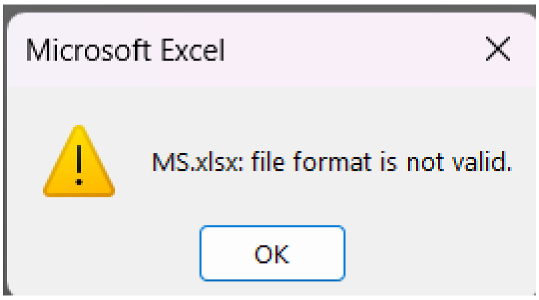
2024 HACKTHEON SEJONG

팀 이 름 Team Name	*Jun9k00k*
문 제 이 름 Question	*MS Office*

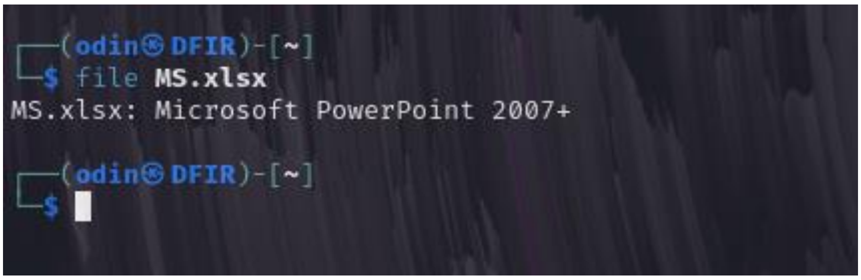
문제 풀이과정 작성 (캡처화면 필수) / Write-up Details (The screenshot is mandatory)

This challenge give us a file named MS.xlsx, and when I tried to open it, it's not work!



MS.xlsx: file format is not valid.

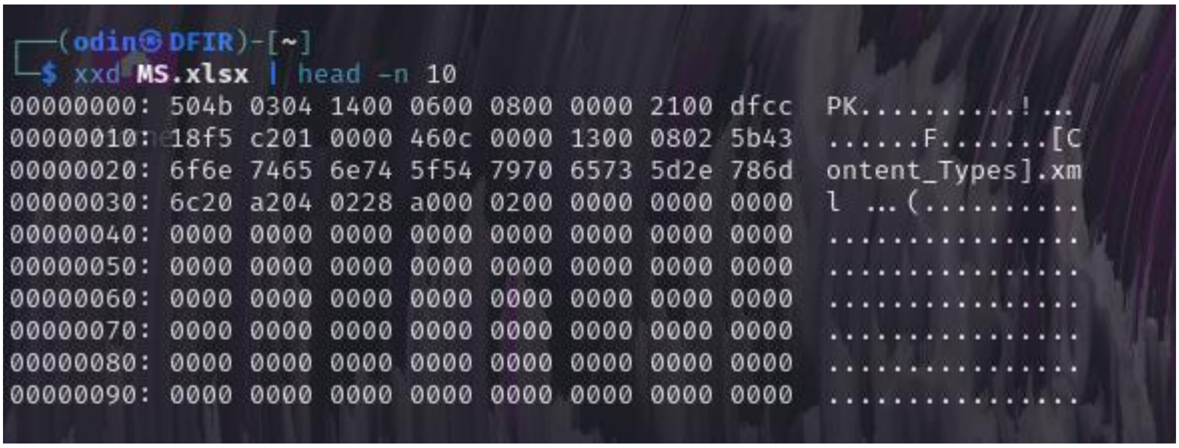
So that, I continued to check it by using **file** command:



```
┌──(odin㉿DFIR)-[~]
└─$ file MS.xlsx
MS.xlsx: Microsoft PowerPoint 2007+

┌──(odin㉿DFIR)-[~]
└─$ 
```

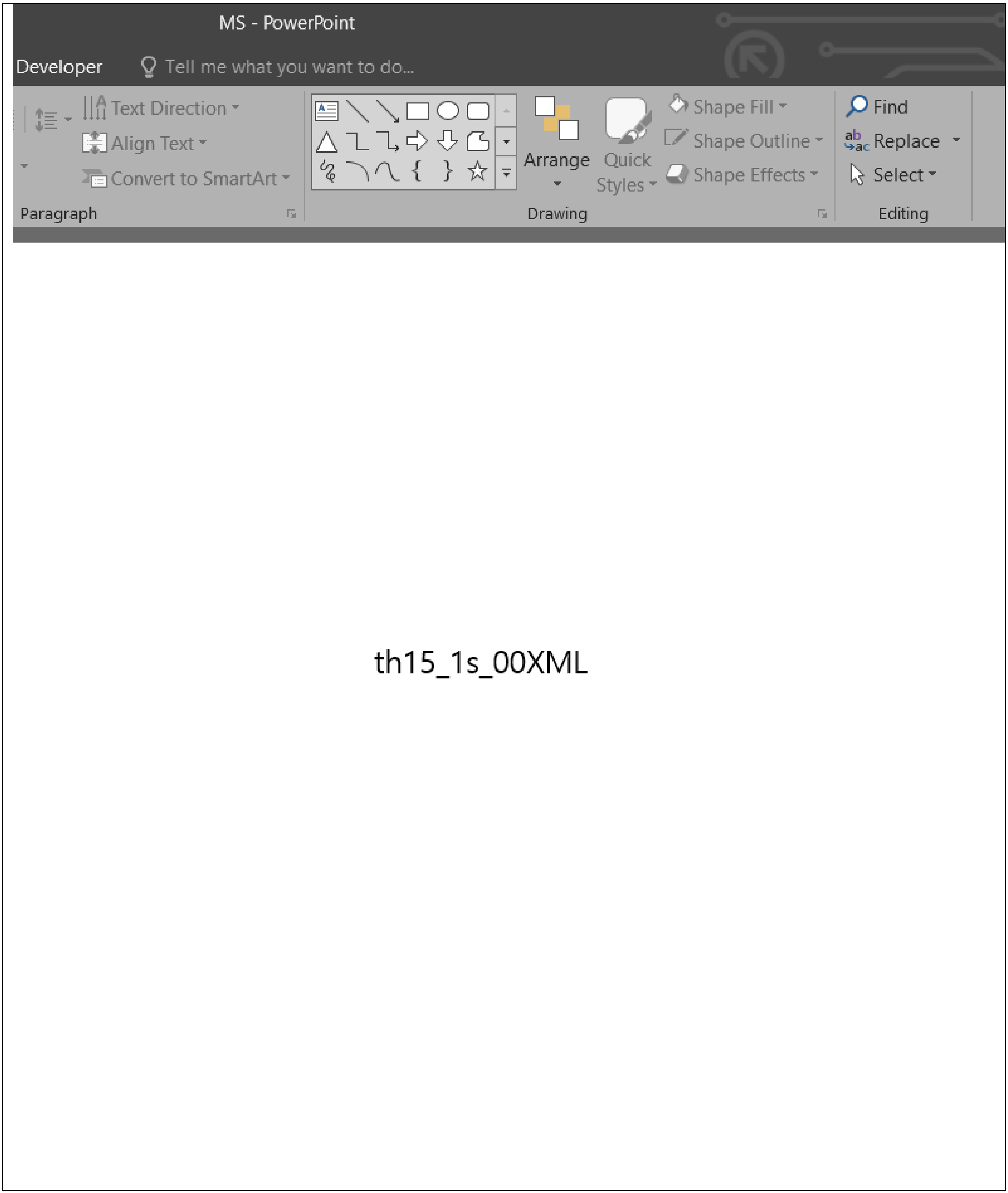
It's powerpoint file, and even if it's a Word file or Powerpoint file or Excel file, its origin is a compression of many files in a zip format, to ensure about my thinking, I checked file signature by using **xxd:**



```
┌──(odin㉿DFIR)-[~]
└─$ xxd MS.xlsx | head -n 10
00000000: 504b 0304 1400 0600 0800 0000 2100 dfcc  PK..........!...
00000010: 18f5 c201 0000 460c 0000 1300 0802 5b43  ......F.......[C
00000020: 6f6e 7465 6e74 5f54 7970 6573 5d2e 786d  ontent_Types].xm
00000030: 6c20 a204 0228 a000 0200 0000 0000 0000  l ...(..........
00000040: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000050: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000080: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000090: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Yeah it's correct! So we just do only thing is: rename the file from "MS.xlsx" to "MS.pptx" and enjoy the result:

2024 HACKTHEON SEJONG

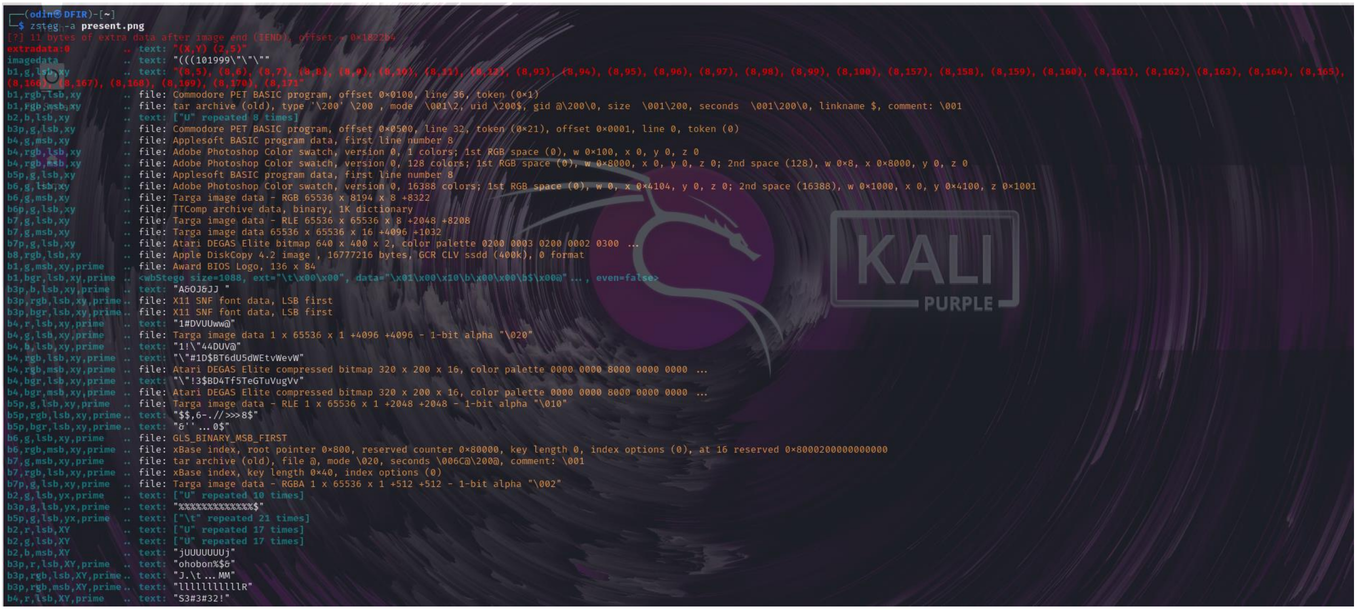


th15_1s_00XML

팀 이 름 **Team Name**	*Jun9k00k*
문 제 이 름 **Question**	*Stego4rt*

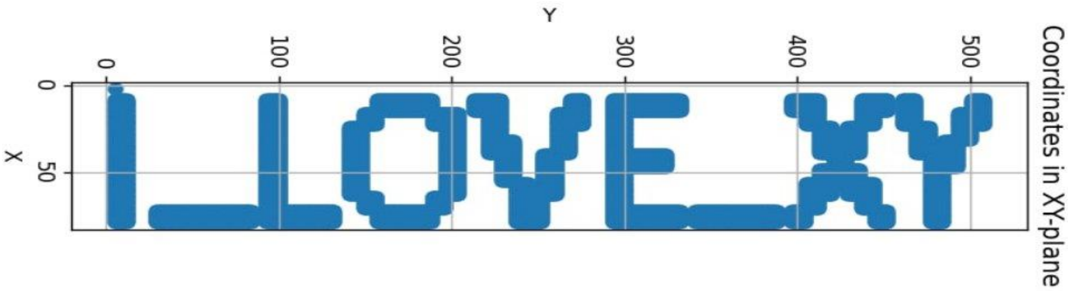
문제 풀이과정 작성 (캡처화면 필수) / Write-up Details (The screenshot is mandatory)

For this challenge, all we have is an image. Because of PNG format, I use **zsteg** to check for hidden information:



```
┌──(odin㉿DFIR)-[~]
└─$ zsteg -a present.png
[?] 11 bytes of extra data after image end (IEND), offset = 0x1822b4
extradata:0         .. text: "(X,Y) (2,5)"
imagedata           .. text: "(((101999\"\"\""
b1,g,lsb,xy         .. text: "(8,5), (8,6), (8,7), (8,8), (8,9), (8,10), (8,11), (8,12), (8,93), (8,94), (8,95), (8,96), (8,97), (8,98), (8,99), (8,100), (8,157), (8,158), (8,159), (8,160), (8,161), (8,162), (8,163), (8,164), (8,165),
(8,166), (8,167), (8,168), (8,169), (8,170), (8,171)"
b1,rgb,lsb,xy       .. file: Commodore PET BASIC program, offset 0x0100, line 36, token (0x1)
b1,rgb,msb,xy       .. file: tar archive (old), type '\200' \200 , mode  \001\2, uid \200$, gid @\200\0, size  \001\200, seconds  \001\200\0, linkname $, comment: \001
b2,b,lsb,xy         .. text: ["U" repeated 8 times]
b3p,g,lsb,xy        .. file: Commodore PET BASIC program, offset 0x0500, line 32, token (0x21), offset 0x0001, line 0, token (0)
b4,g,msb,xy         .. file: Applesoft BASIC program data, first line number 8
b4,rgb,lsb,xy       .. file: Adobe Photoshop Color swatch, version 0, 1 colors; 1st RGB space (0), w 0x100, x 0, y 0, z 0
b4,rgb,msb,xy       .. file: Adobe Photoshop Color swatch, version 0, 128 colors; 1st RGB space (0), w 0x8000, x 0, y 0, z 0; 2nd space (128), w 0x8, x 0x8000, y 0, z 0
b5p,g,lsb,xy        .. file: Applesoft BASIC program data, first line number 8
b6,g,lsb,xy         .. file: Adobe Photoshop Color swatch, version 0, 16388 colors; 1st RGB space (0), w 0, x 0x4104, y 0, z 0; 2nd space (16388), w 0x1000, x 0, y 0x4100, z 0x1001
b6,g,msb,xy         .. file: Targa image data - RGB 65536 x 8194 x 8 +8322
b6p,g,lsb,xy        .. file: TTComp archive data, binary, 1K dictionary
b7,g,lsb,xy         .. file: Targa image data - RLE 65536 x 65536 x 8 +2048 +8208
b7,g,msb,xy         .. file: Targa image data 65536 x 65536 x 16 +4096 +1032
b7p,g,lsb,xy        .. file: Atari DEGAS Elite bitmap 640 x 400 x 2, color palette 0200 0003 0200 0002 0300 ...
b8,rgb,lsb,xy       .. file: Apple DiskCopy 4.2 image , 16777216 bytes, GCR CLV ssdd (400k), 0 format
b1,g,msb,xy,prime   .. file: Award BIOS Logo, 136 x 84
b1,bgr,lsb,xy,prime .. <wbStego size=1088, ext="\t\x00\x00", data="\x01\x00\x10\b\x00\x00\b$\x00@"... , even=false>
b3p,b,lsb,xy,prime  .. text: "A&OJ&JJ "
b3p,rgb,lsb,xy,prime.. file: X11 SNF font data, LSB first
b3p,bgr,lsb,xy,prime.. file: X11 SNF font data, LSB first
b4,r,lsb,xy,prime   .. text: "1#DVUUww@"
b4,g,lsb,xy,prime   .. file: Targa image data 1 x 65536 x 1 +4096 +4096 - 1-bit alpha "\020"
b4,b,lsb,xy,prime   .. text: "1!\"44DUV@"
b4,rgb,lsb,xy,prime .. text: "\"#1D$BT6dU5dWEtvWevW"
b4,rgb,msb,xy,prime .. file: Atari DEGAS Elite compressed bitmap 320 x 200 x 16, color palette 0000 0000 8000 0000 0000 ...
b4,bgr,lsb,xy,prime .. text: "\"!3$BD4Tf5TeGTuVugVv"
b4,bgr,msb,xy,prime .. file: Atari DEGAS Elite compressed bitmap 320 x 200 x 16, color palette 0000 0000 8000 0000 0000 ...
b5p,g,lsb,xy,prime  .. file: Targa image data - RLE 1 x 65536 x 1 +2048 +2048 - 1-bit alpha "\010"
b5p,rgb,lsb,xy,prime.. text: "$$,6-.//>>>8$"
b5p,bgr,lsb,xy,prime.. text: "&''...0$"
b6,g,lsb,xy,prime   .. file: GLS_BINARY_MSB_FIRST
b6,rgb,msb,xy,prime .. file: xBase index, root pointer 0x800, reserved counter 0x80000, key length 0, index options (0), at 16 reserved 0x8000200000000000
b7,g,msb,xy,prime   .. file: tar archive (old), file @, mode \020, seconds \006C@\200@, comment: \001
b7,rgb,lsb,xy,prime .. file: xBase index, key length 0x40, index options (0)
b7p,g,lsb,xy,prime  .. file: Targa image data - RGBA 1 x 65536 x 1 +512 +512 - 1-bit alpha "\002"
b2,g,lsb,yx,prime   .. text: ["U" repeated 10 times]
b3p,g,lsb,yx,prime  .. text: "%%%%%%%%%%%%%$"
b5p,g,lsb,yx,prime  .. text: ["\t" repeated 21 times]
b2,r,lsb,XY         .. text: ["U" repeated 17 times]
b2,g,lsb,XY         .. text: ["U" repeated 17 times]
b2,b,msb,XY         .. text: "jUUUUUUUj"
b3p,r,lsb,XY,prime  .. text: "ohobon%$&"
b3p,rgb,lsb,XY,prime.. text: "J.\t...MM"
b3p,rgb,msb,XY,prime.. text: "lllllllllllR"
b4,r,lsb,XY,prime   .. text: "S3#3#32!"
```

At **b1, g, lsb, xy and extractdata:0** there's a string looks like coordinates, so I thought I had to display these coordinates in xy-plane, for that I wrote a small Python script to automate my process:

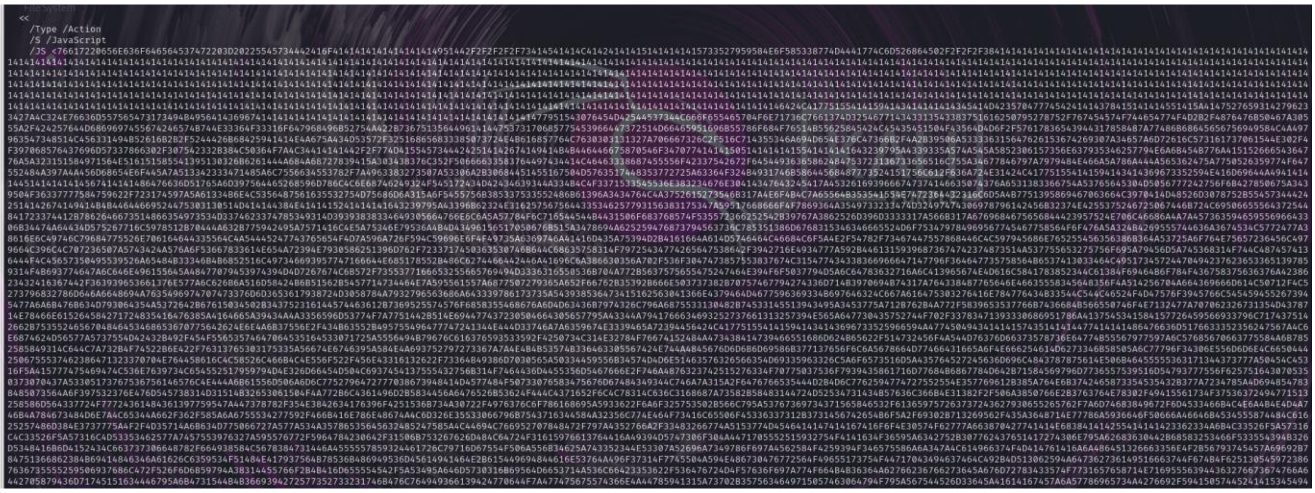


팀 이 름 Team Name	*Jun9k00k*
문 제 이 름 Question	*Confidential*

문제 풀이과정 작성 (캡처화면 필수) / Write-up Details (The screenshot is mandatory)

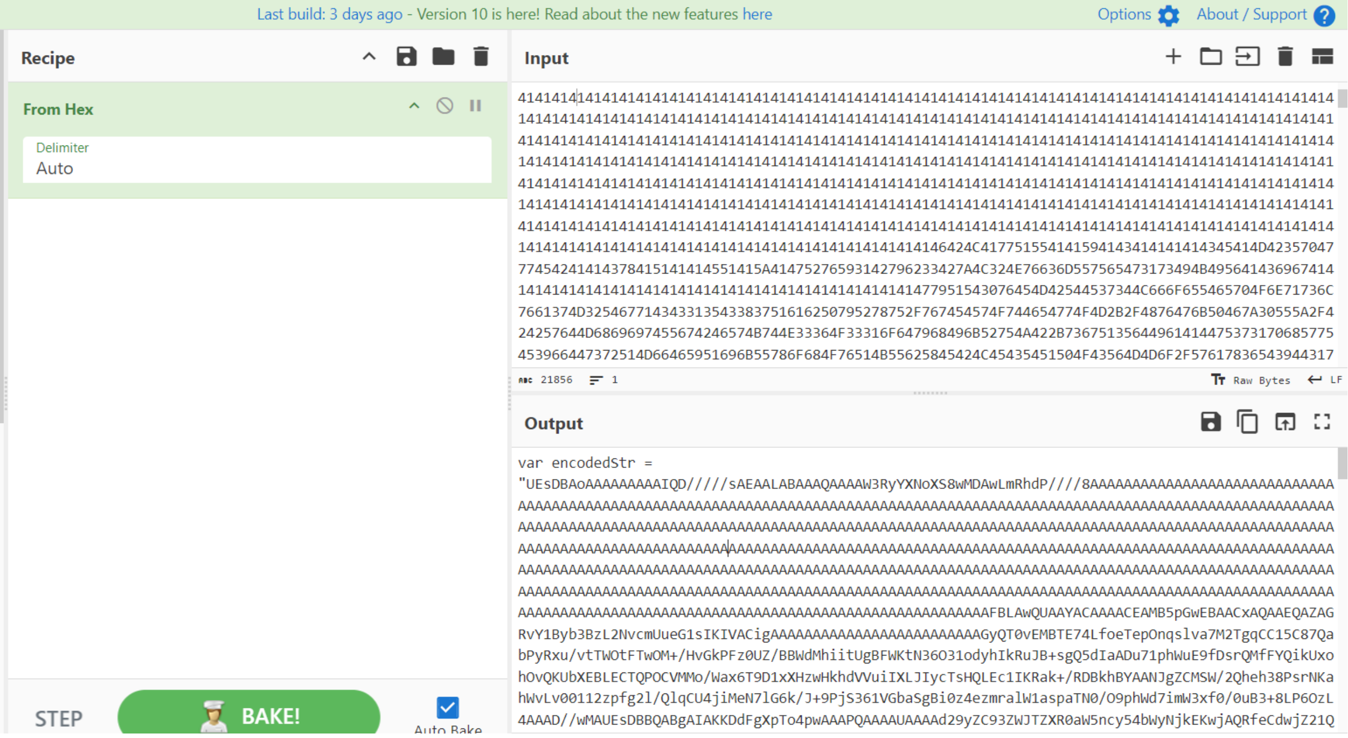
2024 HACKTHEON SEJONG

This challenge gave us a PDF file, and our mission is finding the hidden information inside the file. Not waiting, I used **pdf-parser** to parse all things inside this file. After parsed, I found a Javascript tag which contains a long hex value:



```
<<
  /Type /Action
  /S /JavaScript
  /JS <76617220656E636F646564537472203D2022554573442416F41414141414141414141414951442F2F2F2F2F7341454141414C41424141415141414141[illegible]
```

Put it in CyberChef, decode it and we have a **real** JS script:



Last build: 3 days ago - Version 10 is here! Read about the new features here
Options
About / Support
Recipe
From Hex
Input
Output
STEP
BAKE!
Auto Bake

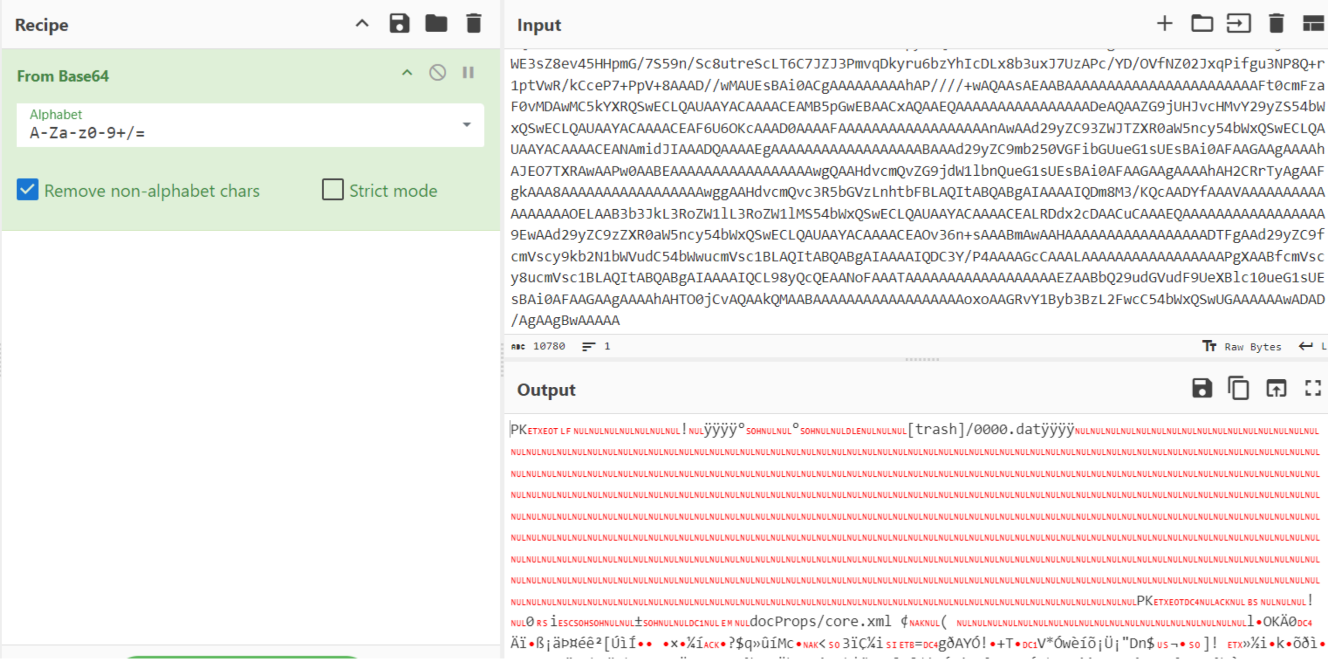
Delimiter: Auto

```
4141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414
1414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141
4141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414
1414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141
4141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414
1414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141
4141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414
14141414141414141414141414141414141414141414141414146424C4177515541415941434141414345414D42357047
7745424141437841514141455141 5A4147527659314279623342 7A4C324E76636D5575654731734 94B4956414369674 14
```

```
abc 21856    1                                   Raw Bytes   LF
```

```
var encodedStr =
"UEsDBAoAAAAAAAAAIQD/////sAEAALABAAAQAAAAW3RyYXNoXS8wMDAwLmRhdP////8AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFBLAwQUAAYACAAAACEAMB5pGwEBAACxAQAAEQAZAG
RvY1Byb3BzL2NvcmUueG1sIKIVACigAAAAAAAAAAAAAAAAAAAAAAAAAGyQT0vEMBTE74LfoeTepOnqslva7M2TgqCC15C87Qa
bPyRxu/vtTWOtFTwOM+/HvGkPFz0UZ/BBWdMhiitUgBFWKtN36O31odyhIkRuJB+sgQ5dIaADu71phWuE9fDsrQMfFYQikUxo
hOvQKUbXEBLECTQPOCVMMo/Wax6T9D1xXHzwHkhdVVuiIXLJIycTsHQLEc1IKRak+/RDBkhBYAANJgZCMSW/2Qheh38PsrNKa
hWvLv00112zpfg2l/QlqCU4jiMeN7lG6k/J+9PjS361VGbaSgBi0z4ezmralW1aspaTN0/O9phWd7imW3xf0/0uB3+8LP6OzL
4AAAD//wMAUEsDBBQABgAIAKKDdFgXpTo4pwAAAPQAAAAUAAAAd29yZC93ZWJTZXR0aW5ncy54bWyNjkEKwjAQRfeCdwjZ21Q
```

Read the code, you will see that it will decode base64 string and print it, so yeah, just decode it, and we got a Word document. Open file and enjoy the result:



Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars ☐ Strict mode

```
WE3sZ8ev45HHpmG/7S59n/Sc8utreScLT6C7JZJ3PmvqDkyru6bzYhIcDLx8b3uxJ7UzAPc/YD/OVfNZ02JxqPifgu3NP8Q+r
1ptVwR/kCceP7+PpV+8AAAD//wMAUEsBAi0ACgAAAAAAAAhAP////+wAQAAsAEAABAAAAAAAAAAAAAAAAAAAAAFt0cmFza
F0vMDAwMC5kYXRQSwECLQAUAAYACAAAACEAMB5pGwEBAACxAQAAEQAAAAAAAAAAAAAAAADeAQAAZG9jUHJvcHMvY29yZS54bW
xQSwECLQAUAAYACAAAACEAF6U6OKcAAAD0AAAAFAAAAAAAAAAAAAAAAAnAwAAd29yZC93ZWJTZXR0aW5ncy54bWxQSwECLQA
UAAYACAAAACEANAmidJIAAADQAAAAEgAAAAAAAAAAAAAAAAABAAAd29yZC9mb250VGFibGUueG1sUEsBAi0AFAAGAAgAAAAh
AJEO7TXRAwAAPw0AABEAAAAAAAAAAAAAAAAAwgQAAHdvcmQvZG9jdW1lbnQueG1sUEsBAi0AFAAGAAgAAAAhAH2CRrTyAgAAF
gkAAA8AAAAAAAAAAAAAAAAAwggAAHdvcmQvc3R5bGVzLnhtbFBLAQItABQABgAIAAAAIQDm8M3/KQcAADYfAAAVAAAAAAAAA
AAAAAAAOELAAB3b3JkL3RoZW1lL3RoZW1lMS54bWxQSwECLQAUAAYACAAAACEALRDdx2cDAACuCAAAEQAAAAAAAAAAAAAAAA
9EwAAd29yZC9zZXR0aW5ncy54bWxQSwECLQAUAAYACAAAACEAOv36n+sAAABmAwAAHAAAAAAAAAAAAAAAAADTFgAAd29yZC9f
cmVscy9kb2N1bWVudC54bWwucmVsc1BLAQItABQABgAIAAAAIQDC3Y/P4AAAAGcCAAALAAAAAAAAAAAAAAAAAPgXAABfcmVsc
y8ucmVsc1BLAQItABQABgAIAAAAIQCL98yQcQEAANoFAAATAAAAAAAAAAAAAAAAAEZAABbQ29udGVudF9UeXBlc10ueG1sUE
sBAi0AFAAGAAgAAAAhAHTO0jCvAQAAkQMAABAAAAAAAAAAAAAAAAAoxoAAGRvY1Byb3BzL2FwcC54bWxQSwUGAAAAAAwADAD
/AgAAgBwAAAAA
```

abc 10780 1 Raw Bytes

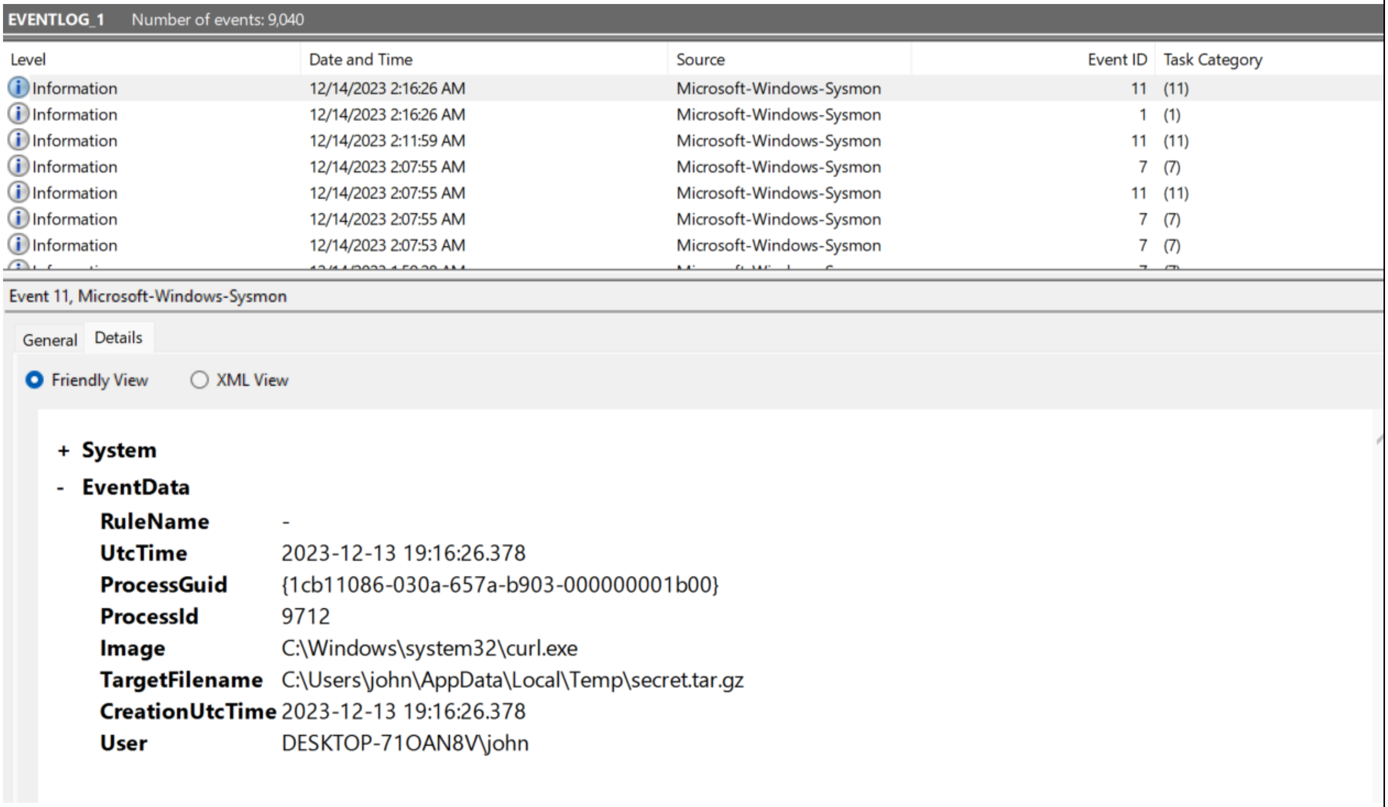
```
PKETXEOTLFNULNULNULNULNULNULNUL!NULÿÿÿÿ°SOHNULNUL°SOHNULNULDLENULNULNUL[trash]/0000.datÿÿÿÿNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULPKETXEOTDC4NULACKNULBSNULNULNUL!
NUL0RSiESCSOHSOHNULNUL±SOHNULNULDC1NULEMNULdocProps/core.xml ¢NAKNUL( NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULl•OKÄ0DC4
Äï•ß¡äÞ¤éê²[ÚìÍ•• •x•¼íACK•?$q»ûíMc•NAK<SO3ïÇ¼iSIETB=DC4gðAYÓ!•+T•DC1V*Ówèíõ¡Ü¡"Dn$US¬•SO]! ETX»½i•k•õðì•
```

I_cant_b3li3v3_y0u_put_a_fil3_1n_a_PDF

팀 이 름 Team Name	*Jun9k00k*
문 제 이 름 Question	*Rumor 5*
문제 풀이과정 작성 (캡처화면 필수) / Write-up Details (The screenshot is mandatory)	

2024 HACKTHEON SEJONG

After established the reverse shell, they use curl to extract a file called **secret.tar.gz:**



EVENTLOG_1 Number of events: 9,040

```
Level          Date and Time           Source                      Event ID  Task Category
Information    12/14/2023 2:16:26 AM   Microsoft-Windows-Sysmon          11  (11)
Information    12/14/2023 2:16:26 AM   Microsoft-Windows-Sysmon           1  (1)
Information    12/14/2023 2:11:59 AM   Microsoft-Windows-Sysmon          11  (11)
Information    12/14/2023 2:07:55 AM   Microsoft-Windows-Sysmon           7  (7)
Information    12/14/2023 2:07:55 AM   Microsoft-Windows-Sysmon          11  (11)
Information    12/14/2023 2:07:55 AM   Microsoft-Windows-Sysmon           7  (7)
Information    12/14/2023 2:07:53 AM   Microsoft-Windows-Sysmon           7  (7)
```

Event 11, Microsoft-Windows-Sysmon

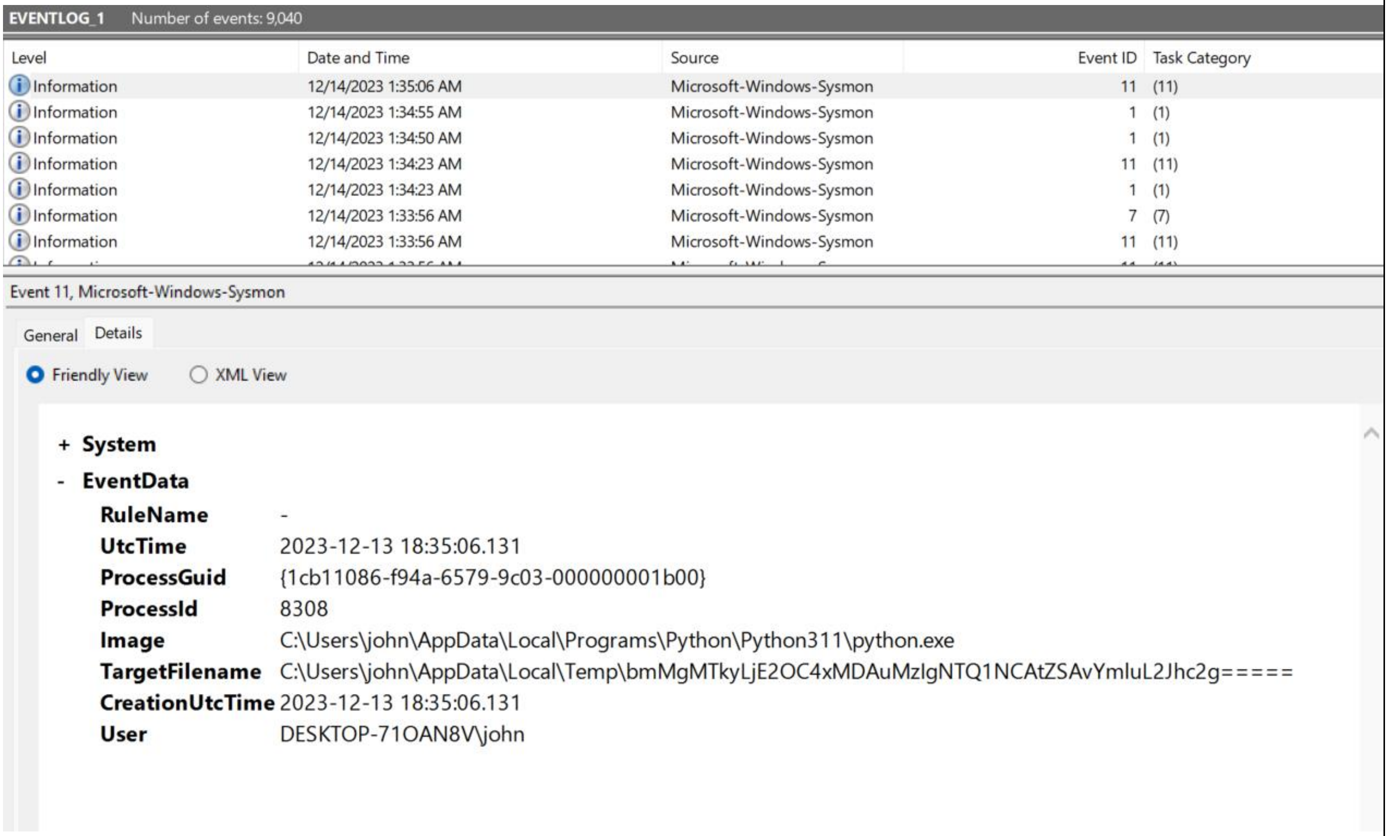
```
+ System
- EventData
    RuleName          -
    UtcTime           2023-12-13 19:16:26.378
    ProcessGuid       {1cb11086-030a-657a-b903-000000001b00}
    ProcessId         9712
    Image             C:\Windows\system32\curl.exe
    TargetFilename    C:\Users\john\AppData\Local\Temp\secret.tar.gz
    CreationUtcTime   2023-12-13 19:16:26.378
    User              DESKTOP-71OAN8V\john
```

팀 이 름 Team Name	*Jun9k00k*
문 제 이 름 Question	*Rumor 4*

문제 풀이과정 작성 (캡처화면 필수) / Write-up Details (The screenshot is mandatory)

2024 HACKTHEON SEJONG

Just follow the timeline after networking scan, you will find a file whose name is base64 string, decode it and we get the reverse shell follow the description:



EVENTLOG_1 Number of events: 9,040

Level	Date and Time	Source	Event ID	Task Category
Information	12/14/2023 1:35:06 AM	Microsoft-Windows-Sysmon	11	(11)
Information	12/14/2023 1:34:55 AM	Microsoft-Windows-Sysmon	1	(1)
Information	12/14/2023 1:34:50 AM	Microsoft-Windows-Sysmon	1	(1)
Information	12/14/2023 1:34:23 AM	Microsoft-Windows-Sysmon	11	(11)
Information	12/14/2023 1:34:23 AM	Microsoft-Windows-Sysmon	1	(1)
Information	12/14/2023 1:33:56 AM	Microsoft-Windows-Sysmon	7	(7)
Information	12/14/2023 1:33:56 AM	Microsoft-Windows-Sysmon	11	(11)

Event 11, Microsoft-Windows-Sysmon

+ System

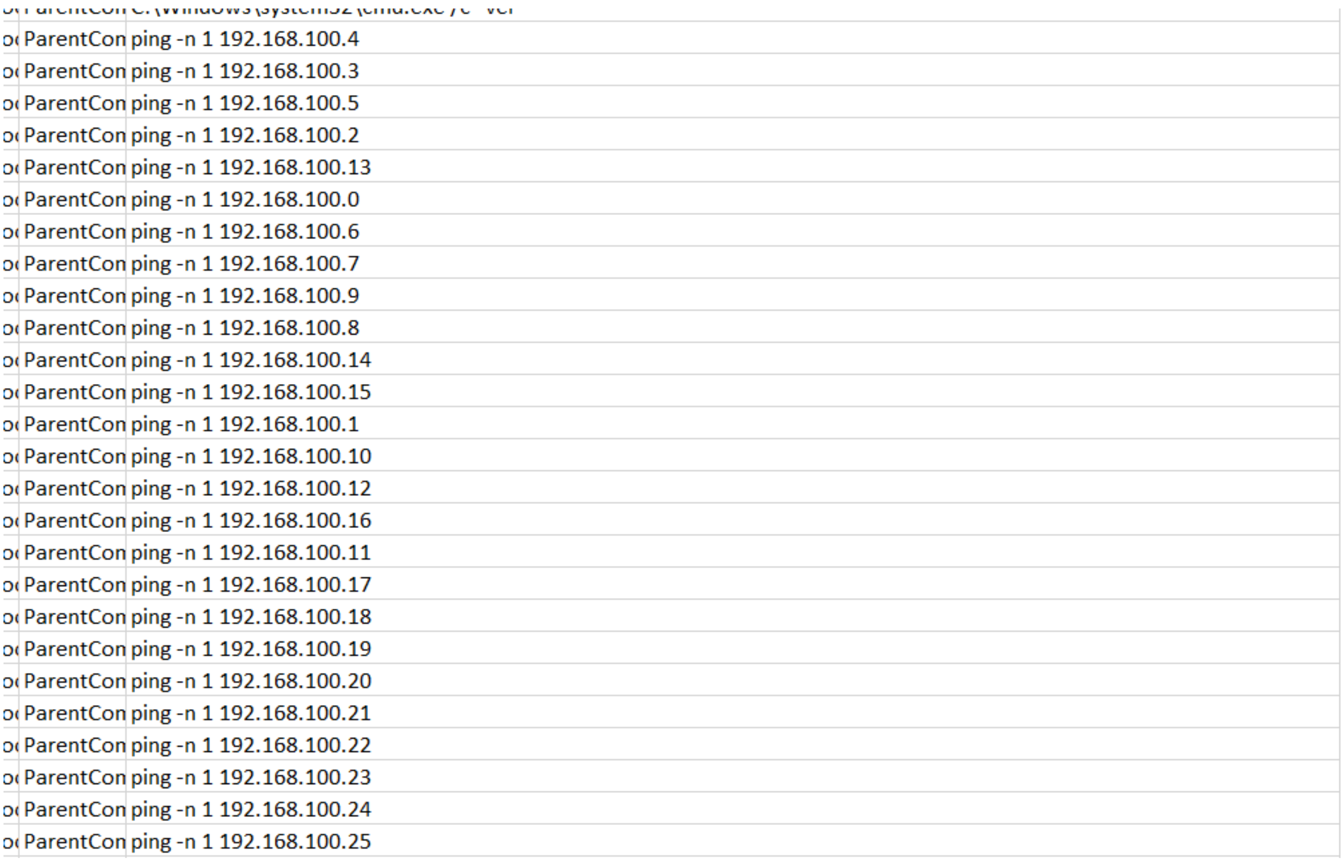
- EventData

RuleName	-
UtcTime	2023-12-13 18:35:06.131
ProcessGuid	{1cb11086-f94a-6579-9c03-000000001b00}
ProcessId	8308
Image	C:\Users\john\AppData\Local\Programs\Python\Python311\python.exe
TargetFilename	C:\Users\john\AppData\Local\Temp\bmMgMTkyLjE2OC4xMDAuMzIgNTQ1NCAtZSAvYmluL2Jhc2g=====
CreationUtcTime	2023-12-13 18:35:06.131
User	DESKTOP-71OAN8V\john

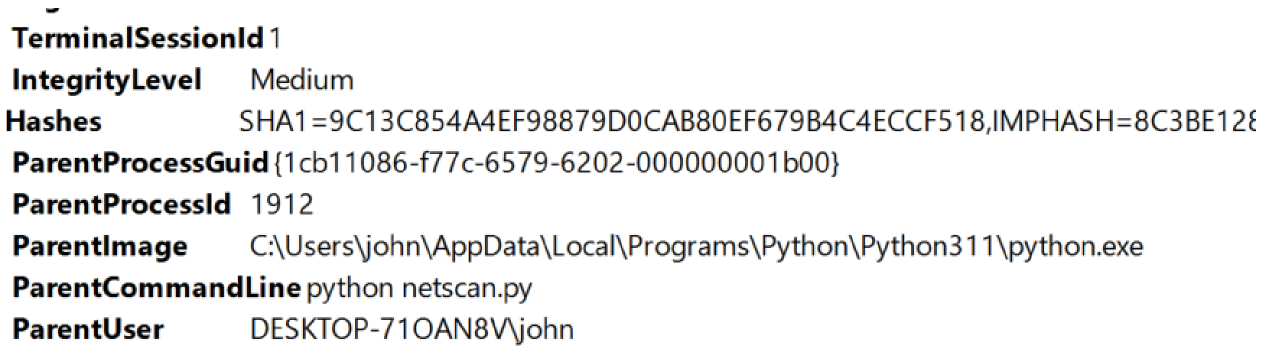
팀 이 름 **Team Name**	*Jun9k00k*
문 제 이 름 **Question**	*Rumor 3*

문제 풀이과정 작성 (캡처화면 필수) / Write-up Details (The screenshot is mandatory)

Follow deeper, we will see a list of scan actions which parent process is **netscan.py:**



```
o( ParentCon C:\Windows\System32\cmd.exe /c ver
o( ParentCon ping -n 1 192.168.100.4
o( ParentCon ping -n 1 192.168.100.3
o( ParentCon ping -n 1 192.168.100.5
o( ParentCon ping -n 1 192.168.100.2
o( ParentCon ping -n 1 192.168.100.13
o( ParentCon ping -n 1 192.168.100.0
o( ParentCon ping -n 1 192.168.100.6
o( ParentCon ping -n 1 192.168.100.7
o( ParentCon ping -n 1 192.168.100.9
o( ParentCon ping -n 1 192.168.100.8
o( ParentCon ping -n 1 192.168.100.14
o( ParentCon ping -n 1 192.168.100.15
o( ParentCon ping -n 1 192.168.100.1
o( ParentCon ping -n 1 192.168.100.10
o( ParentCon ping -n 1 192.168.100.12
o( ParentCon ping -n 1 192.168.100.16
o( ParentCon ping -n 1 192.168.100.11
o( ParentCon ping -n 1 192.168.100.17
o( ParentCon ping -n 1 192.168.100.18
o( ParentCon ping -n 1 192.168.100.19
o( ParentCon ping -n 1 192.168.100.20
o( ParentCon ping -n 1 192.168.100.21
o( ParentCon ping -n 1 192.168.100.22
o( ParentCon ping -n 1 192.168.100.23
o( ParentCon ping -n 1 192.168.100.24
o( ParentCon ping -n 1 192.168.100.25
```



```
TerminalSessionId   1
IntegrityLevel      Medium
Hashes              SHA1=9C13C854A4EF98879D0CAB80EF679B4C4ECCF518,IMPHASH=8C3BE12
ParentProcessGuid   {1cb11086-f77c-6579-6202-000000001b00}
ParentProcessId     1912
ParentImage         C:\Users\john\AppData\Local\Programs\Python\Python311\python.exe
ParentCommandLine   python netscan.py
ParentUser          DESKTOP-71OAN8V\john
```

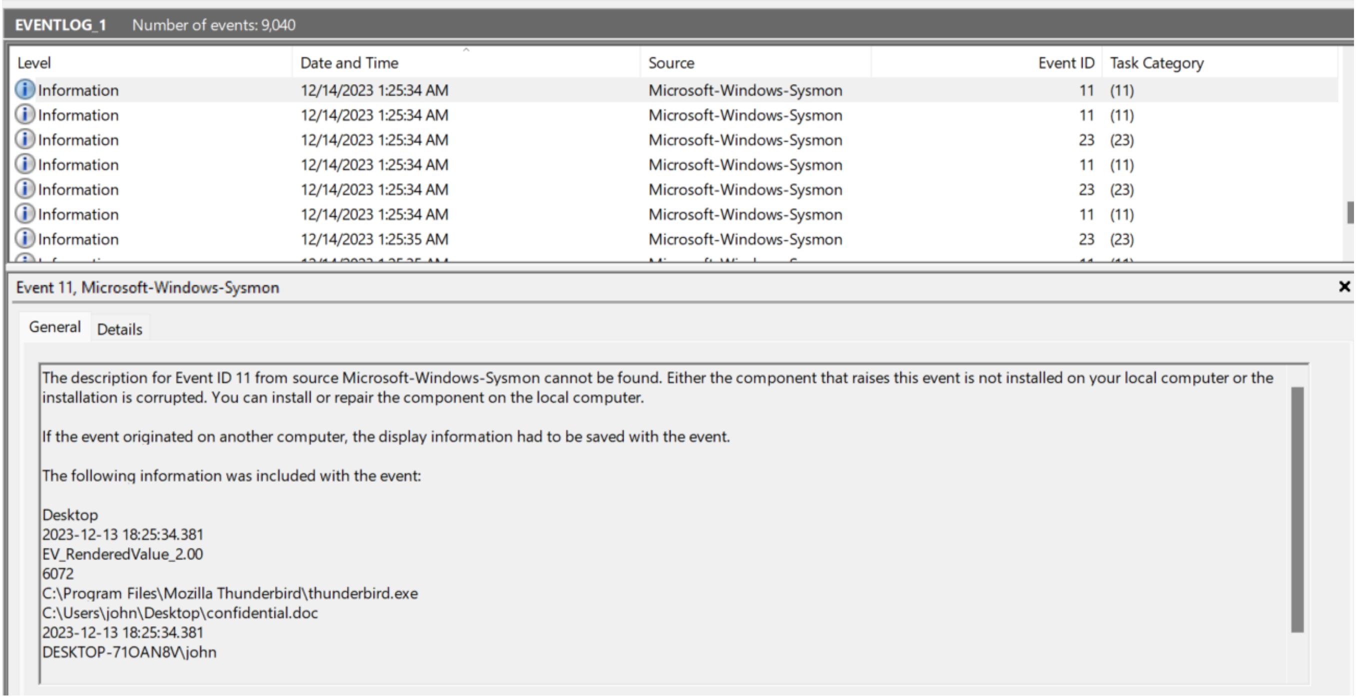
The range was 192.168.100.1 to 192.168.100.255 => **192.168.100.0/24**

팀 이 름 **Team Name**	*Jun9k00k*
문 제 이 름 **Question**	*Rumor 2*

문제 풀이과정 작성 (캡처화면 필수) / Write-up Details (The screenshot is mandatory)

2024 HACKTHEON SEJONG

Continuing to analyze the log, a suspicious activity appeared around **1:26 AM 14/12/2023**:



EVENTLOG_1 Number of events: 9,040

Level	Date and Time	Source	Event ID	Task Category
Information	12/14/2023 1:25:34 AM	Microsoft-Windows-Sysmon	11	(11)
Information	12/14/2023 1:25:34 AM	Microsoft-Windows-Sysmon	11	(11)
Information	12/14/2023 1:25:34 AM	Microsoft-Windows-Sysmon	23	(23)
Information	12/14/2023 1:25:34 AM	Microsoft-Windows-Sysmon	11	(11)
Information	12/14/2023 1:25:34 AM	Microsoft-Windows-Sysmon	23	(23)
Information	12/14/2023 1:25:34 AM	Microsoft-Windows-Sysmon	11	(11)
Information	12/14/2023 1:25:35 AM	Microsoft-Windows-Sysmon	23	(23)

Event 11, Microsoft-Windows-Sysmon

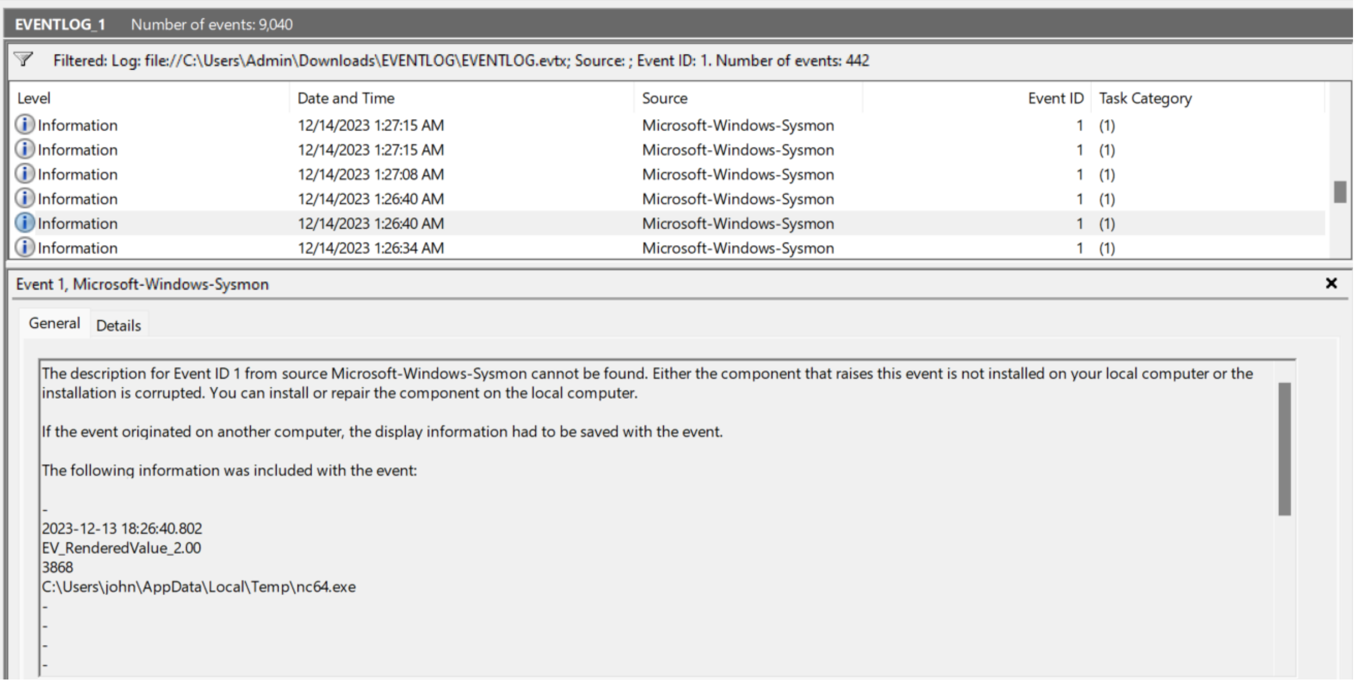
The description for Event ID 11 from source Microsoft-Windows-Sysmon cannot be found. Either the component that raises this event is not installed on your local computer or the installation is corrupted. You can install or repair the component on the local computer.

If the event originated on another computer, the display information had to be saved with the event.

The following information was included with the event:

```
Desktop
2023-12-13 18:25:34.381
EV_RenderedValue_2.00
6072
C:\Program Files\Mozilla Thunderbird\thunderbird.exe
C:\Users\john\Desktop\confidential.doc
2023-12-13 18:25:34.381
DESKTOP-71OAN8V\john
```

Beside, while I was analysing, I found netcat was executed near the time word document was downloaded. Hence, this highly suggest a malicious document was sent and downloaded via email phishing:



EVENTLOG_1 Number of events: 9,040

Filtered: Log: file://C:\Users\Admin\Downloads\EVENTLOG\EVENTLOG.evtx; Source: ; Event ID: 1. Number of events: 442

Level	Date and Time	Source	Event ID	Task Category
Information	12/14/2023 1:27:15 AM	Microsoft-Windows-Sysmon	1	(1)
Information	12/14/2023 1:27:15 AM	Microsoft-Windows-Sysmon	1	(1)
Information	12/14/2023 1:27:08 AM	Microsoft-Windows-Sysmon	1	(1)
Information	12/14/2023 1:26:40 AM	Microsoft-Windows-Sysmon	1	(1)
Information	12/14/2023 1:26:40 AM	Microsoft-Windows-Sysmon	1	(1)
Information	12/14/2023 1:26:34 AM	Microsoft-Windows-Sysmon	1	(1)

Event 1, Microsoft-Windows-Sysmon

The description for Event ID 1 from source Microsoft-Windows-Sysmon cannot be found. Either the component that raises this event is not installed on your local computer or the installation is corrupted. You can install or repair the component on the local computer.

If the event originated on another computer, the display information had to be saved with the event.

The following information was included with the event:

```
-
2023-12-13 18:26:40.802
EV_RenderedValue_2.00
3868
C:\Users\john\AppData\Local\Temp\nc64.exe
-
-
-
-
```

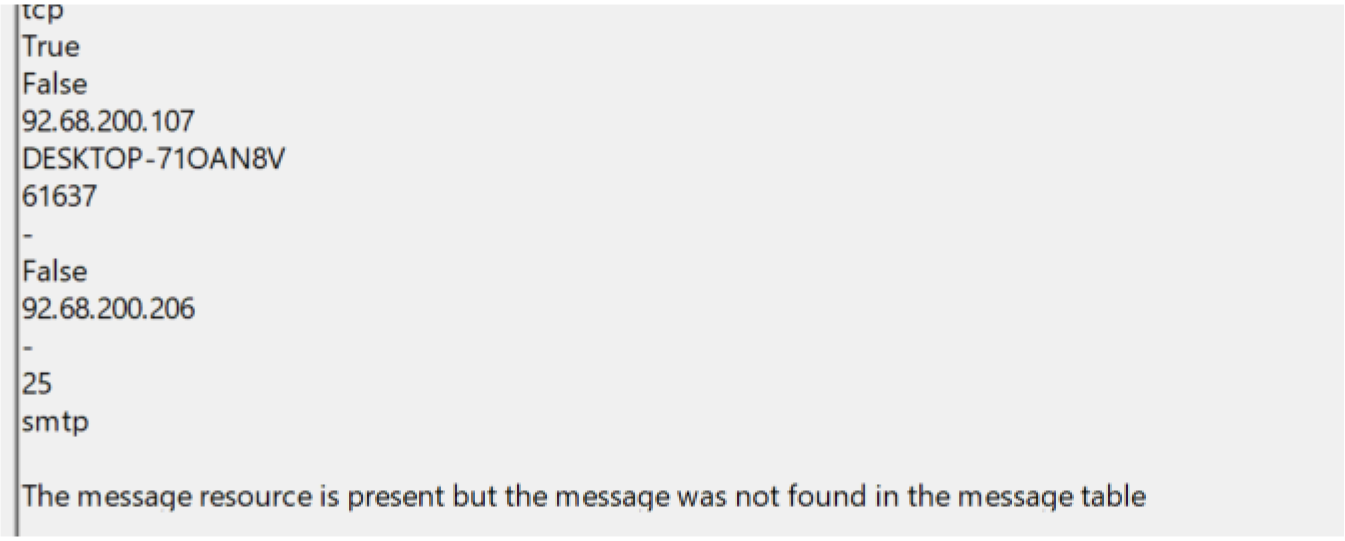
⇨ Event ID: 3868

팀 이 름 Team Name	*Jun9k00k*
문 제 이 름 Question	*Rumor 1*

문제 풀이과정 작성 (캡처화면 필수) / Write-up Details (The screenshot is mandatory)

In this challenge, we have a Windows Event Log file, it's a file that record all activities happened in a Windows computer. Follow the question, they asked us to find the IP address of the mail server used by the PC.

After analysing this log, it seems the event log is Sysmon. Moreover, they asked about mail server, so I searched **SMTP** and I found the answer:



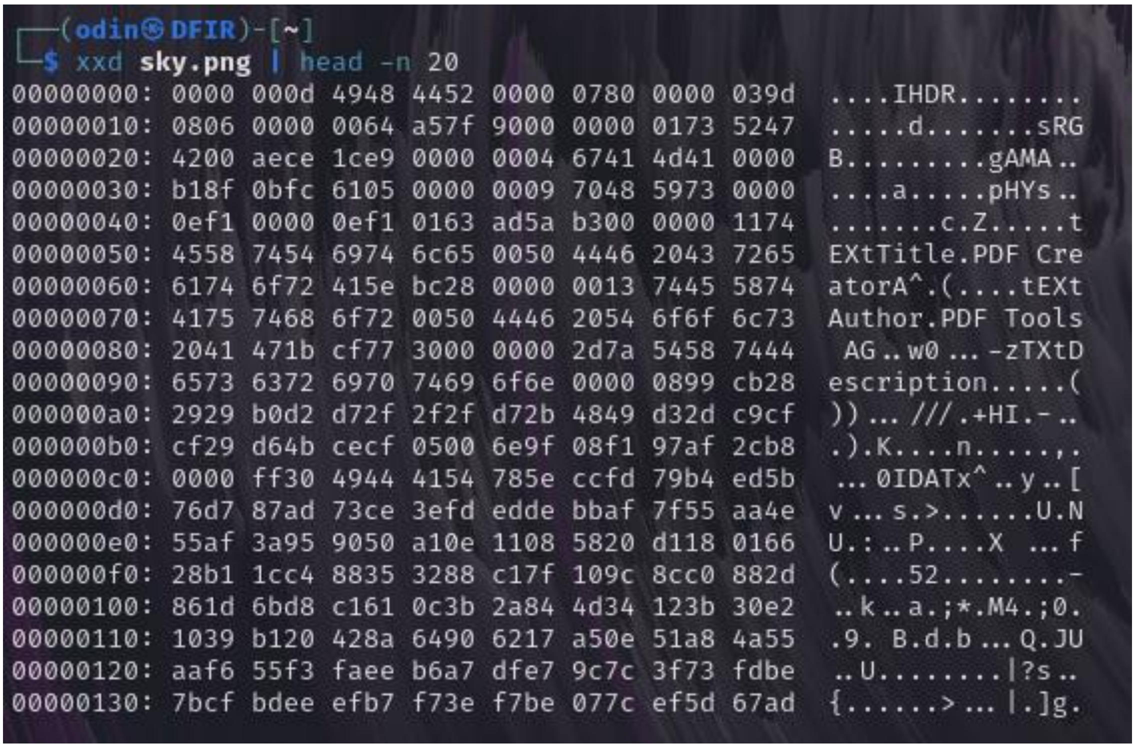
```
tcp
True
False
92.68.200.107
DESKTOP-71OAN8V
61637
-
False
92.68.200.206
-
25
smtp

The message resource is present but the message was not found in the message table
```

팀 이 름 **Team Name**	*Jun9k00k*
문 제 이 름 **Question**	*PNG*
문제 풀이과정 작성 (캡처화면 필수) / Write-up Details (The screenshot is mandatory)	

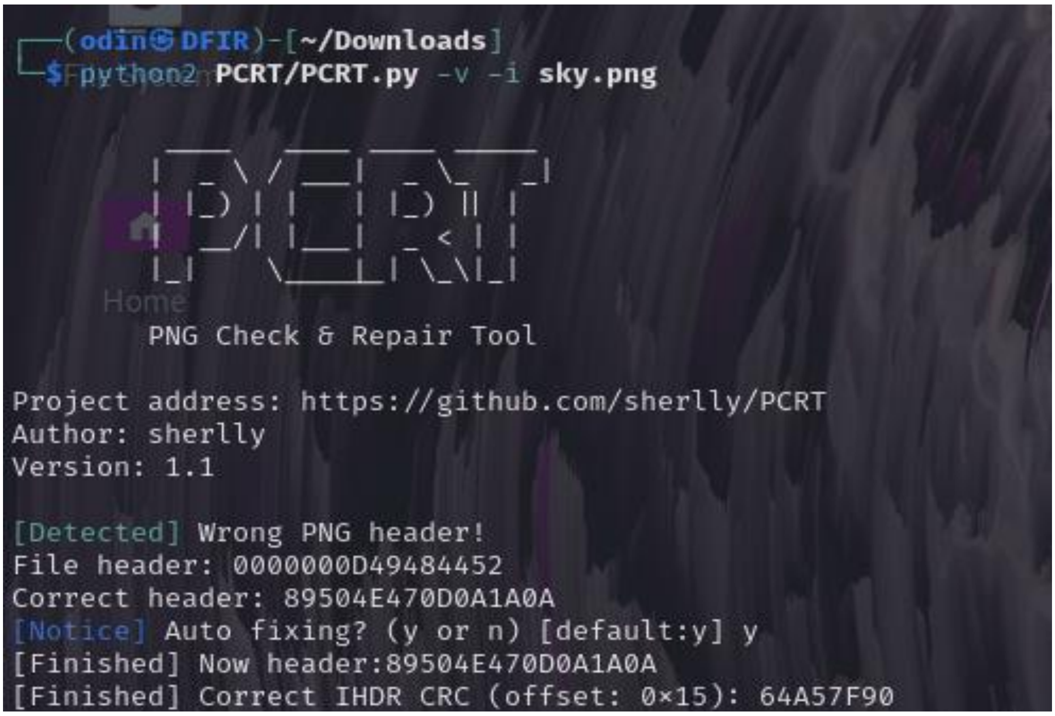
2024 HACKTHEON SEJONG

This challenge gave us a file named **sky.png** and at first, it could not open
After that, I use **xxd** to see hex value inside the image:



```
┌──(odin㉿DFIR)-[~]
└─$ xxd sky.png | head -n 20
00000000: 0000 000d 4948 4452 0000 0780 0000 039d  ....IHDR........
00000010: 0806 0000 0064 a57f 9000 0000 0173 5247  .....d.......sRG
00000020: 4200 aece 1ce9 0000 0004 6741 4d41 0000  B.........gAMA..
00000030: b18f 0bfc 6105 0000 0009 7048 5973 0000  ....a.....pHYs..
00000040: 0ef1 0000 0ef1 0163 ad5a b300 0000 1174  .......c.Z.....t
00000050: 4558 7454 6974 6c65 0050 4446 2043 7265  EXtTitle.PDF Cre
00000060: 6174 6f72 415e bc28 0000 0013 7445 5874  atorA^.(....tEXt
00000070: 4175 7468 6f72 0050 4446 2054 6f6f 6c73  Author.PDF Tools
00000080: 2041 471b cf77 3000 0000 2d7a 5458 7444   AG..w0...-zTXtD
00000090: 6573 6372 6970 7469 6f6e 0000 0899 cb28  escription.....(
000000a0: 2929 b0d2 d72f 2f2f d72b 4849 d32d c9cf  ))...///.+HI.-..
000000b0: cf29 d64b cecf 0500 6e9f 08f1 97af 2cb8  .).K....n.....,.
000000c0: 0000 ff30 4944 4154 785e ccfd 79b4 ed5b  ...0IDATx^..y..[
000000d0: 76d7 87ad 73ce 3efd edde bbaf 7f55 aa4e  v...s.>......U.N
000000e0: 55af 3a95 9050 a10e 1108 5820 d118 0166  U.:..P....X ...f
000000f0: 28b1 1cc4 8835 3288 c17f 109c 8cc0 882d  (....52........-
00000100: 861d 6bd8 c161 0c3b 2a84 4d34 123b 30e2  ..k..a.;*.M4.;0.
00000110: 1039 b120 428a 6490 6217 a50e 51a8 4a55  .9. B.d.b...Q.JU
00000120: aaf6 55f3 faee b6a7 dfe7 9c7c 3f73 fdbe  ..U........|?s..
00000130: 7bcf bdee efb7 f73e f7be 077c ef5d 67ad  {......>...|.]g.
```

You can see that png format is wrong, so we need to fix it so that we can see the picture. I'm a bit lazy in fixing it by hand, so I use a tool name PCRT :



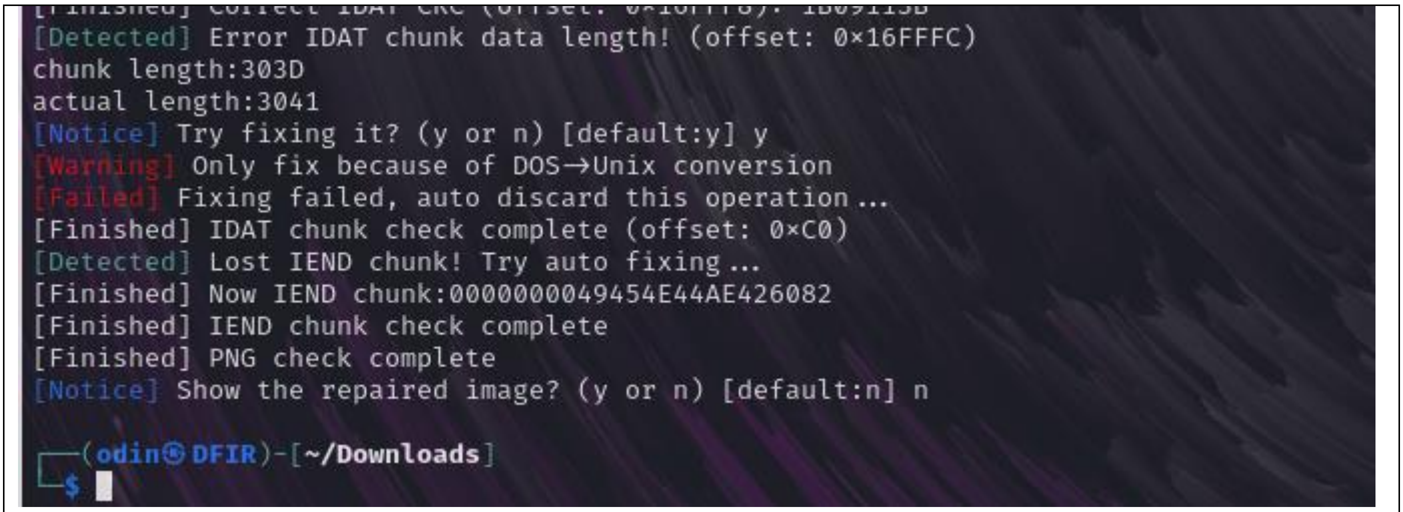
```
┌──(odin㉿DFIR)-[~/Downloads]
└─$ python2 PCRT/PCRT.py -v -i sky.png

         ____   ____ ____ _____
        |  _ \ / ___|  _ \_   _|
        | |_) | |   | |_) || |
        |  __/| |___|  _ < | |
        |_|    \____|_| \_\|_|

        PNG Check & Repair Tool

Project address: https://github.com/sherlly/PCRT
Author: sherlly
Version: 1.1

[Detected] Wrong PNG header!
File header: 0000000D49484452
Correct header: 89504E470D0A1A0A
[Notice] Auto fixing? (y or n) [default:y] y
[Finished] Now header:89504E470D0A1A0A
[Finished] Correct IHDR CRC (offset: 0x15): 64A57F90
```



```
[Finished] Correct IDAT CRC (offset: 0x16FFF8): 1B09113B
[Detected] Error IDAT chunk data length! (offset: 0×16FFFC)
chunk length:303D
actual length:3041
[Notice] Try fixing it? (y or n) [default:y] y
[Warning] Only fix because of DOS→Unix conversion
[Failed] Fixing failed, auto discard this operation ...
[Finished] IDAT chunk check complete (offset: 0×C0)
[Detected] Lost IEND chunk! Try auto fixing ...
[Finished] Now IEND chunk:0000000049454E44AE426082
[Finished] IEND chunk check complete
[Finished] PNG check complete
[Notice] Show the repaired image? (y or n) [default:n] n

┌──(odin㉿DFIR)-[~/Downloads]
└─$ 
```

And the result will be saved in **output.png**, open it and enjoy the result: