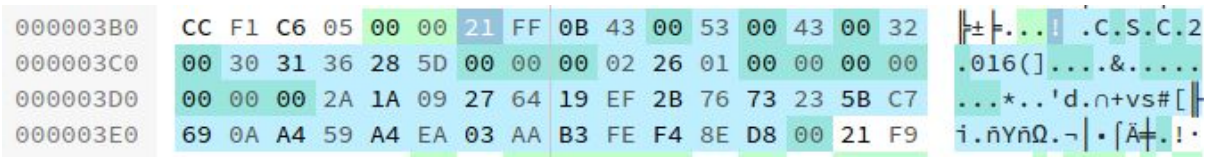
reading_between_the_lines - 300pts

The task title suggests that we should look in a file for additional information, but I did not solve this one until it was already 2 hints in.
The first hint suggested that I should retrace my steps (key.gif) and the second hint was a link to a gif format specification document.

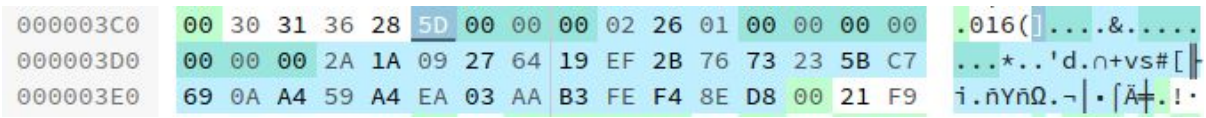
So I opened key.gif in a hex editor to start reading its contents (hexed.it). I noticed that the gif had more pictures(QR codes) in it but they were not relevant for this task. From the gif format reference, an application extension is specified like this:

- header: 0x21 0xff
- size: 0xb
- identifier: 8 bytes
- authentication code: 3 bytes
- block size: 1 byte (n)
- application data: n bytes
- terminator: 0x0

I specifically read about the application extension because in the hex editor I could find fragments of data that contained the CSC2016 bytes:



I extracted all of the application data blocks (we have only 1 for each picture in the gif) using dd.



After all this I was not really sure what to do with all these dumps so I tried putting them all together. They didn't yield any result in the beginning but after running binwalk over the file:

```
~/ctf/csc_2016/forensics/reading_between_the_lines$ binwalk compresed_out

DECIMAL       HEX             DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             LZMA compressed data, properties: 0x5D, dictionary
```

With binwalk -e we can extract the file and tadaaaaa:

```
:~/ctf/csc_2016/forensics/reading_between_the_lines$ cat 0
This task is part of CSC Romania 2016 (http://www.cybersecuritychallenge.ro),
a qualifying event for ECSC 2016 (http://www.europeancybersecuritychallenge.eu/about/2016).
Please do NOT hand over your solution to others as it will spoil the competition.
csc2016{4VSzu2tybWYvnxc2y4uwN4ggaoGK1SMi}
```