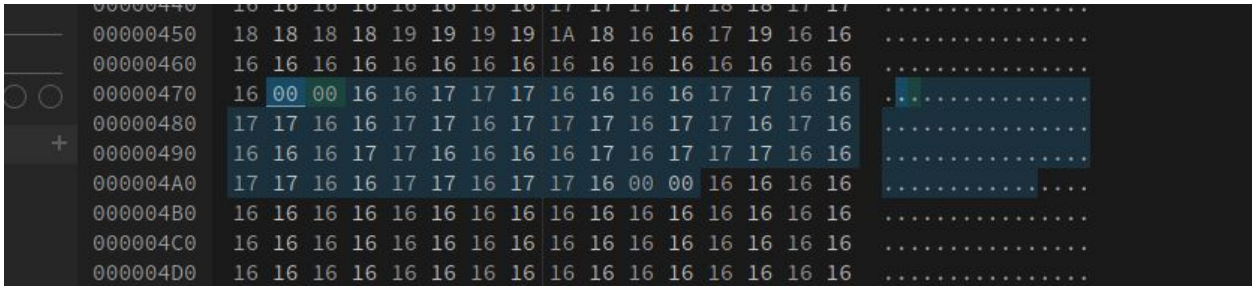
Flags

1. 837has6
2. jb298abc9qb2
3. n38f298hsjf
4. p68cq1hb
5. k8x0w5bbuq
6. nothere

No code was used to get the flags

Task 1

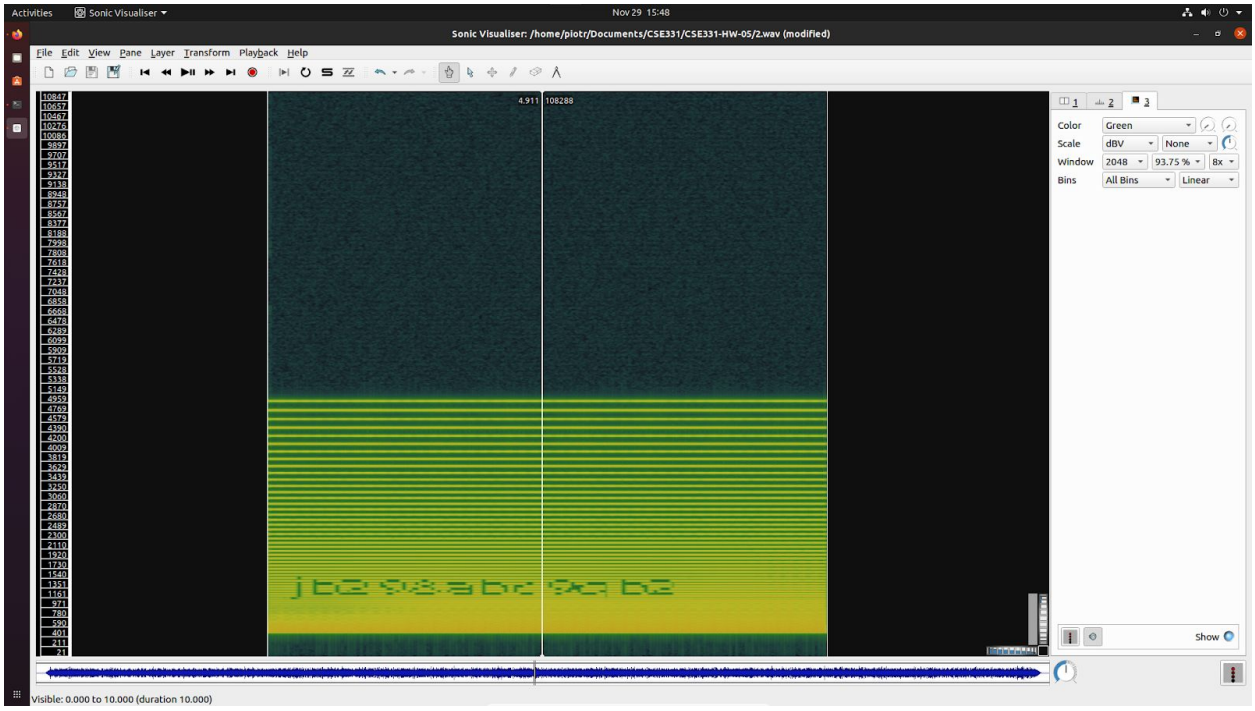
The hint for this task recommended looking for two null bytes so I decided to open up the bmp file in a hex editor (specifically https://hexed.it/) and I then used its search feature to look for two consecutive null bytes. There were a lot at the beginning but I'm pretty sure that's data to describe the file so I looked further and I found this,



a segment that is wrapped in 2 null bytes at each side. I was a bit confused with what to do at first but eventually it clicked that this might be binary so I translated the 16 and 17 into 0 and 1 respectively and then used this https://www.rapidtables.com/convert/number/binary-to-ascii.html binary to ascii converter. The result was 8fnÐÂæ which didn't make sense until I noticed that in the binary code which was "00111000 01100110 01101110 11010000 11000010 11100110 0110110" the last section was missing a bit. I then went around adding bits to see if an added bit somewhere would lead to a coherent outcome and luckily on my second try I added a 0 to the second run of 0s and got the flag 837has6.

Task 2

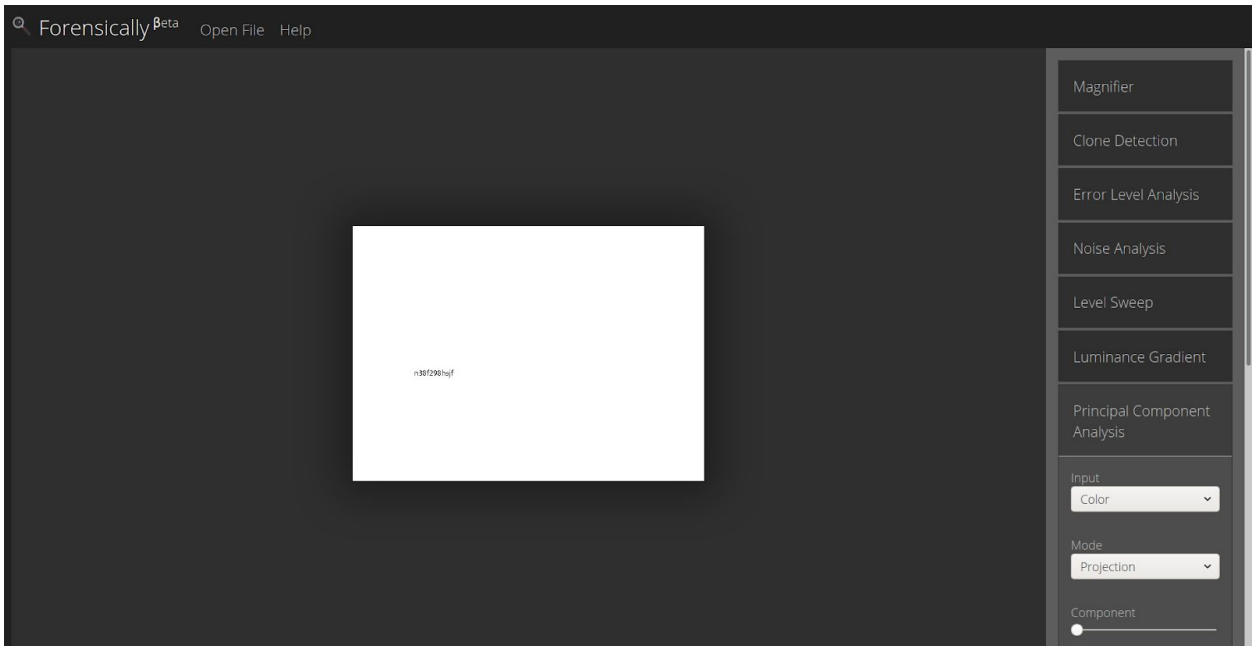
The hint for this task asked us to visualize sound and linked us to sonic visualizer. I uploaded the wav file to it and it didn't look like much at first so I started adding different panes to see which visualization would give me something and the spectrogram pane gave me this,



which is a bit hard to read but I played around with the settings on the left and I'm pretty sure it says jb298abc9qb2 which I assume is the flag.

Task 3

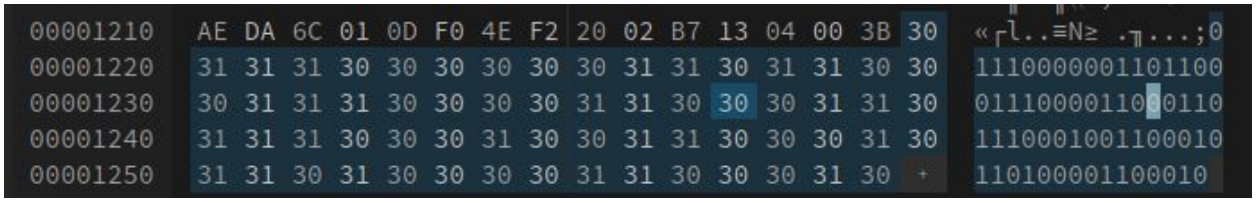
For this task the hint was to look at the pixels more carefully so the first thing I did was open up the image and zoom in but I couldn't find anything. I decided to look for tools to see if anything could let me do analysis on the pixels and I managed to find this tool: https://29a.ch/photo-forensics/#forensic-magnifier which had a bunch of cool analysis tools. I basically clicked through each of them to see if anything jumped out at me and thankfully it did on the noise analysis section. It was a bit hard to read though so I continued checking each analysis tool and in the principle component analysis one I got a clear view of the flag



which is n38f298hsjf.

Task 4

The hint for this task is that the password is in the end so I first tried to get the gif to be like a video so that I could just navigate to the last frame but upon doing this I found out that the gif is only a single frame. I then used the same hex editor as in task 1 and I looked at the end and I found interesting binary code at the end of the file which looked like this:

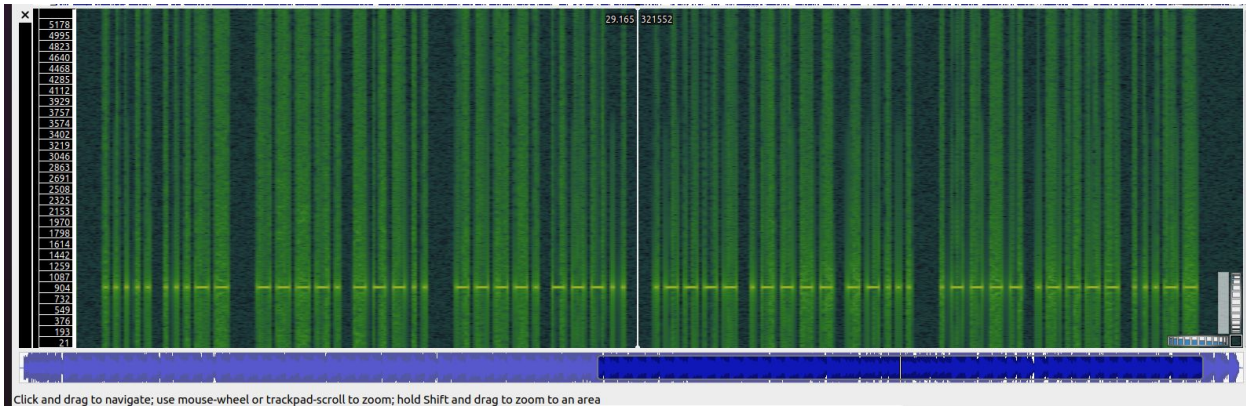


I copied the binary code on the left into this binary code to text translator https://www.rapidtables.com/convert/number/binary-to-ascii.html and I got p68cq1hb which looks like the flag.

Task 5

I started this task by listening to it and I spent the next 10 minutes jamming out. I didn't get anything from jamming out so I decided to put it into the sonic visualizer that we used for

task 2 and when I clicked on the spectrogram again I found that upon zooming in it looked a lot like morse code which would make sense since the hint is morse code. Here is what I saw,



I then translated this into morse code and got .---- ----- --... | ..... -.... | .---- ..--- ----- | ....- ---.. | .---- .---- ----. | ....-- | ----. ---.. | ----. ---.. | .---- .---- --... | .---- .---- ...-- which I then put into a morse code translator (specifically this one https://morsecode.world/international/translator.html) and I got this sequence of numbers: 107 56 120 48 119 53 98 98 117 113. I then put that sequence of numbers into this decimal to text translator https://cryptii.com/pipes/decimal-text and got the text k8x0w5bbuq which is the flag.

Task 6

The hint for this task said that bacon is delicious unless it's a cipher so I looked up if a bacon cipher was a thing and it was so I guessed that was what we are dealing with. I then looked at the jpg and was a bit confused until I realized that the bold and non-bold text created a pattern similar to bacon cipher so I translated it into a series of As and Bs where As are non-bold characters and Bs are bold characters. I then put that sequence into this bacon cipher decoder https://www.dcode.fr/bacon-cipher and it took me a while to get the right sequence cause it was so hard to distinguish some characters from being bold and not being bold. The correct sequence that I got was BAABB AABBB AABAA ABBBB AAAAA BAABA BAABA BABBA ABBBA BAAAB AAABB ABAAA BAABA ABBAB ABBBA BAABB AABBB AABAA BAAAB AABAA which translated to THEPASSWORDISNOTHERE so I assume the flag is nothere.