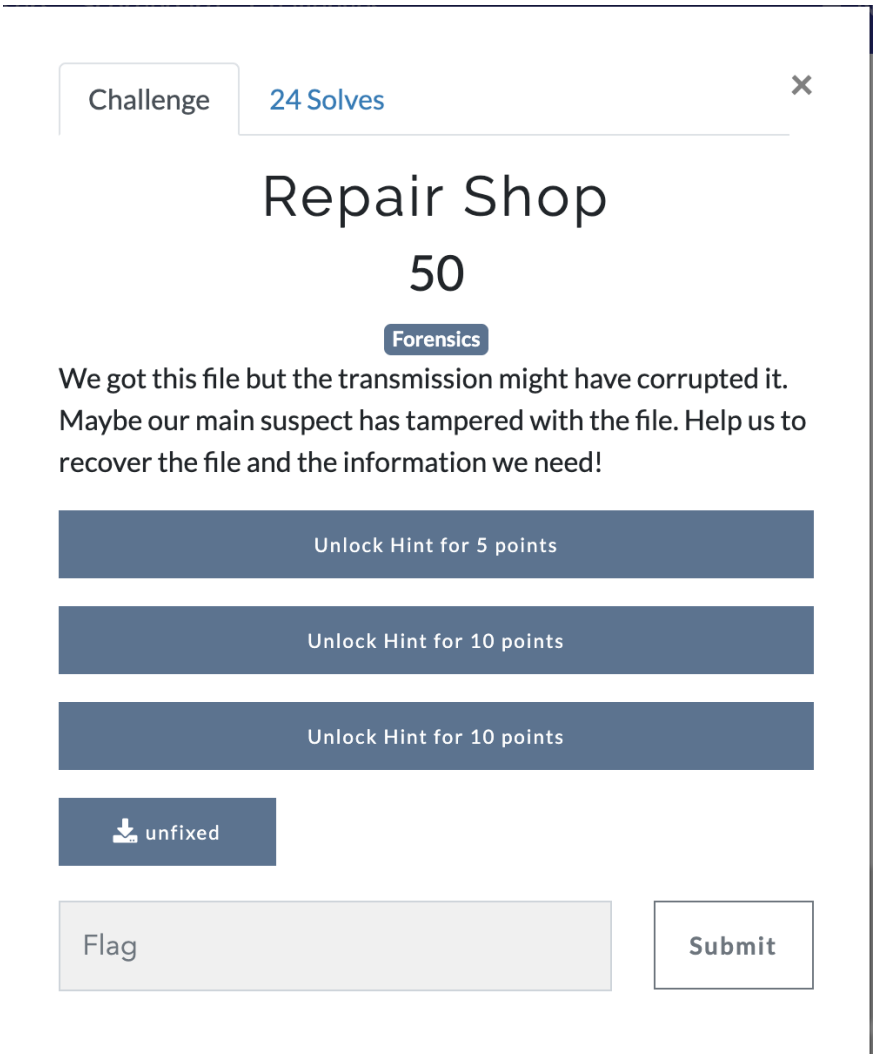
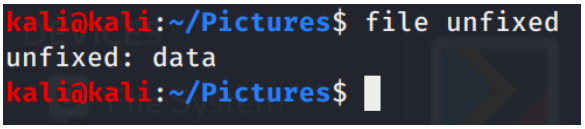
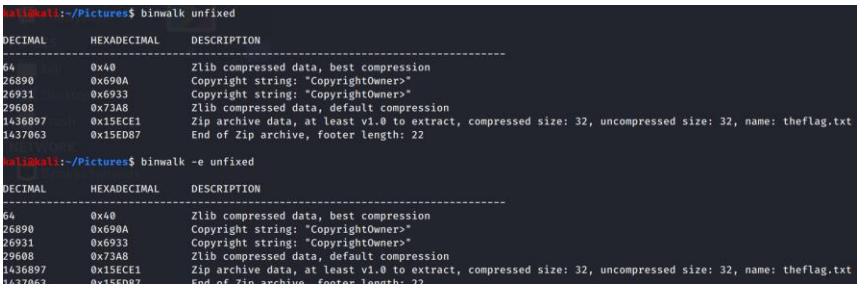
Forensics Challenge 1



Answer: We are given a file named unfixed, looks be just raw data



Let's use binwalk to see if there is anything inside of it. Looks to be a txt file called theflag.txt lets extract everything



```
kali@kali:~/Pictures$ binwalk unfixed

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
64            0x40            Zlib compressed data, best compression
26890         0x690A          Copyright string: "CopyrightOwner>"
26931         0x6933          Copyright string: "CopyrightOwner>"
29608         0x73A8          Zlib compressed data, default compression
1436897       0x15ECE1        Zip archive data, at least v1.0 to extract, compressed size: 32, uncompressed size: 32, name: theflag.txt
1437063       0x15ED87        End of Zip archive, footer length: 22

kali@kali:~/Pictures$ binwalk -e unfixed

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
64            0x40            Zlib compressed data, best compression
26890         0x690A          Copyright string: "CopyrightOwner>"
26931         0x6933          Copyright string: "CopyrightOwner>"
29608         0x73A8          Zlib compressed data, default compression
1436897       0x15ECE1        Zip archive data, at least v1.0 to extract, compressed size: 32, uncompressed size: 32, name: theflag.txt
1437063       0x15ED87        End of Zip archive, footer length: 22
```

Cat out theflag.txt to find a base64 string and decode the string to get a false flag.

```
kali@kali:~/Pictures/_unfixed-1.extracted$ cat theflag.txt
aGlsbHRvcE1TR3trZWVwX2xvb2tpbmd9kali@kali:~/Pictures/_unfixed-1.extracted$ cat theflag.txt | base64 -d
hilltopMSG{keep_looking}kali@kali:~/Pictures/_unfixed-1.extracted$
```

False Flag: hilltopMSG{keep_looking}

Ok Hilltop lets keep looking, what other files did we get.

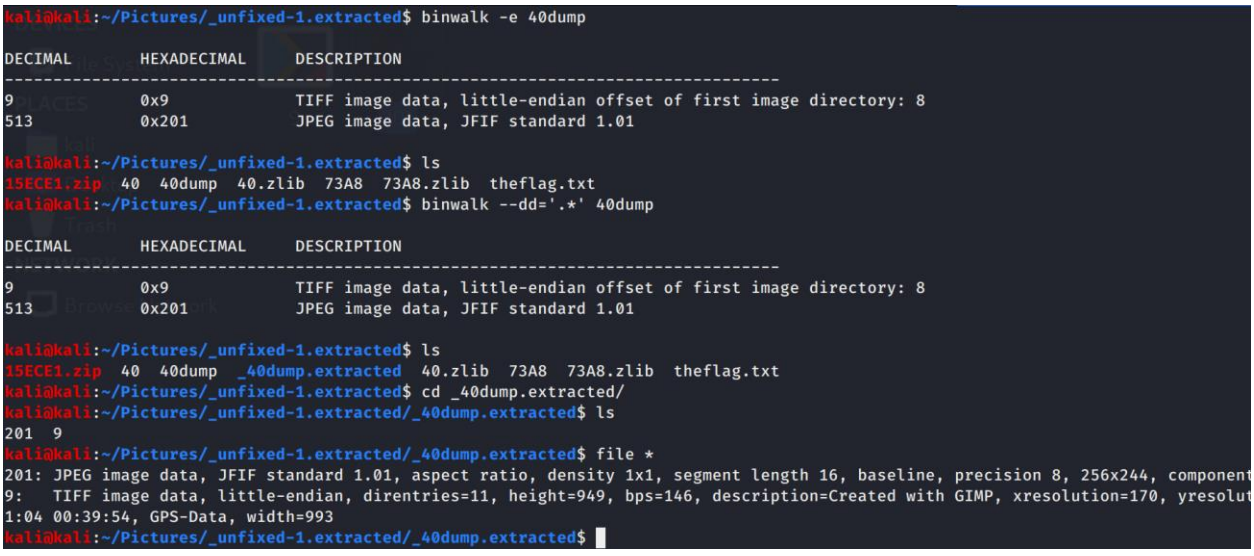
```
kali@kali:~/Pictures/_unfixed-1.extracted$ file *
15ECE1.zip:  Zip archive data, at least v?[0x30a] to extract
40:          ASCII text
40.zlib:     zlib compressed data
73A8:        empty
73A8.zlib:   zlib compressed data
theflag.txt: ASCII text, with no line terminators
kali@kali:~/Pictures/_unfixed-1.extracted$
```

Another ascii file, 40. Cat the file looks to be a hex dump, let's decode and store the results.

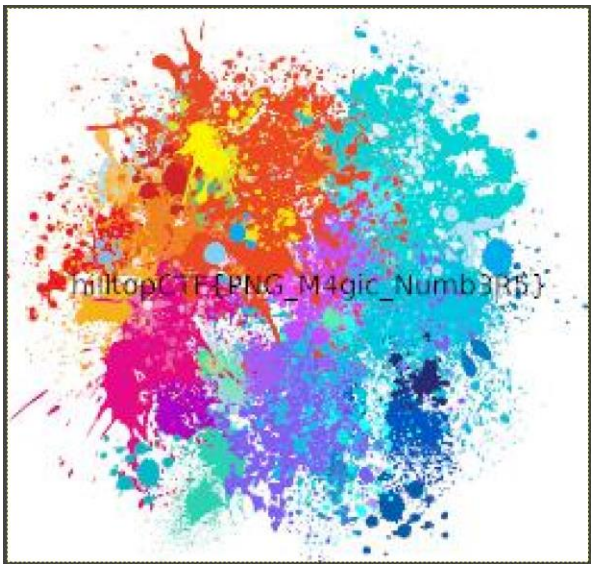
```
381d480718cf38c7a0c55abed4743b8b97d1daf209e5c8f36d7979188c6171f873e9edd6
a5d0bc39a6f84ac2e661e521dd24d34e14a8099271824e028e3f0f7ae2c4627dac55d34f
a2ee7a187c3fb26f95dd7e46c436a15d259163f3514a2b46a540538246327b8ab3819ce3
9aa5a56ab67ad69d15fd84a64b6973b1ca95ce090783cf506aed79d2ba76677269aba0a2
8a290c28a28a0028a28a0028a28a0028a28a0028a28a00298ea0e0b1202f3d7f9d3e9323
27d7140104abb54b1760339fbd8c0eff00e7f9561cbbaea46748e691cab060141c292a3a
31c763f5e3ad6f9059b86ed9c85fcaa0783e56c2839c903279f5e29a67356a5cfa743234
fd0a2b44692191c5d4a09f3028050139c60e45685a598b7999d242013c9231b873dbf11e
9d3e956d2290290c54b13f794638e83f4c53843f3abee3b80c63b7e54377153c3429a4a2
ad60442b3121c00c3ee63bfafebfcbf191115176a28551d853b14523a52b1ccf8c34dd43
5286ce1b154daf288e762818aa12a73cff00082a091df02b456c96d64b58a269d22b688e
23455db30c018638e0f43d467f038d5a3ad6dede5c8a1d15ff001199d3cb0b496a5dcb2c
8db802c50803e6071d700803e879cd433693a7cf72975259c53481b7abaae18b13c13eb8
07bfe1562eec127b9f3992276f29a24df16fc6eeb9e7a102992584924f6e639d6286d5c9
10226030db8553e98ce78a719256b3b7f5fd2327abd8cad5bc2d06ababc77d3cf27c91f9
7e4a1c0cf27920838209e073cf5aa0fa6697e20f09dff87f43ba36ee98de763ae189dd86
cf241c11dce3d6bb1488189564018839f6cfb564f887c45a6f856da2babd8e4db3ca23cc
29939c1393ed81551ab52568abb6b6ff00863274a0bde924975efafe4637863e1fdb693a
0dc699ab1835059dcbe0c7811e400769ea0f03918e83d2b8ef10ea1e1bf036ae57c3d611
cbaa236d93ce690adb100f2b9c12583e3838c28e2bd8274866b678e700c522ec60dc020f
18fd6a9dde85a5df5a476d756104f1478d8b2aeec63007279ed554f12f9dcaa5da7bad82
a61d7228d3b26b67b9c7f83f4281b45b7f14c162d71aedc46cdbeeee4f2492a5810b8191
9238ef8cf535bde1d875b4d2ae63f155d595d1232db40c2291f32bf01718c76ee7935d04
51470c4b1448a91a0c2aa8c003d00a1e249119191595c61c1190c318e6b2a959cdbbf5fc
3c91a42928256ff87f32ae95269d269b09d24db9b20088fecf8d83079c638eb9abb55eca
c6d74db54b5b2b78e08133b638d70064e4d58aca566dd8d637b6a1451452185145140051
4514005145140051451400514514005046 68a28022627ce09d88cd483bd1450242d14514
0c28a28a0028a28a0085e21e6c454b28525b6a9c024fafaf53f8d52d1ee64ba1765f811d
ccb1a8049e03b0cf24ff0087a014515a2f81ff005d40d3a6bc692001d1580391919c1a28
acc06cf6f0dcc5e55c431cb1e41db2286190720e0fbd4945140051451400514514005145
1400514514005145140 1ffd9
kali@kali:~/Pictures/_unfixed-1.extracted$ cat 40 | xxd -r -p > 40dump
kali@kali:~/Pictures/_unfixed-1.extracted$ file 40dump
40dump: data
kali@kali:~/Pictures/_unfixed-1.extracted$
```

Interesting another data file. If we cat the 40dump we get nothing really useful so let try binwalk again.

A JPEG file, lets extract it and open it.



So using the normal -e option with binwalk failed, this can happen, so let’s use dd instead. Now we have the JPEG lets open it up.



Looks like we have our flag

FLAG: hilltopCTF{PNG_M4gic_Numb3R5}