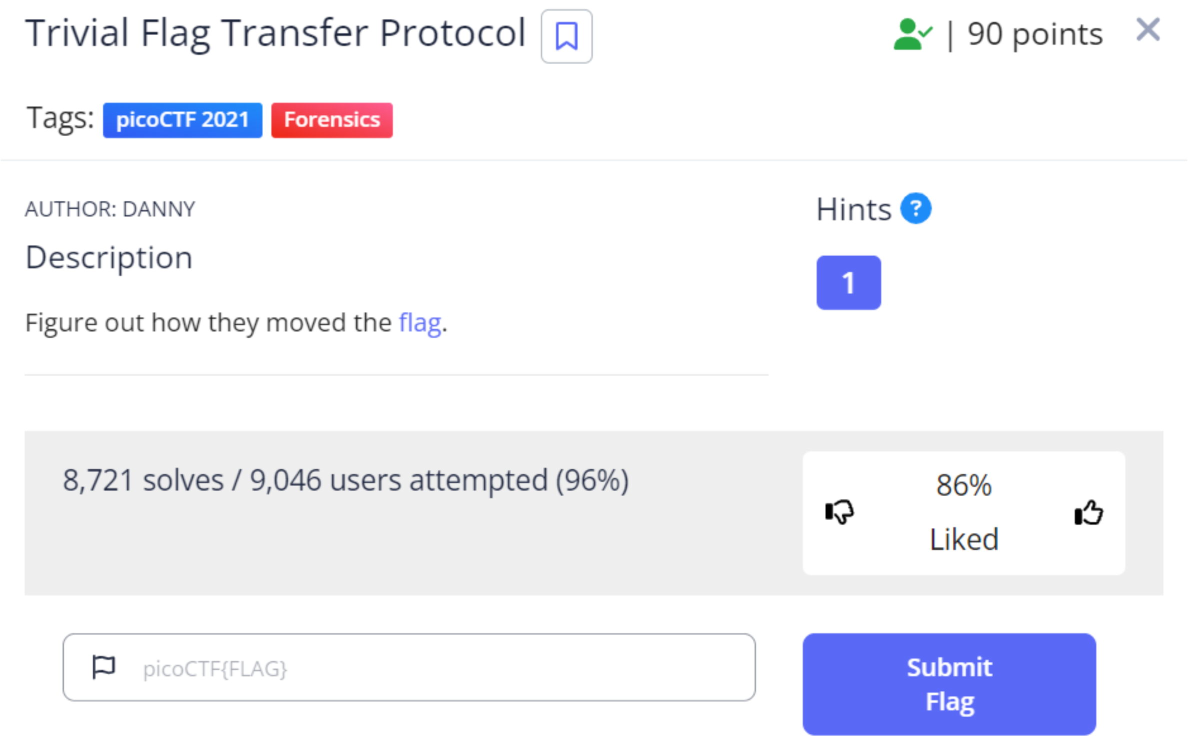
Trivial FTP:

The question is as follows:



Trivial Flag Transfer Protocol

90 points

Tags: picoCTF 2021 Forensics

AUTHOR: DANNY

Description

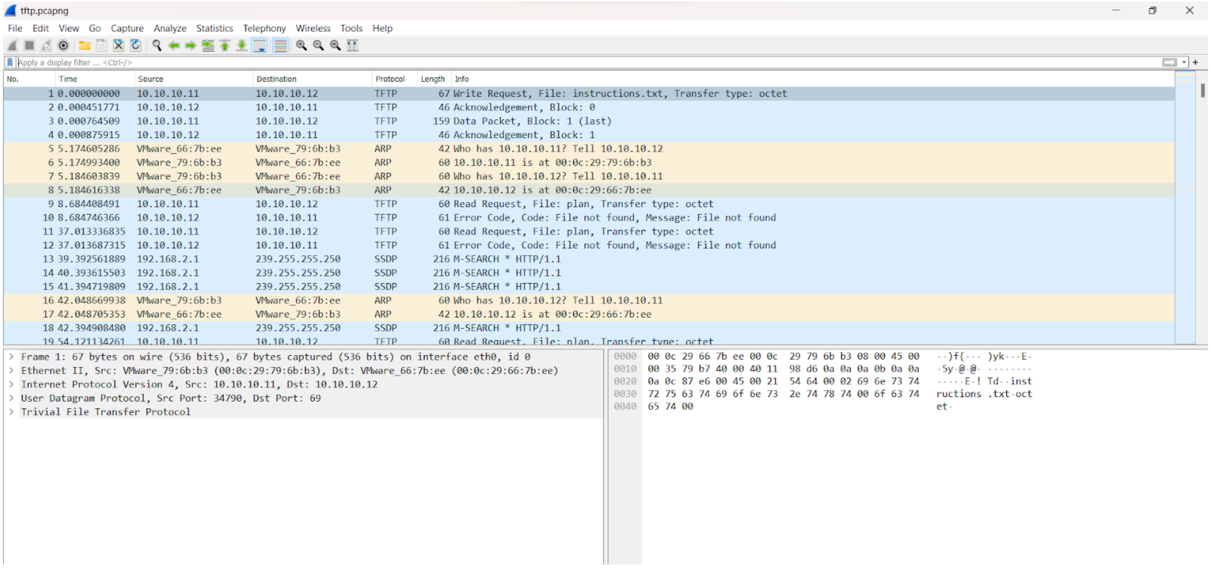
Figure out how they moved the flag.

8,721 solves / 9,046 users attempted (96%)

86% Liked

A wireshark capture file is downloaded.

I opened this using wireshark (a packet sniffing software)


tftp.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

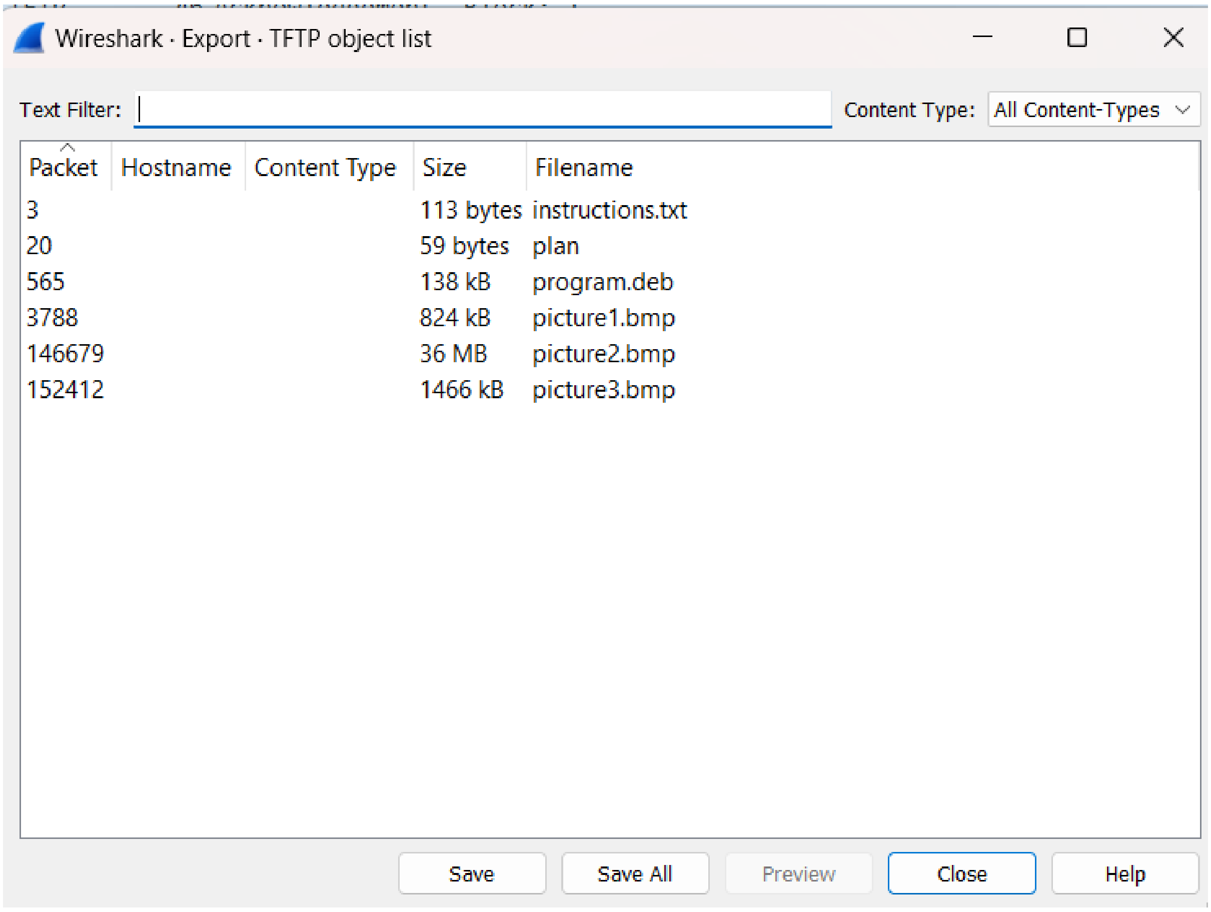
```
No.   Time          Source            Destination       Protocol  Length Info
    1 0.000000000   10.10.10.11       10.10.10.12       TFTP        67 Write Request, File: instructions.txt, Transfer type: octet
    2 0.000451771   10.10.10.12       10.10.10.11       TFTP        46 Acknowledgement, Block: 0
    3 0.000764509   10.10.10.11       10.10.10.12       TFTP       159 Data Packet, Block: 1 (last)
    4 0.000875915   10.10.10.12       10.10.10.11       TFTP        46 Acknowledgement, Block: 1
    5 5.174605286   VMware_66:7b:ee   VMware_79:6b:b3   ARP         42 Who has 10.10.10.11? Tell 10.10.10.12
    6 5.174993400   VMware_79:6b:b3   VMware_66:7b:ee   ARP         60 10.10.10.11 is at 00:0c:29:79:6b:b3
    7 5.184603839   VMware_79:6b:b3   VMware_66:7b:ee   ARP         60 Who has 10.10.10.12? Tell 10.10.10.11
    8 5.184616338   VMware_66:7b:ee   VMware_79:6b:b3   ARP         42 10.10.10.12 is at 00:0c:29:66:7b:ee
    9 8.684408491   10.10.10.11       10.10.10.12       TFTP        60 Read Request, File: plan, Transfer type: octet
   10 8.684746366   10.10.10.12       10.10.10.11       TFTP        61 Error Code, Code: File not found, Message: File not found
   11 37.013336835  10.10.10.11       10.10.10.12       TFTP        60 Read Request, File: plan, Transfer type: octet
   12 37.013687315  10.10.10.12       10.10.10.11       TFTP        61 Error Code, Code: File not found, Message: File not found
   13 39.392561889  192.168.2.1       239.255.255.250   SSDP       216 M-SEARCH * HTTP/1.1
   14 40.393615503  192.168.2.1       239.255.255.250   SSDP       216 M-SEARCH * HTTP/1.1
   15 41.394719809  192.168.2.1       239.255.255.250   SSDP       216 M-SEARCH * HTTP/1.1
   16 42.048669938  VMware_79:6b:b3   VMware_66:7b:ee   ARP         60 Who has 10.10.10.12? Tell 10.10.10.11
   17 42.048705353  VMware_66:7b:ee   VMware_79:6b:b3   ARP         42 10.10.10.12 is at 00:0c:29:66:7b:ee
   18 42.394908480  192.168.2.1       239.255.255.250   SSDP       216 M-SEARCH * HTTP/1.1
   19 54.121134261  10.10.10.11       10.10.10.12       TFTP        60 Read Request, File: plan, Transfer type: octet
```

```
> Frame 1: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface eth0, id 0
> Ethernet II, Src: VMware_79:6b:b3 (00:0c:29:79:6b:b3), Dst: VMware_66:7b:ee (00:0c:29:66:7b:ee)
> Internet Protocol Version 4, Src: 10.10.10.11, Dst: 10.10.10.12
> User Datagram Protocol, Src Port: 34790, Dst Port: 69
> Trivial File Transfer Protocol
```

```
0000  00 0c 29 66 7b ee 00 0c  29 79 6b b3 08 00 45 00   ··)f{··· )yk···E·
0010  00 35 79 b7 40 00 40 11  98 d6 0a 0a 0a 0b 0a 0a   ·5y·@·@· ········
0020  0a 0c 87 e6 00 45 00 21  54 64 00 02 69 6e 73 74   ·····E·! Td··inst
0030  72 75 63 74 69 6f 6e 73  2e 74 78 74 00 6f 63 74   ructions .txt·oct
0040  65 74 00                                           et·
```

There are many tftp protocols here. tftp stands for trivial file transfer protocol. Its similar to http(using tcp to communicate information with other pages,browsers etc.) and other transfer protocols.

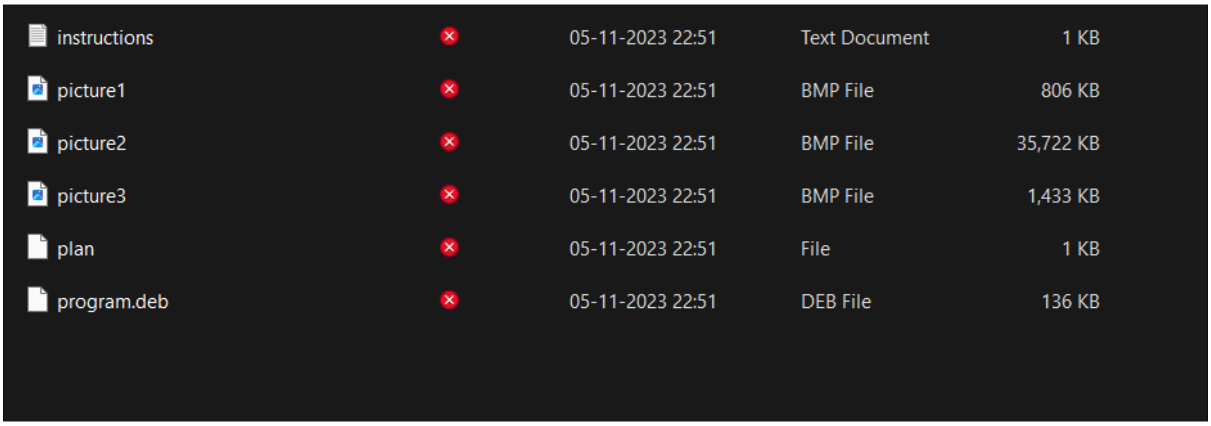
Wireshark can export tftp objects, enabling us to store the actual files being transferred. Since the flag is the file being transferred as per the question, we export the objects


Wireshark · Export · TFTP object list
Text Filter:
Content Type: All Content-Types
Save
Save All
Preview
Close
Help

Packet	Hostname	Content Type	Size	Filename
3			113 bytes	instructions.txt
20			59 bytes	plan
565			138 kB	program.deb
3788			824 kB	picture1.bmp
146679			36 MB	picture2.bmp
152412			1466 kB	picture3.bmp

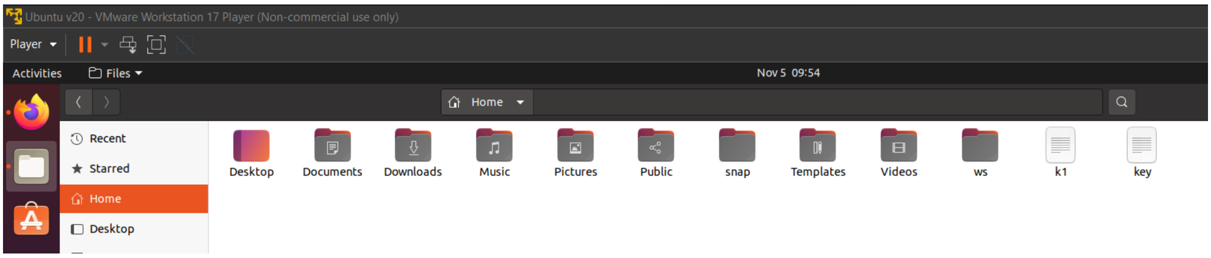
I stored these in a file 'ws'

Viewing contents,

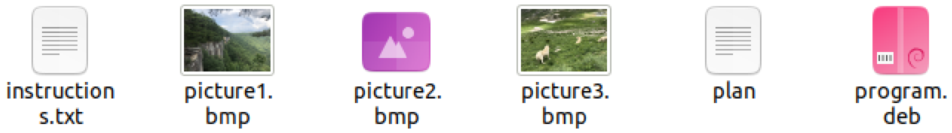


instructions	⊗	05-11-2023 22:51	Text Document	1 KB
picture1	⊗	05-11-2023 22:51	BMP File	806 KB
picture2	⊗	05-11-2023 22:51	BMP File	35,722 KB
picture3	⊗	05-11-2023 22:51	BMP File	1,433 KB
plan	⊗	05-11-2023 22:51	File	1 KB
program.deb	⊗	05-11-2023 22:51	DEB File	136 KB

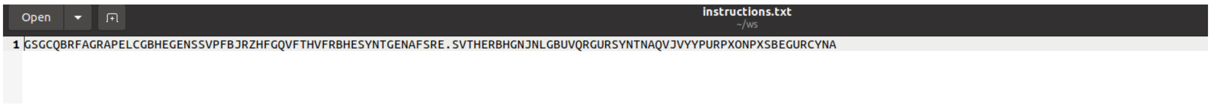
Note that this contains a .deb file, meaning we have to use a linux environment. So I copied the same folder onto a VM.



Following are the contents:

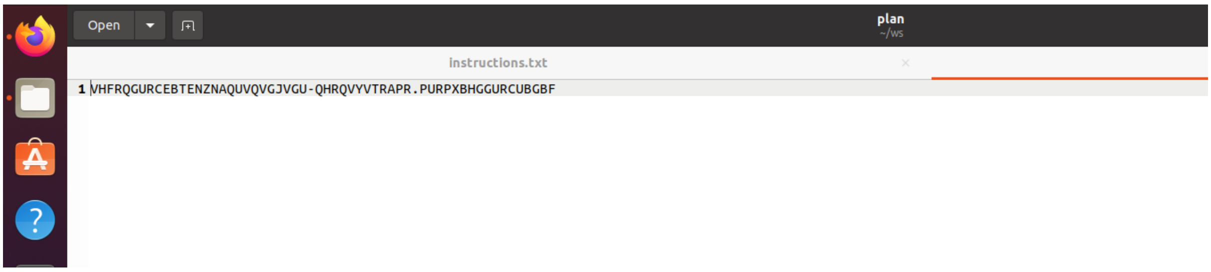


instructions.txt:



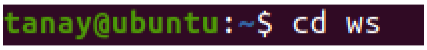
```
GSGCQBRFAGRAPELCGBHEGENSSVPFBJRZHFGQVFTHVFRBHESYNTGENAFSRE.SVTHERBHGNJNLGBUVQRGURSYNTNAQVJVYYPURPXONPXSBEGURCYNA
```

plan:



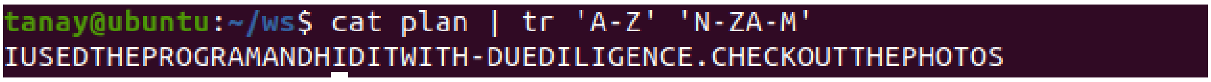
```
VHFRQGURCEBTENZNAQUVQVGJVGU-QHRQVYVTRAPR.PURPXBHGGURCUBGBF
```

I changed my cwd to ws as that is where all the files are.

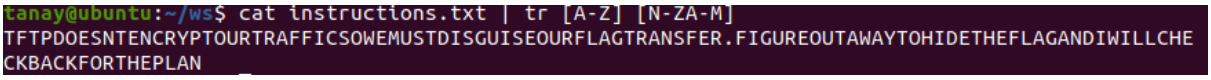


```
tanay@ubuntu:~$ cd ws
```

Both of these are ciphered in ROT13 at first glance. So I ran a check and:

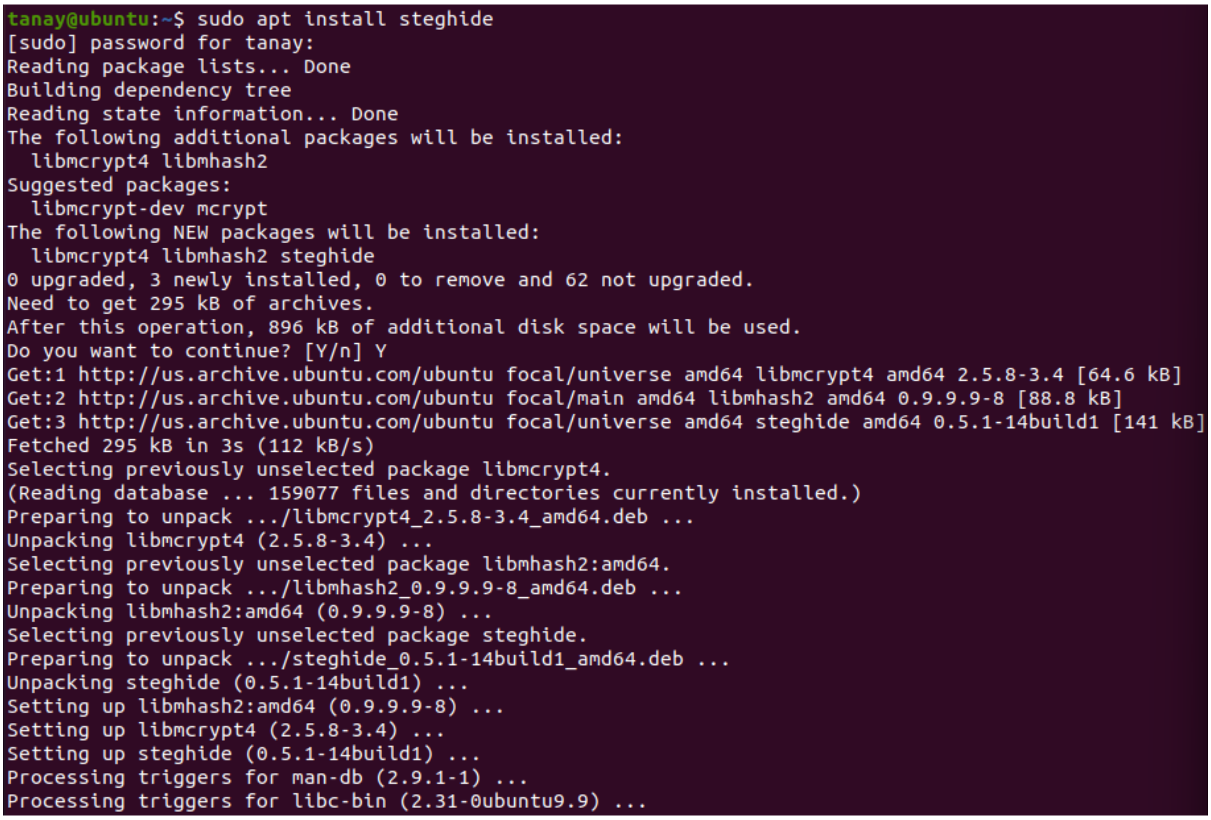


```
tanay@ubuntu:~/ws$ cat plan | tr 'A-Z' 'N-ZA-M'
IUSEDTHEPROGRAMANDHIDITWITH-DUEDILIGENCE.CHECKOUTTHEPHOTOS
```



```
tanay@ubuntu:~/ws$ cat instructions.txt | tr [A-Z] [N-ZA-M]
TFTPDOESNTENCRYPTOURTRAFFICSOWEMUSTDISGUISEOURFLAGTRANSFER.FIGUREOUTAWAYTOHIDETHEFLAGANDIWILLCHE
CKBACKFORTHEPLAN
```

The program.deb file was to install the steghide package, but since the VM could not install it from the file, I installed it in the terminal instead.

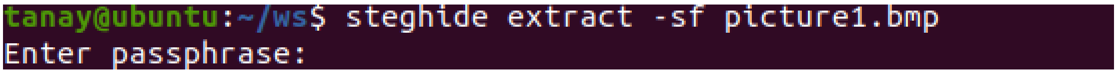


```
tanay@ubuntu:~$ sudo apt install steghide
[sudo] password for tanay:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libmcrypt4 libmhash2
Suggested packages:
  libmcrypt-dev mcrypt
The following NEW packages will be installed:
  libmcrypt4 libmhash2 steghide
0 upgraded, 3 newly installed, 0 to remove and 62 not upgraded.
Need to get 295 kB of archives.
After this operation, 896 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 libmcrypt4 amd64 2.5.8-3.4 [64.6 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libmhash2 amd64 0.9.9.9-8 [88.8 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 steghide amd64 0.5.1-14build1 [141 kB]
Fetched 295 kB in 3s (112 kB/s)
Selecting previously unselected package libmcrypt4.
(Reading database ... 159077 files and directories currently installed.)
Preparing to unpack .../libmcrypt4_2.5.8-3.4_amd64.deb ...
Unpacking libmcrypt4 (2.5.8-3.4) ...
Selecting previously unselected package libmhash2:amd64.
Preparing to unpack .../libmhash2_0.9.9.9-8_amd64.deb ...
Unpacking libmhash2:amd64 (0.9.9.9-8) ...
Selecting previously unselected package steghide.
Preparing to unpack .../steghide_0.5.1-14build1_amd64.deb ...
Unpacking steghide (0.5.1-14build1) ...
Setting up libmhash2:amd64 (0.9.9.9-8) ...
Setting up libmcrypt4 (2.5.8-3.4) ...
Setting up steghide (0.5.1-14build1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
```

With steghide now installed, I could take a look at the pictures for the flag. (steghide is a tool used to hide data inside of bmp,jpg, au and wav files. It is mostly used in Kali Linux)

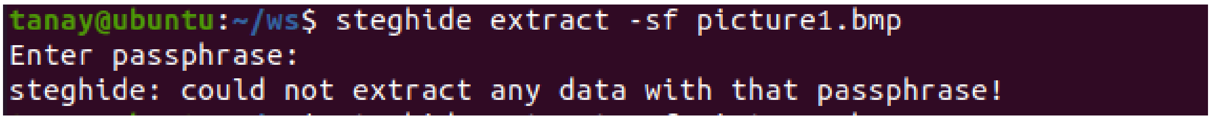
After some reading online, I figured out how to extract data from files using steghide. The command is:

steghide extract -sf <file>



```
tanay@ubuntu:~/ws$ steghide extract -sf picture1.bmp
Enter passphrase:
```

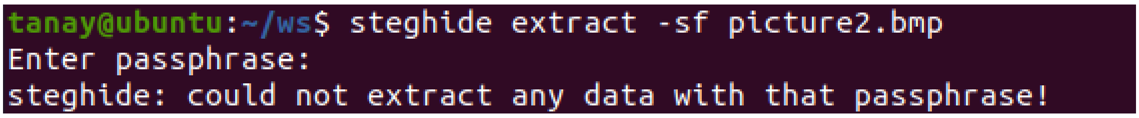
The passphrase is DUEDILIGENCE (as mentioned in plan.txt)



```
tanay@ubuntu:~/ws$ steghide extract -sf picture1.bmp
Enter passphrase:
steghide: could not extract any data with that passphrase!
```

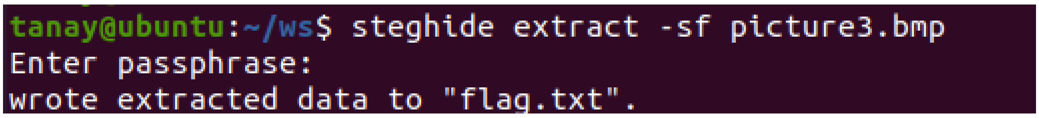
So there is no relevant data to us in picture1.bmp

Same is the case with picture2.bmp



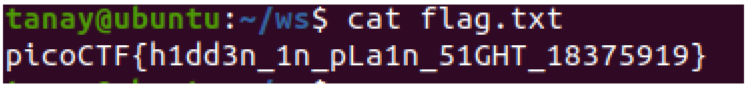
```
tanay@ubuntu:~/ws$ steghide extract -sf picture2.bmp
Enter passphrase:
steghide: could not extract any data with that passphrase!
```

But in picture3.bmp :



```
tanay@ubuntu:~/ws$ steghide extract -sf picture3.bmp
Enter passphrase:
wrote extracted data to "flag.txt".
```

So the flag is stored in flag.txt . All we have to do is simply display its contents.



```
tanay@ubuntu:~/ws$ cat flag.txt
picoCTF{h1dd3n_1n_pLa1n_51GHT_18375919}
```

This is our flag.

[Sources: dCode, StackOverflow, Google]