CyTaka™
WORLD CYBER
CHAMPIONSHIP

# File.pcap

## Description

You start your descent…
Finding the flag in every layer should be rather easy,
But can you reach the bottom?

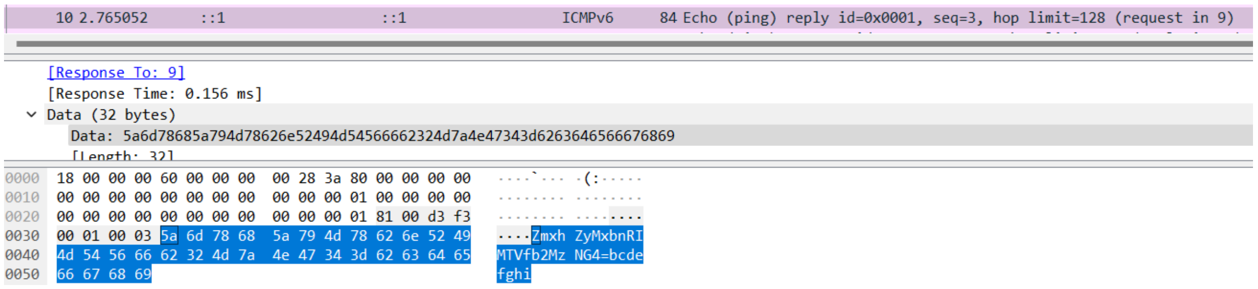
## Recommended Tools

Wireshark
Hex editor [for example HXD]
Disassembler [for example Ida freeware]

## Solutions

### 1 - flag#1n_tH15_oc34n

Open the pcap in Wireshark. One of the ICMP packets has a flag hidden in its data section.



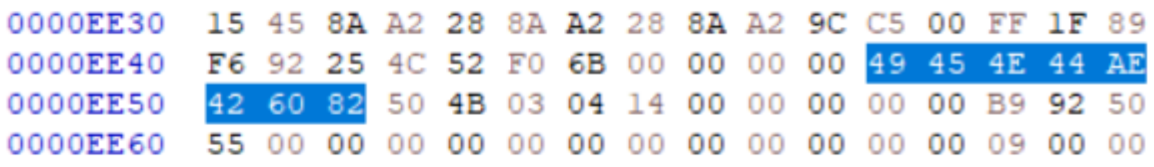
```
10 2.765052     ::1                ::1                ICMPv6     84 Echo (ping) reply id=0x0001, seq=3, hop limit=128 (request in 9)

    [Response To: 9]
    [Response Time: 0.156 ms]
  ⌄ Data (32 bytes)
      Data: 5a6d78685a794d78626e52494d54566662324d7a4e47343d6263646566676869
      [Length: 32]

0000  18 00 00 00 60 00 00 00  00 28 3a 80 00 00 00 00   ····`··· ·(:·····
0010  00 00 00 00 00 00 00 00  00 00 00 01 00 00 00 00   ········ ········
0020  00 00 00 00 00 00 00 00  00 00 00 01 81 00 d3 f3   ········ ········
0030  00 01 00 03 5a 6d 78 68  5a 79 4d 78 62 6e 52 49   ····Zmxh ZyMxbnRI
0040  4d 54 56 66 62 32 4d 7a  4e 47 34 3d 62 63 64 65   MTVfb2Mz NG4=bcde
0050  66 67 68 69                                        fghi
```

we_reign.png has been transmitted through HTTP. Export it and save it with Wireshark.

### 2 - flag#dR0wn1Ng_1n_J34L0U5y

To grab the flag just open the PNG.

There is a file hidden in the PNG. Opening it up with a hex editor reveals that there is a lot of data after the IEND section of the png. Here's the IEND section:



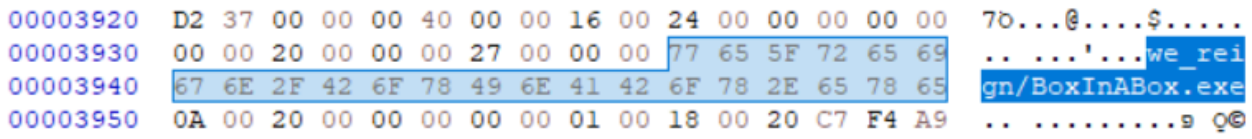
```
0000EE30  15 45 8A A2 28 8A A2 28 8A A2 9C C5 00 FF 1F 89
0000EE40  F6 92 25 4C 52 F0 6B 00 00 00 00 49 45 4E 44 AE
0000EE50  42 60 82 50 4B 03 04 14 00 00 00 00 00 B9 92 50
0000EE60  55 00 00 00 00 00 00 00 00 00 00 00 00 09 00 00
```

Create a new file with all of the content that is after the IEND section.

**3 - flag#MI574King_pRID3**

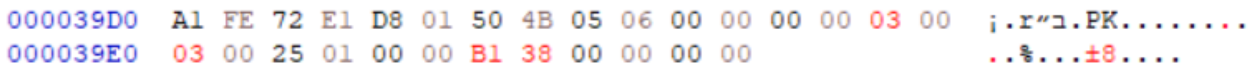
You probably found the flag even without understanding what is this file, because it's plaintext. To proceed you have to take a closer look. By looking at the magic number you can tell that's a zip. But if you'll unzip it you'll only get the text file with the flag.

The zip file itself is way larger than that text file, so it must contain more. By looking at the headers at the end of the file you can see there is one file that is missing from the unzipped directory:



```
00003920  D2 37 00 00 00 40 00 00 16 00 24 00 00 00 00 00   7ὁ...@.....$......
00003930  00 00 20 00 00 00 27 00 00 00 77 65 5F 72 65 69   .. ...'...we_rei
00003940  67 6E 2F 42 6F 78 49 6E 41 42 6F 78 2E 65 78 65   gn/BoxInABox.exe
00003950  0A 00 20 00 00 00 00 00 01 00 18 00 20 C7 F4 A9   .. ......... פ Q©
```

To fix the zip file modify the three headers at the end of it. set 'LocationOfCentralDir' to the first file header and 'ThisDiskItemEntries' and 'DiskItemEntries' to 3:



```
000039D0  A1 FE 72 E1 D8 01 50 4B 05 06 00 00 00 00 03 00   ¡.r״ב.PK........
000039E0  03 00 25 01 00 00 B1 38 00 00 00 00               ..%...±8....
```

Now it unzips properly.

**4 - flag#re4Ch_Th3_5uNLigHt**

The next flag can also be found easily, it is plaintext.

To get to the next level you need to realize this is a UPX-packed executable. download UPX and unpack it (or use any other method of unpacking).

**5 - flag#t3n_th0u5aNd_M3t3r2_Und3r**

Running strings or opening a disassembler reveals a base64 string. Decode it twice and grab the flag.