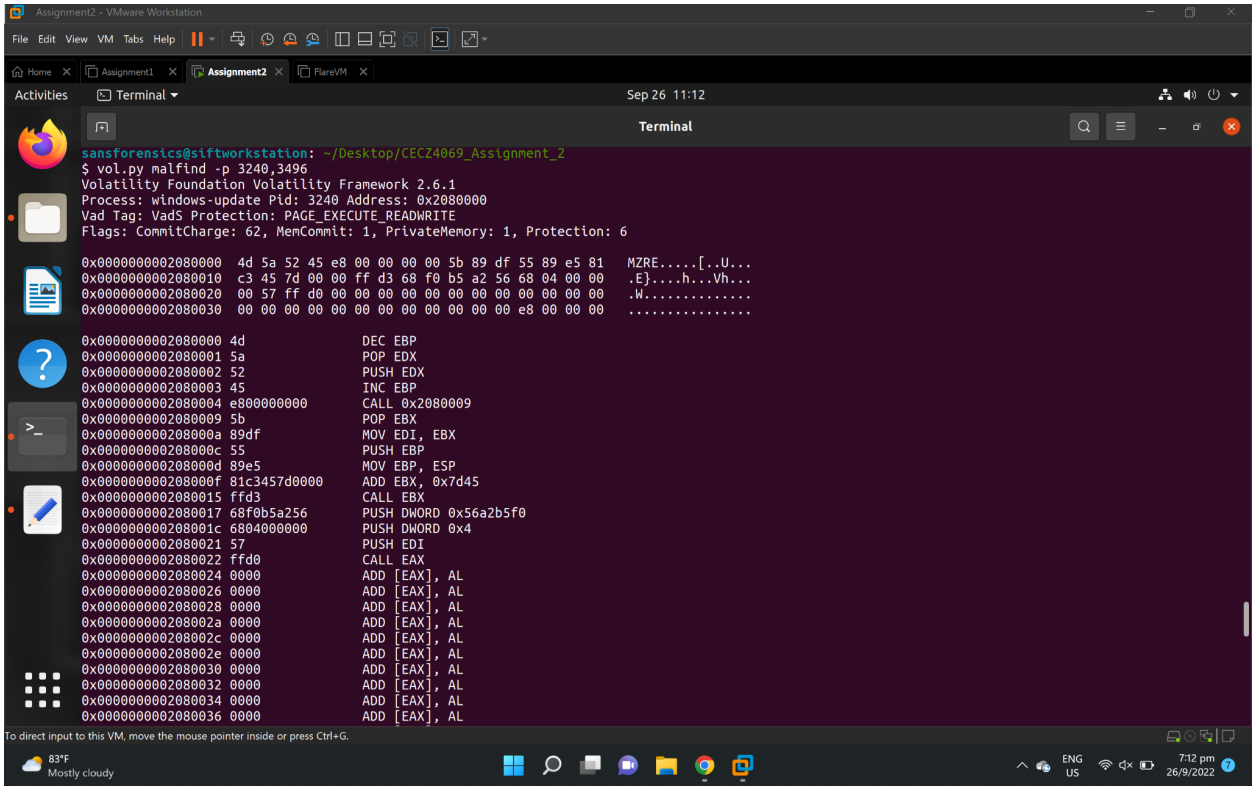
However, I did not conclude this process to be suspicious solely based on that single reason. If I may add, here are some of the other reasons:

- 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"

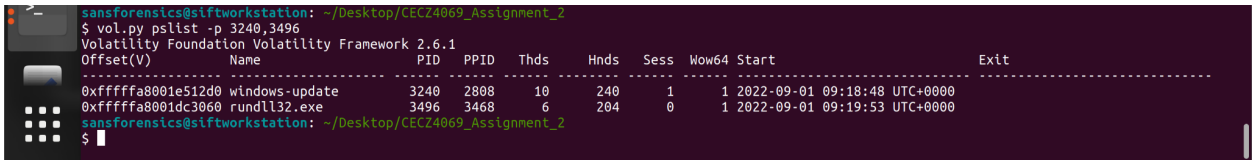
- Appeared in malfind output with MZ header indicating a PE file / executable, may be a possible process injection



b. Process 2:

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

Name: **rundll32.exe**
PID: **3496**

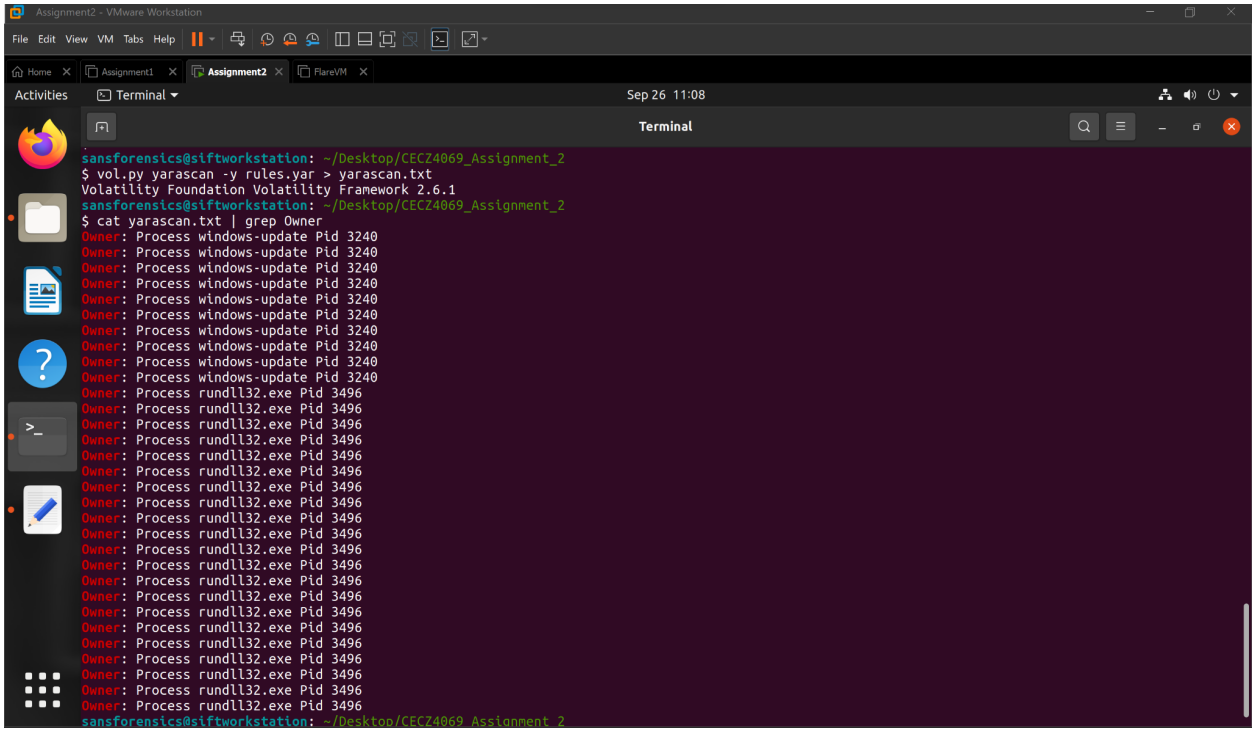


ii. Give the PID of the parent process. (0.5 mark)

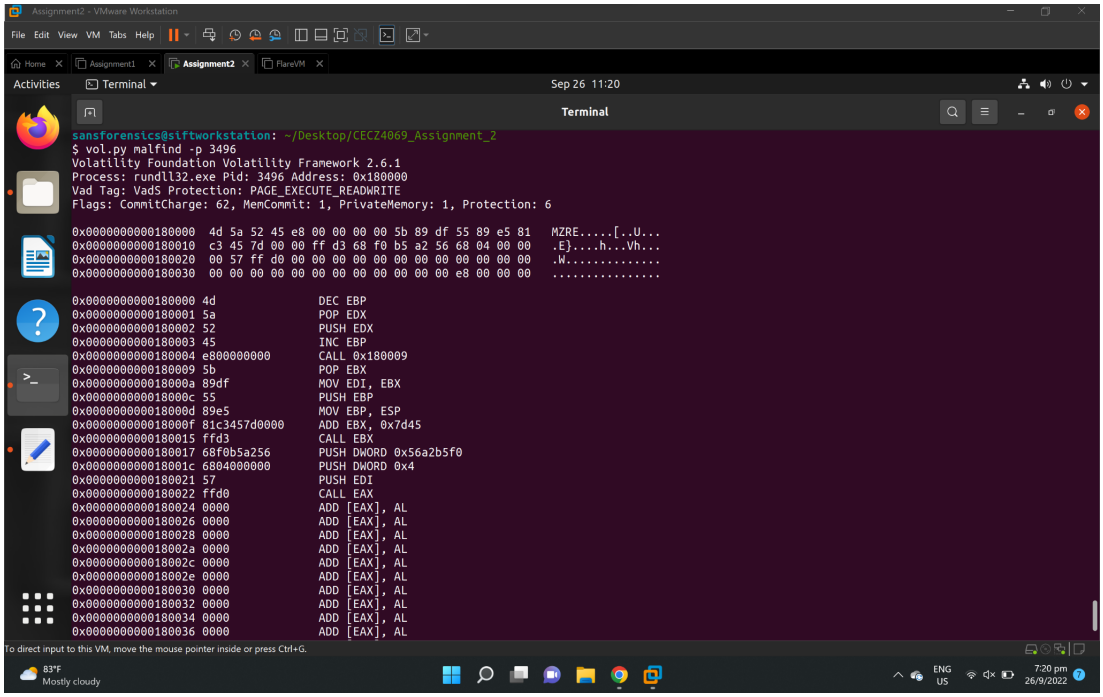
PID of the parent process (PPID): **3468**

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

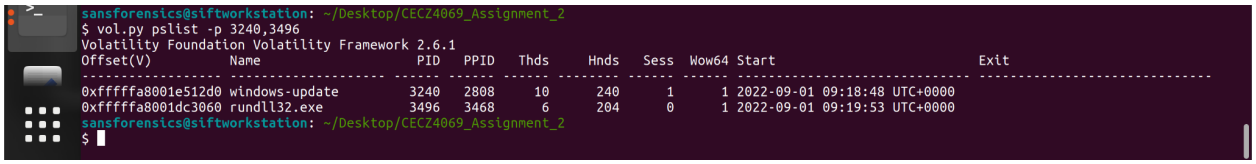
For me, the single strongest reason was that this process had multiple hits when I ran yarascan using rules.yar (obtained from https://github.com/Te-k/cobaltstrike), which contained some rules for detecting cobalt strike.



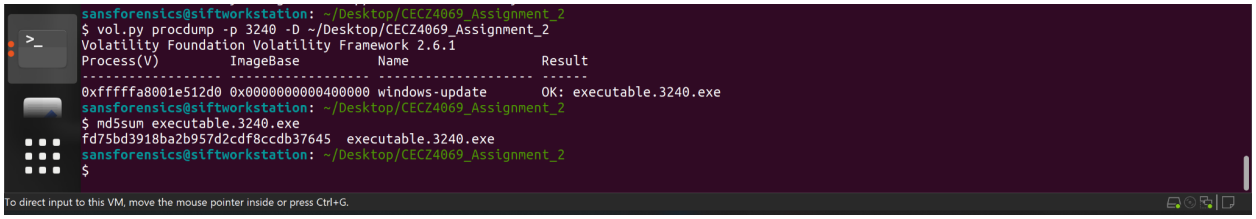
- It also appeared in malfind output with MZ header indicating a PE file / executable, may be a possible process injection



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)



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**



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-6f9d24af02d4

iii. Last Write Time: 2021-10-18 13:48:02 UTC+0000

(3 marks)

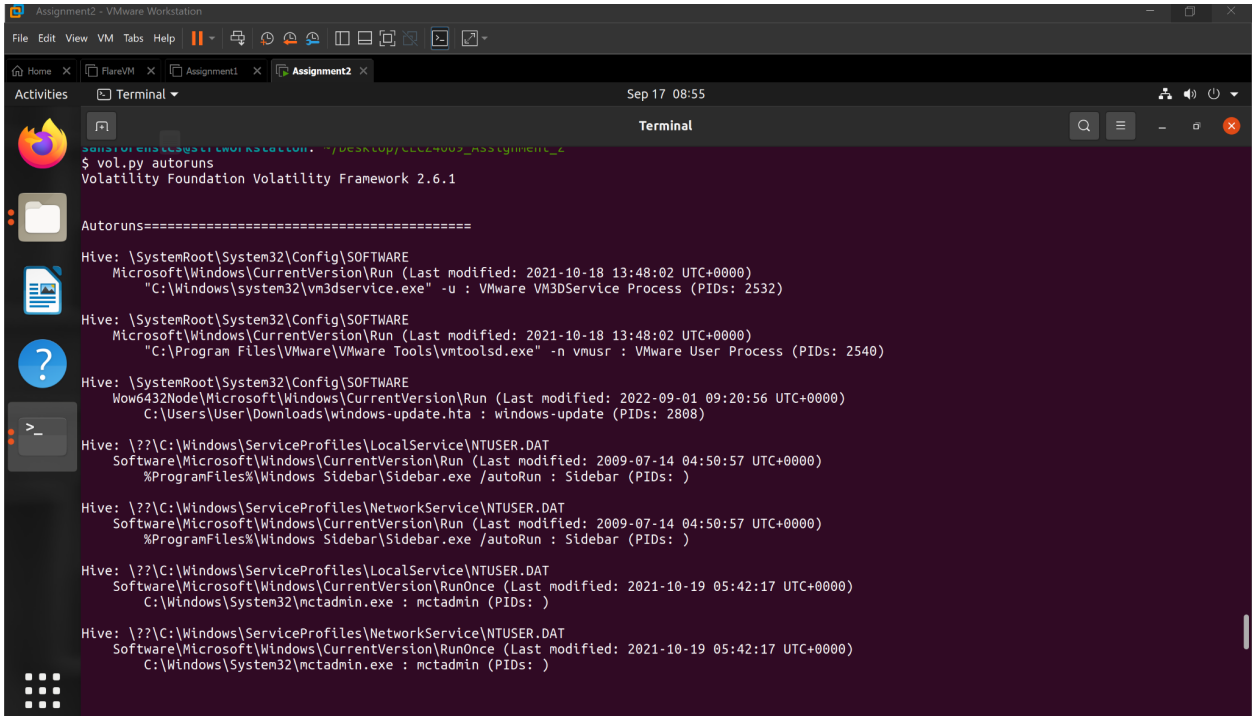
i. Registry Key: **Wow6432Node\Microsoft\Windows\CurrentVersion\Run**

ii. Registry Value:

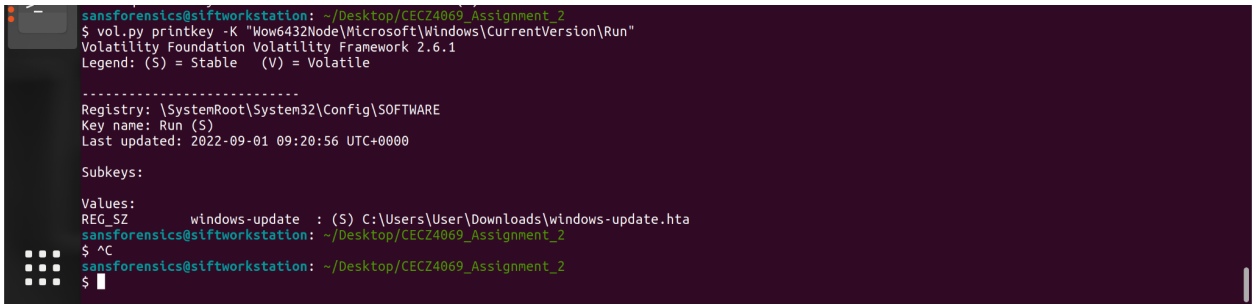
windows-update : (S) **C:\Users\User\Downloads\windows-update.hta**

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.



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

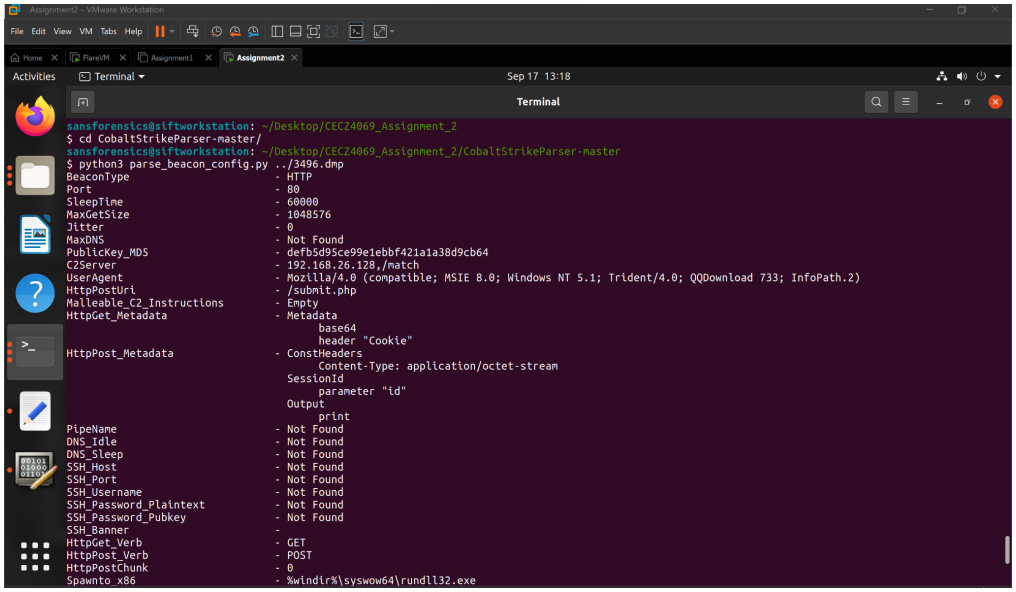


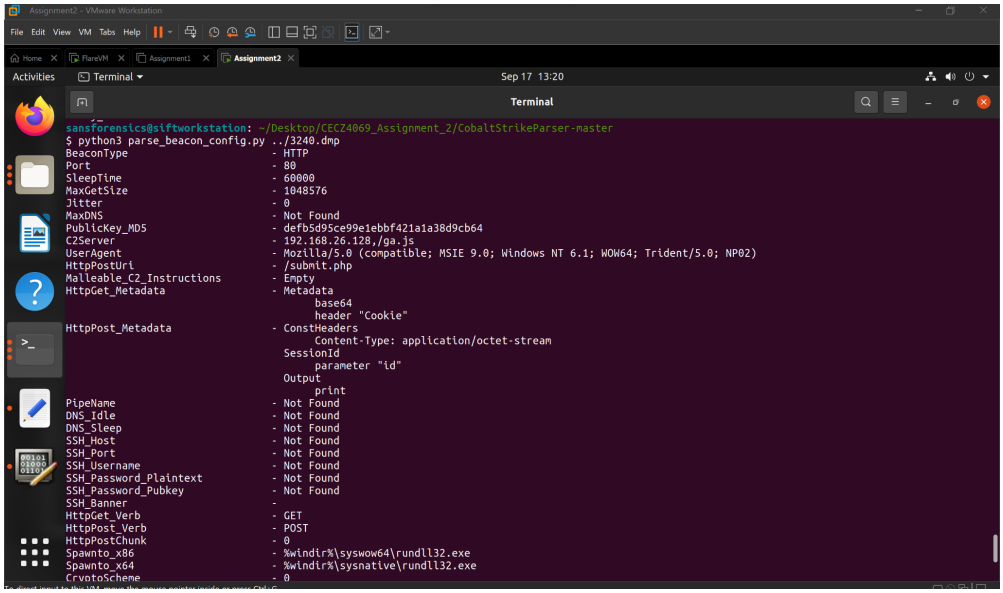
Network analysis:

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

IP address: **192.168.26.128**

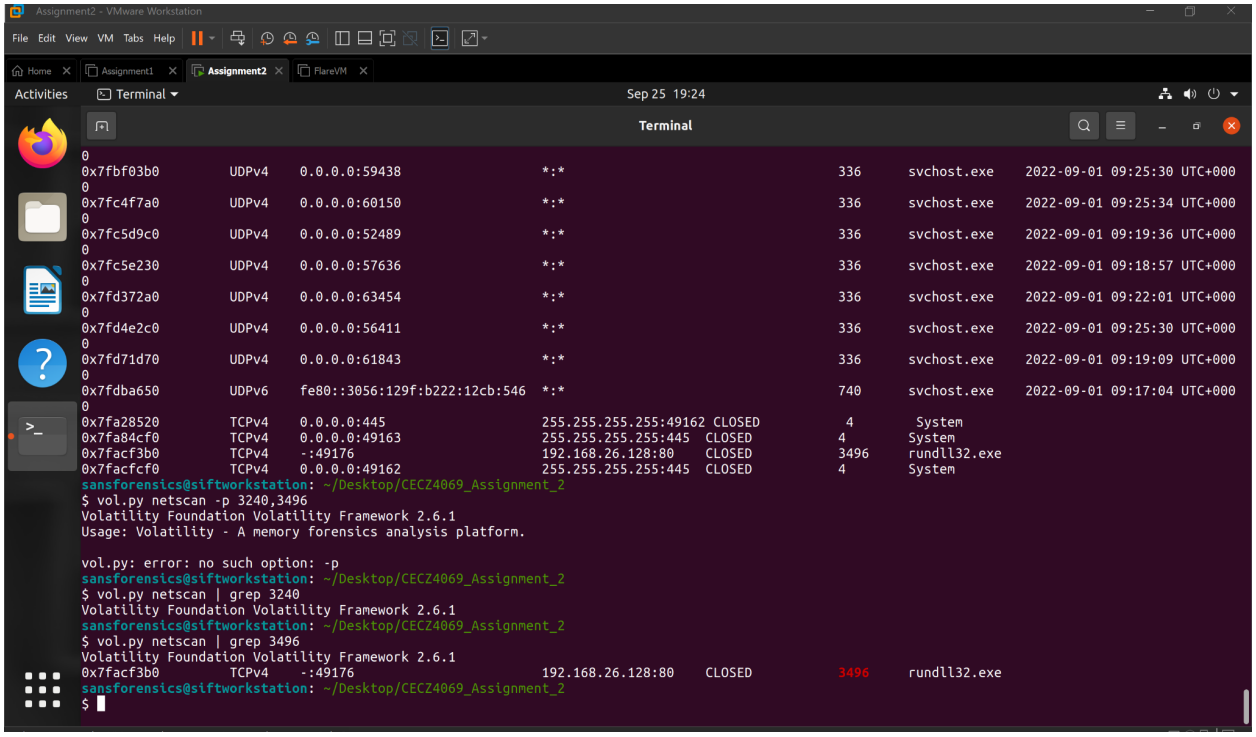
Port: **80**





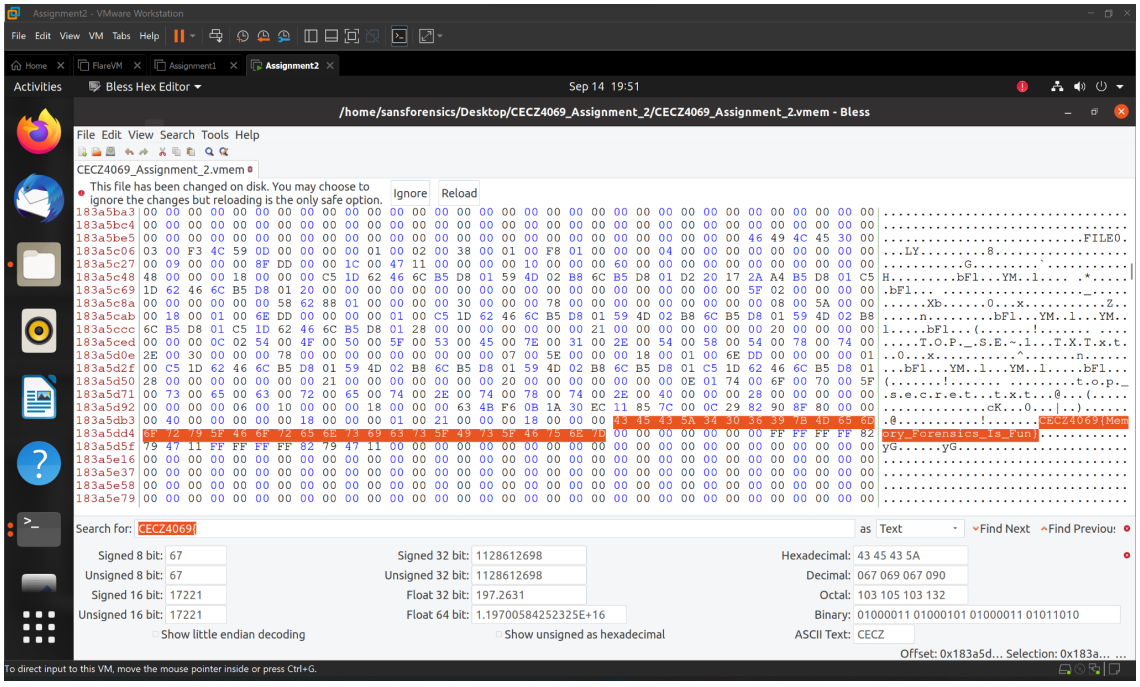
I extracted the memdump from both those processes identified in Q1 and ran a parse_beacon_config.py script on both to extract the cobalt strike configuration. (Source: https://github.com/Sentinel-One/CobaltStrikeParser) C2 server extracted was 192.168.26.128 and Beacon type is HTTP, which is over port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

To confirm, we can use netscan:



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)



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

CECZ4069{Memory_Forensics_Is_Fun}

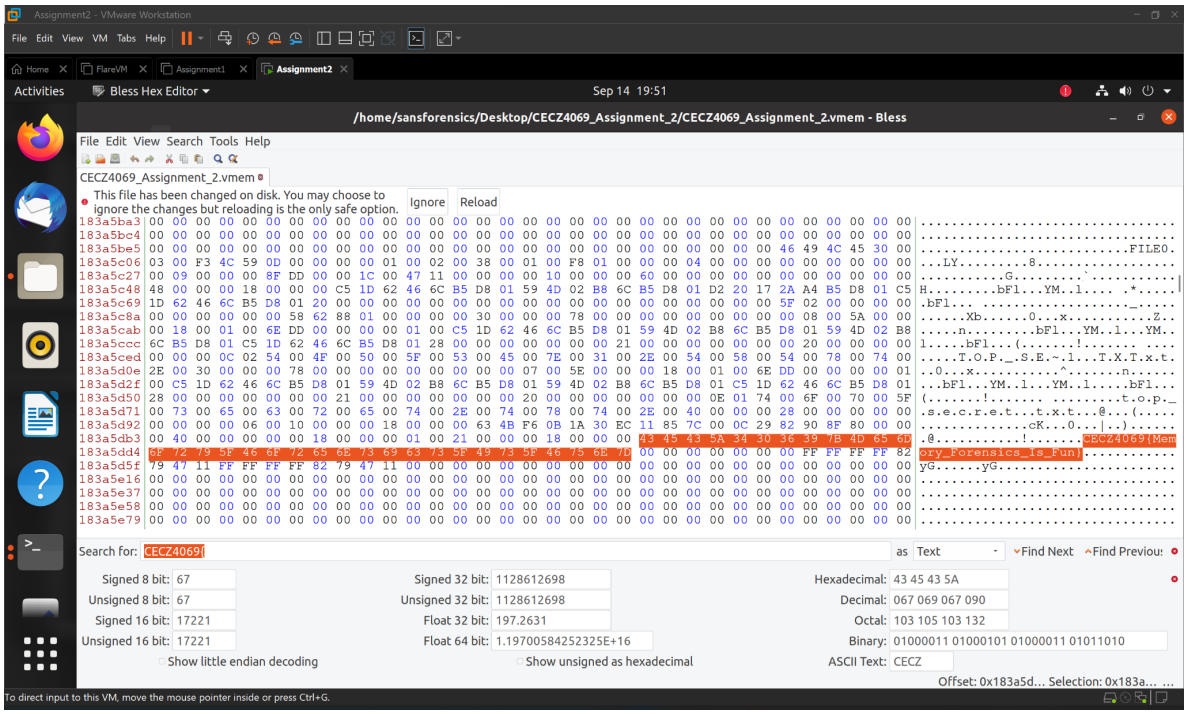
6. Provide the process name and PID of the userland process that contains the flag (in question 5) in their memory space. (2 marks)

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)

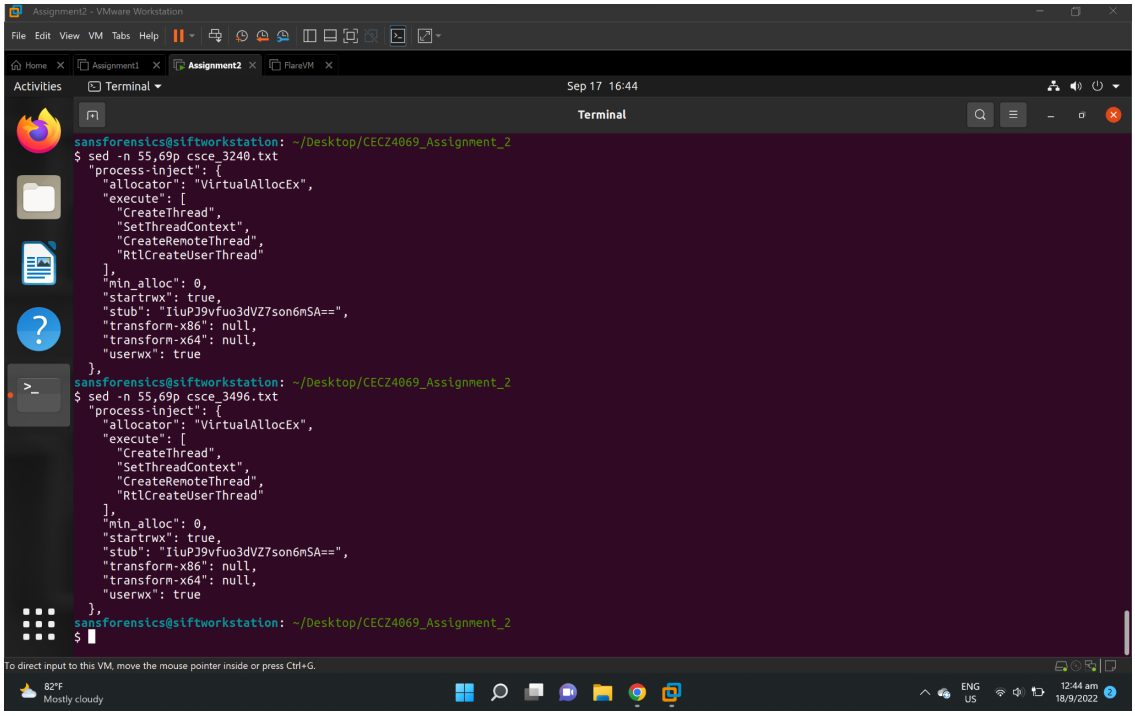
Version 4.4

Explanation:

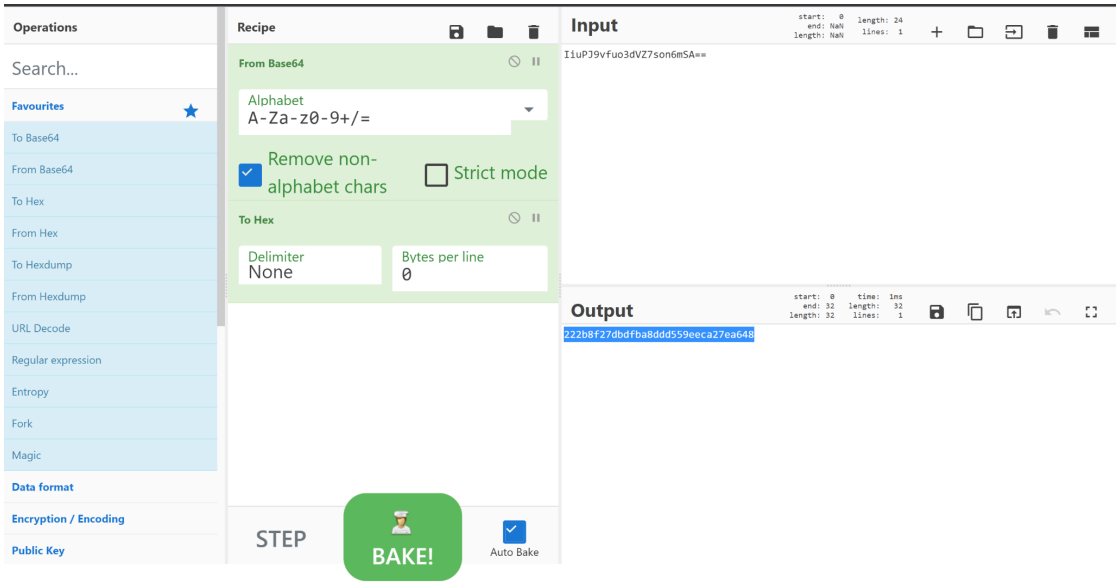
Thanks to this guide at https://www.elastic.co/security-labs/extracting-cobalt-strike-beacon-configurations, was able to correctly determine the version. Following the steps, I first used csce (https://github.com/strozfriedberg/cobaltstrike-config-extractor) to extract config info.

```
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 "IiuPJ9vfuo3dVZ7son6mSA==", the same for both process dumps.



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:

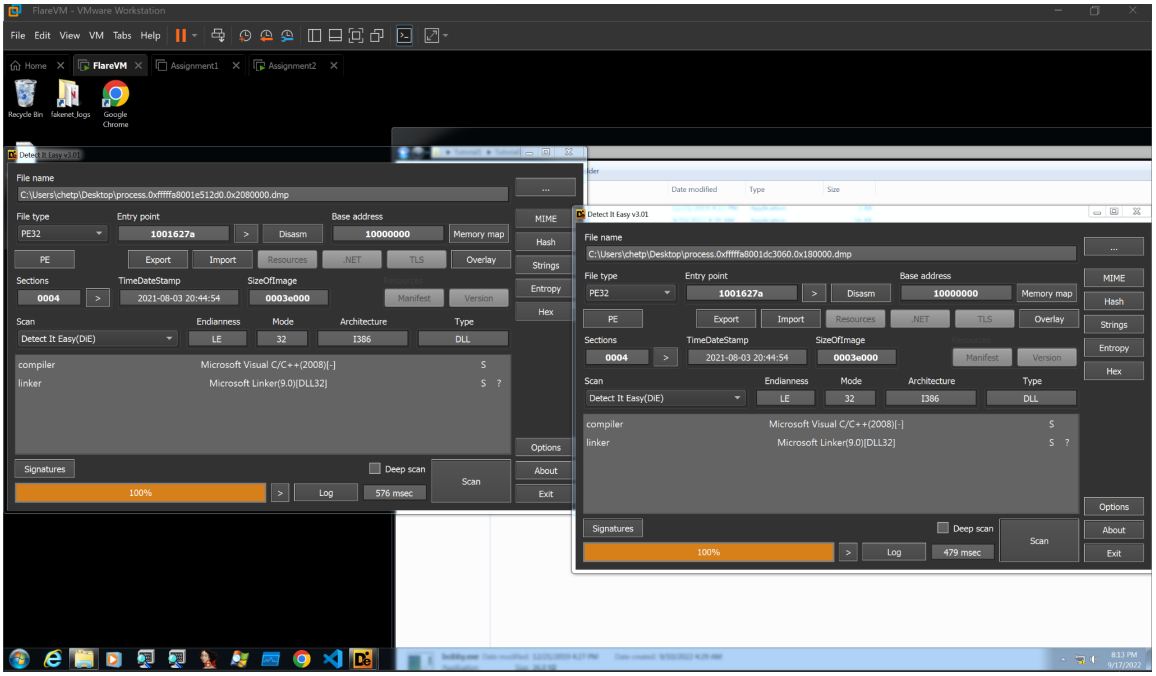


Inputting that MD5 hash into VirusTotal, we get its SHA256 hash (https://www.virustotal.com/gui/file/7af9c759ac78da920395debb443b9007fdf51fa66a48f0fbdaafb30b00a8a858). Then going to https://verify.cobaltstrike.com/, we can find the hash here:

```
# 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)
f696215f7fdf07a6525e91f864b39afc23e4e761cbca97c6ecb6c4ae0ffa8967       Cobalt Strike Windows Distribution Package (cobaltstrike-dist.zip 20220412)

# Cobalt Strike 4.5 (December 14, 2021)
a5e980aac32d9c7af1d2326008537c66d55d7d9ccf777eb732b2a31f4f7ee523       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)
5adf9d086a2f59be9095458f207de9e947a05696e63365a4da02acdc17caa130       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-ebook-finding-beacons-in-the-dark.pdf