**brokenimg**

***Forensic***

**Challenge:**

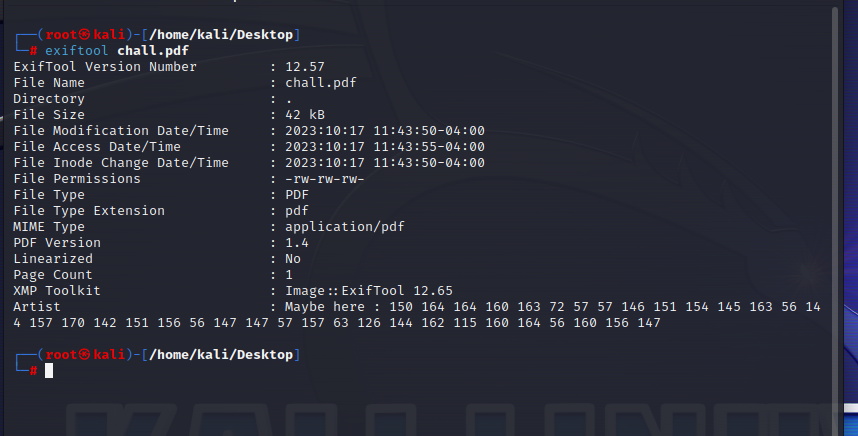
why the picture like this

Attached - chall.pdf

**Approach:**

First of all, the challenge talks about an image, but a PDF file is given?? Hmmm, let’s analyze this file. The PDF file in itself is filled with text related to a very deep love story and the hardships a picture had to face lol (did not made me cry there) I tried strings on the PDF file when a lot of stuff came up, which hinted me towards using the exiftool (the data can also be retrieved from strings but for this writeup purpose i’ll show it using exiftool)

exiftool chall.pdf

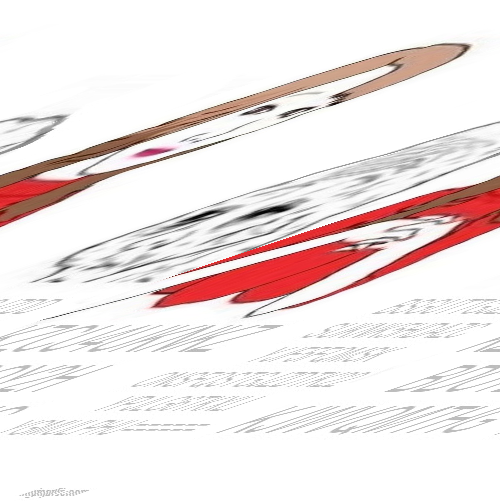


We get this array of numbers that does not make sense like this:

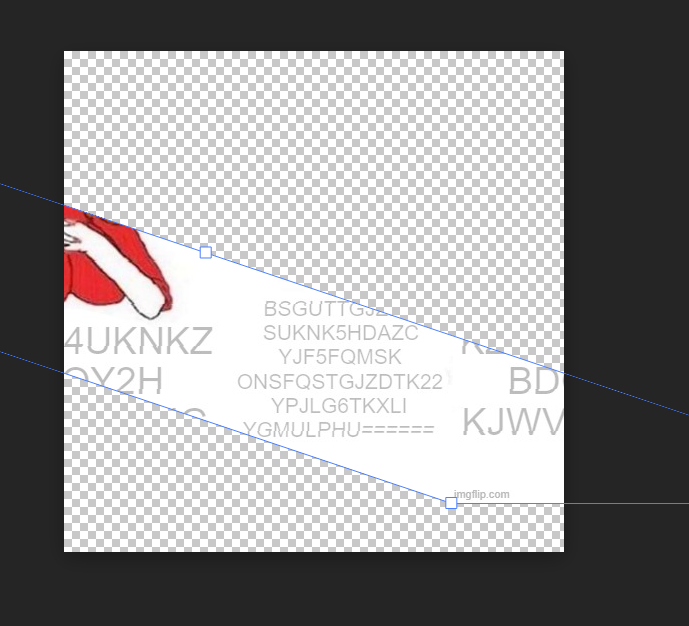
[150 164 164 160 163 72 57 57 146 151 154 145 163 56 144 157 170 142 151 156 56 147 147 57 157 63 126 144 162 115 160 164 56 160 156 147]

We enter this on cyberchef and realize it’s in the octal system and translates to a files.doxbin URL. The files.doxbin had a PNG file in which we downloaded. (Finally an image!!)

<https://files.doxbin.gg/o3VdrMpt.png>



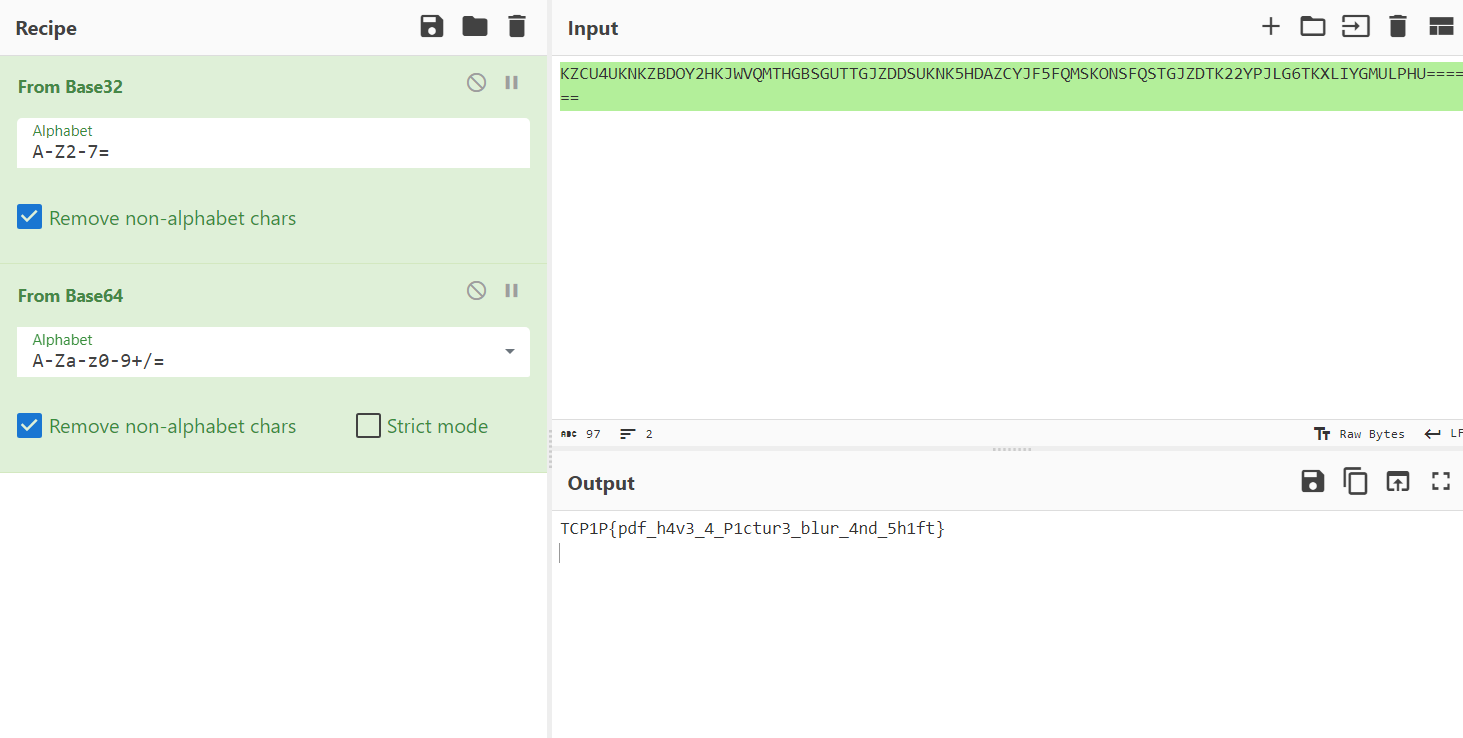
This image on one look, looks like it’s been skewed to its death, so we’re gonna unskew it. I’m gonna use a online photo editing tool named photopea and here is what i got after skewing it:



We notice there are 2 chunks of encrypted text present in the image which is trimmed a bit towards the corners, but the same trimmed text is then displayed on the other corner of the image. After a bit of struggle and a lot of coffee we were able to get both of those texts and combined it into a single one which read:

KZCU4UKNKZBDOY2HKJWVQMTHGBSGUTTGJZDDSUKNK5HDAZCYJF5FQMSKONSFQSTGJZDTK22YPJLG6TKXLIYGMULPHU======

Upon checking it in Cyberchef, we got to know the text was encrypted using Base64 first and then Base32, so we decode it and get this:



Hence our flag!

**Flag: TCP1P{pdf\_h4v3\_4\_P1ctur3\_blur\_4nd\_5h1ft}**

Congrats!!

Happy Hacking!