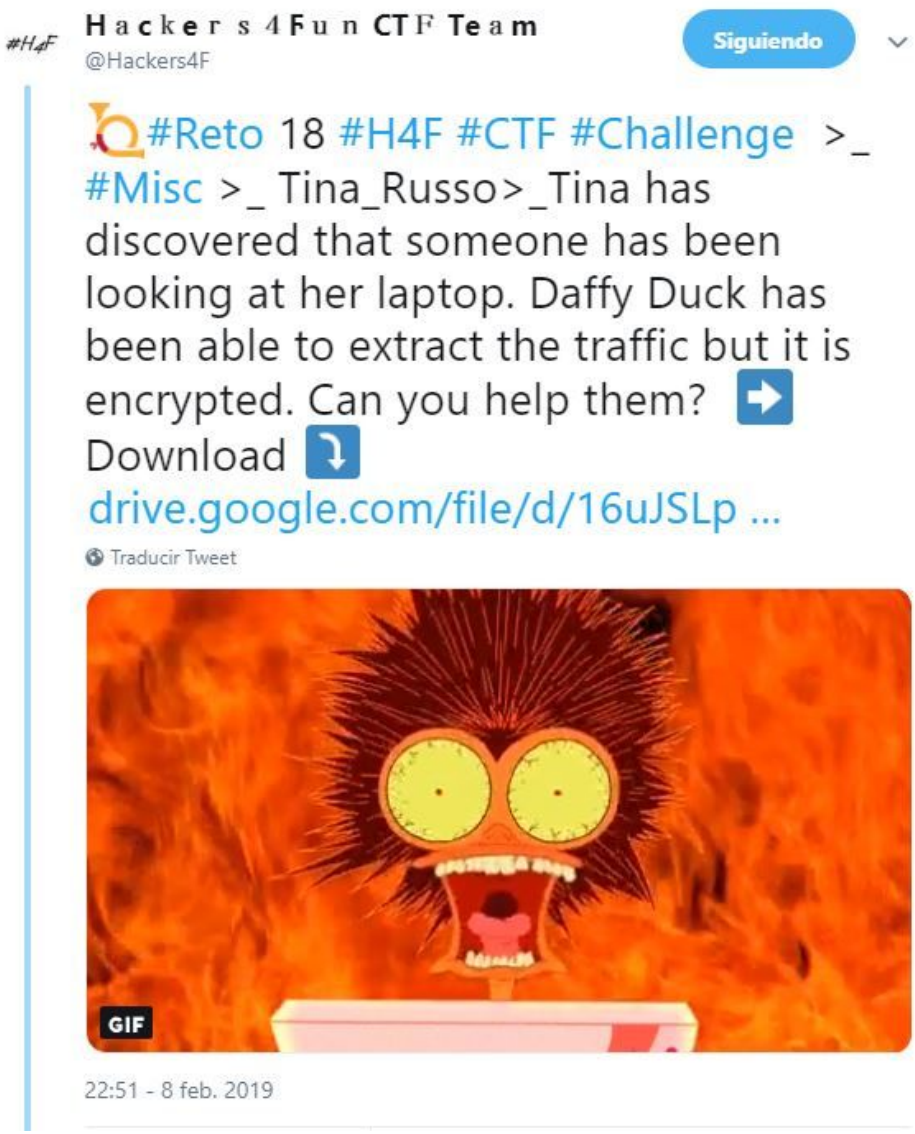



Nos descargamos el fichero .7z y vemos que tiene password el cual no nos proporcionan, así que vamos a crackearlo.

**root@kali:~/hackers4f#** perl 7z2hashcat.pl /media/sf_Downloads/R3t0_18_H4F_T1n4_Russ0.7z > /root/hackers4f/petar.txt
**root@kali:~/hackers4f#** john --wordlist=/usr/share/wordlists/rockyou.txt petar.txt
**root@kali:~/hackers4f#** john --show petar.txt
?:looneytunes

1 password hash cracked, 0 left

Con el password obtenido, descomprimimos el fichero. Tenemos un pcap, y por lo que parece es una captura de dos dispositivos USB. Sabemos que son dos, ya que el usb.device_address es diferente. Uno es el 19 y el otro el 20.

Bien, si nos fijamos un poco más, veremos que los dos parecen ser teclados.

Vamos a separar las pulsaciones de cada uno, de esta forma:

**root@kali:~/hackers4f#** tshark -r R3t0_18_H4F_T1n4_Russ0.pcap -Y "usb.transfer_type == 0x01 && usb.bInterfaceClass==3 && usb.device_address==19" -Tfields -e usb.capdata > keyboard1

**root@kali:~/hackers4f#** tshark -r R3t0_18_H4F_T1n4_Russ0.pcap -Y "usb.transfer_type == 0x01 && usb.bInterfaceClass==3 && usb.device_address==20" -Tfields -e usb.capdata > keyboard2

Bien, lo que vamos a hacer ahora es .. mediante python ver las pulsaciones de cada teclado

********** **BEGIN PYTHON CODE** **********

```
import sys

usb_codes = {
   0x04:"aA", 0x05:"bB", 0x06:"cC", 0x07:"dD", 0x08:"eE", 0x09:"fF",
   0x0A:"gG", 0x0B:"hH", 0x0C:"iI", 0x0D:"jJ", 0x0E:"kK", 0x0F:"lL",
   0x10:"mM", 0x11:"nN", 0x12:"oO", 0x13:"pP", 0x14:"qQ", 0x15:"rR",
   0x16:"sS", 0x17:"tT", 0x18:"uU", 0x19:"vV", 0x1A:"wW", 0x1B:"xX",
   0x1C:"yY", 0x1D:"zZ", 0x1E:"1!", 0x1F:"2@", 0x20:"3#", 0x21:"4$",
   0x22:"5%", 0x23:"6^", 0x24:"7&", 0x25:"8*", 0x26:"9(", 0x27:"0)",
   0x2C:"  ", 0x2D:"-_", 0x2E:"=+", 0x2F:"[{", 0x30:"]}",  0x32:"#~",
   0x33:";:", 0x34:"'\"",  0x36:",<",  0x5d:"5", 0x59:"1", 0x62:"0" , 0x37:".>"
   }

lines = ["","","","","",""]

pos = 0

for x in open(sys.argv[1],"r").readlines():
   code = int(x[6:8],16)

   if code == 0:
      continue
```

```
    # newline or down arrow - move down
    if code == 0x51 or code == 0x28:
        pos += 1
        continue
    # up arrow - move up
    if code == 0x52:
        pos -= 1
        continue

    # select the character based on the Shift key
    if int(x[0:2],16) == 2:
        lines[pos] += usb_codes[code][1]
    else:
        lines[pos] += usb_codes[code][0]


for x in lines:
    print x
```

********** **END PYTHON CODE** **********

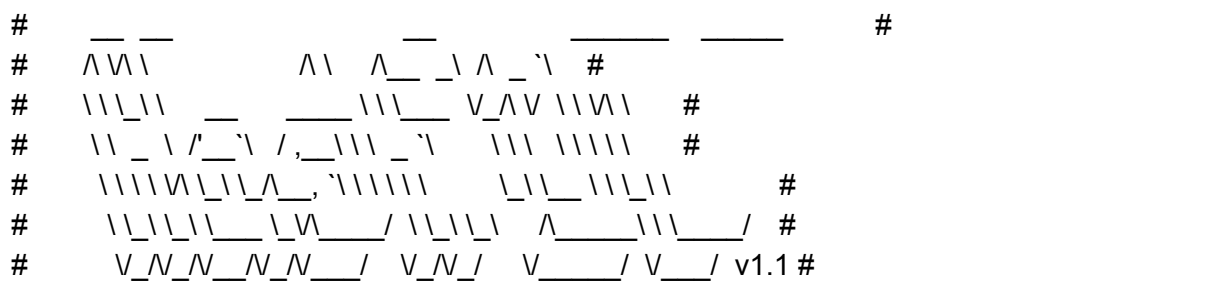
Con este programa tendremos parte de la solución:

**root@kali:~/hackers4f#** python solve.py keyboard1
792115c4d4ed3ea4b04a6af529a95d21

**root@kali:~/hackers4f#** python solve.py keyboard2
H0EFE2HjMmOMnx5BMSEJMyEHDayJER55IyEBp01RFxkAITj5

## Primera cadena:

**root@kali:~/hackers4f#** hash-identifier



```
  #                                                          By Zion3R #
  #                                                     www.Blackploit.com #
  #                                                    Root@Blackploit.com #
  ##############################################################################

  ----------------------------------------------------------------------
 HASH: 792115c4d4ed3ea4b04a6af529a95d21

Possible Hashs:
[+]  MD5

The MD5 hash:
792115C4D4ED3EA4B04A6AF529A95D21
was succesfully reversed into the string:
FR33_M0rt3ru3l0CON_4j04rr13r0L4BS
```

## Segunda Cadena:

**root@kali:~/hackers4f#** audodecoder.py --message H0EFE2HjMmOMnx5BMSEJMyEHDayJER55IyEBp01RFxkAITj5 --levels 3 -p H4F

```
                     (
   (            (       )\ )                (
   )\       (    )\ )    (()/(     (           )\ )   (   (
((((_)(    ))\  (()/(  (  /(_))   ))\  (   (   (()/(  ))\  )(
 )\ _ )\  /((_)  ((_)) )\(_))_    /((_) )\  )\   ((_))/((_)()\
 (_)_\(_)(_))(   _| | ((_)|   \ (_))  ((_)((_)  _| |(_))   ((_)
  / _ \  | || |/ _` |/ _ \| |) |/ -_)/ _|/ _ \/ _` |/ -_) | '_|
 /_/ \_\  \_,_|\__,_|\___/|___/ \___|\__|\___/\__,_|\___| |_|

                  Author: oreos | Twitter: @oreos_ES
```

rot13 > base64 > base64: H4F{H4b3Mu5_M0rT3rU3l02K19}

Finalmente, el flag es el de la segunda cadena, parece ser que el de la primera era solo para despistar.

## DarkEagle