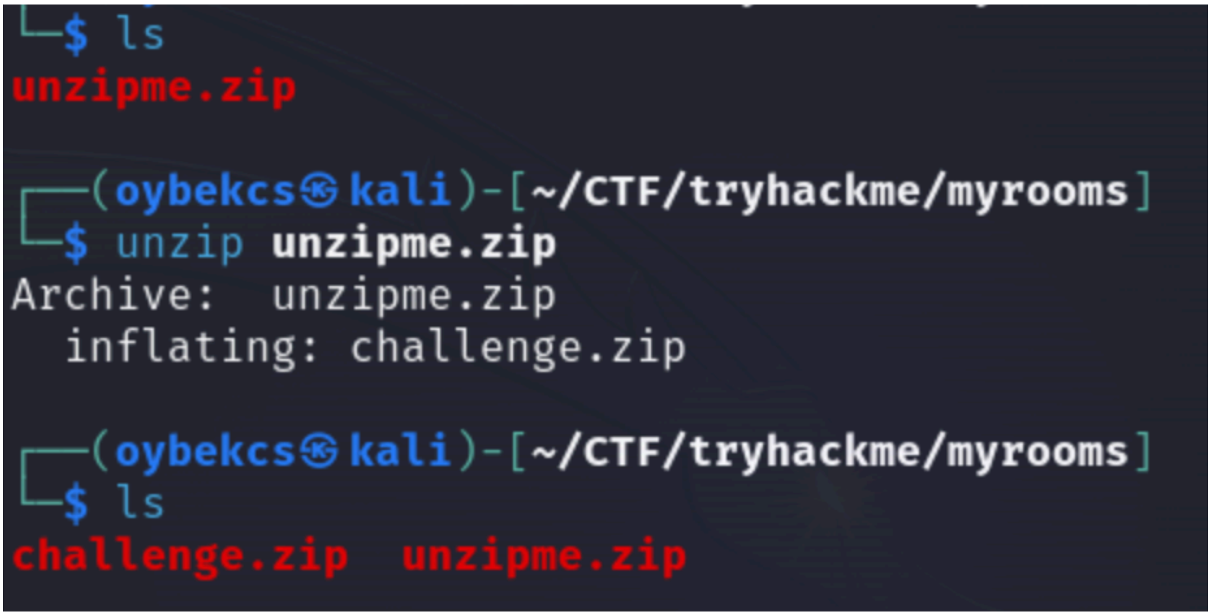
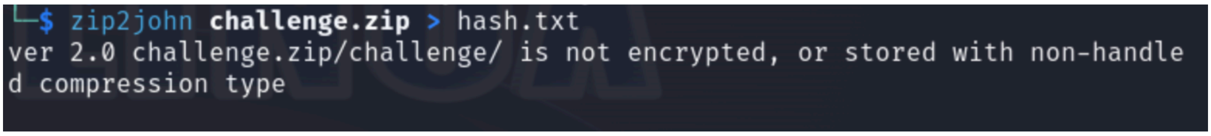
Walkthrough for the Stego challenge

First thing is first zip file has to be downloaded and unzipped.



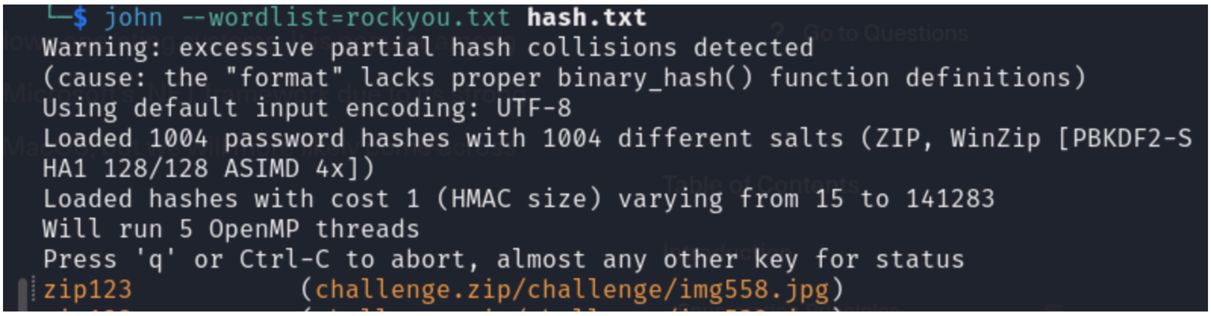
```
└─$ ls
unzipme.zip

┌──(oybekcs㉿kali)-[~/CTF/tryhackme/myrooms]
└─$ unzip unzipme.zip
Archive:  unzipme.zip
  inflating: challenge.zip

┌──(oybekcs㉿kali)-[~/CTF/tryhackme/myrooms]
└─$ ls
challenge.zip  unzipme.zip
```

After unzipping there is another zip file which has a password. So it should be cracked.



```
└─$ zip2john challenge.zip > hash.txt
ver 2.0 challenge.zip/challenge/ is not encrypted, or stored with non-handle
d compression type
```

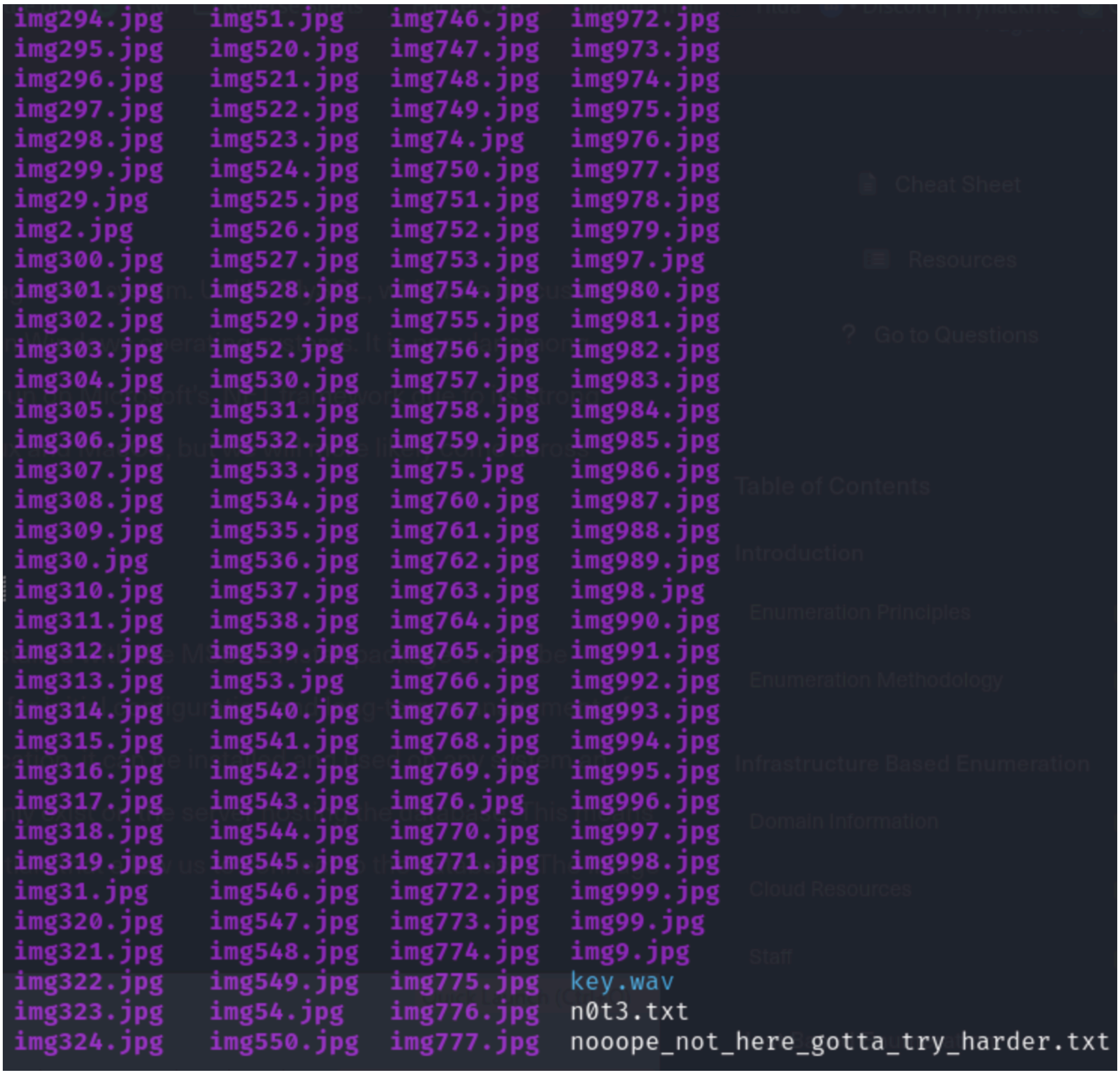
zip file has been converted to a file hash which john the ripper can understand, now john can be used to crack the zip password.



```
  └─$ john --wordlist=rockyou.txt hash.txt
  Warning: excessive partial hash collisions detected
  (cause: the "format" lacks proper binary_hash() function definitions)
  Using default input encoding: UTF-8
  Loaded 1004 password hashes with 1004 different salts (ZIP, WinZip [PBKDF2-S
  HA1 128/128 ASIMD 4x])
  Loaded hashes with cost 1 (HMAC size) varying from 15 to 141283
  Will run 5 OpenMP threads
  Press 'q' or Ctrl-C to abort, almost any other key for status
  zip123           (challenge.zip/challenge/img558.jpg)
```

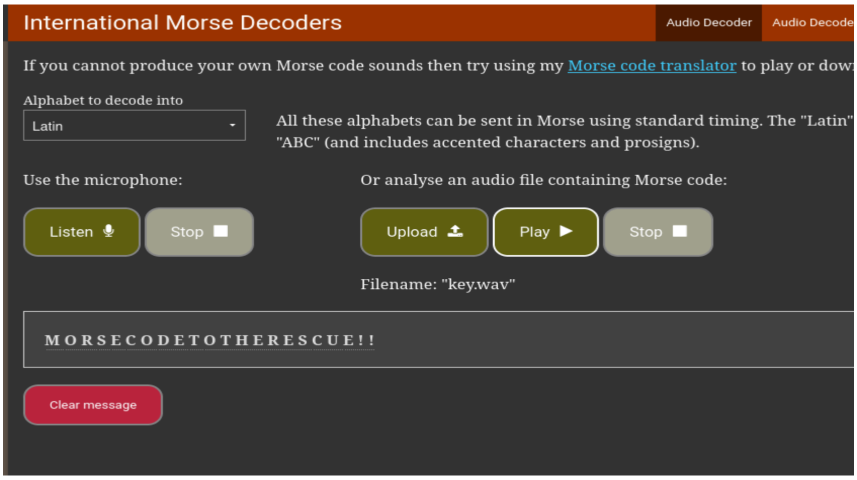
So password is zip123 for the second zipped file.

NOTE: It can take a while so doing it on your main OS is recommended.



```
img294.jpg   img51.jpg   img746.jpg  img972.jpg
img295.jpg   img520.jpg  img747.jpg  img973.jpg
img296.jpg   img521.jpg  img748.jpg  img974.jpg
img297.jpg   img522.jpg  img749.jpg  img975.jpg
img298.jpg   img523.jpg  img74.jpg   img976.jpg
img299.jpg   img524.jpg  img750.jpg  img977.jpg
img29.jpg    img525.jpg  img751.jpg  img978.jpg
img2.jpg     img526.jpg  img752.jpg  img979.jpg
img300.jpg   img527.jpg  img753.jpg  img97.jpg
img301.jpg   img528.jpg  img754.jpg  img980.jpg
img302.jpg   img529.jpg  img755.jpg  img981.jpg
img303.jpg   img52.jpg   img756.jpg  img982.jpg
img304.jpg   img530.jpg  img757.jpg  img983.jpg
img305.jpg   img531.jpg  img758.jpg  img984.jpg
img306.jpg   img532.jpg  img759.jpg  img985.jpg
img307.jpg   img533.jpg  img75.jpg   img986.jpg
img308.jpg   img534.jpg  img760.jpg  img987.jpg
img309.jpg   img535.jpg  img761.jpg  img988.jpg
img30.jpg    img536.jpg  img762.jpg  img989.jpg
img310.jpg   img537.jpg  img763.jpg  img98.jpg
img311.jpg   img538.jpg  img764.jpg  img990.jpg
img312.jpg   img539.jpg  img765.jpg  img991.jpg
img313.jpg   img53.jpg   img766.jpg  img992.jpg
img314.jpg   img540.jpg  img767.jpg  img993.jpg
img315.jpg   img541.jpg  img768.jpg  img994.jpg
img316.jpg   img542.jpg  img769.jpg  img995.jpg
img317.jpg   img543.jpg  img76.jpg   img996.jpg
img318.jpg   img544.jpg  img770.jpg  img997.jpg
img319.jpg   img545.jpg  img771.jpg  img998.jpg
img31.jpg    img546.jpg  img772.jpg  img999.jpg
img320.jpg   img547.jpg  img773.jpg  img99.jpg
img321.jpg   img548.jpg  img774.jpg  img9.jpg
img322.jpg   img549.jpg  img775.jpg  key.wav
img323.jpg   img54.jpg   img776.jpg  n0t3.txt
img324.jpg   img550.jpg  img777.jpg  nooope_not_here_gotta_try_harder.txt
```

We have bunch of files wav file has some morse code in it using online tools such as morsecode.world can help with this



International Morse Decoders

If you cannot produce your own Morse code sounds then try using my Morse code translator to play or dow

Alphabet to decode into: Latin

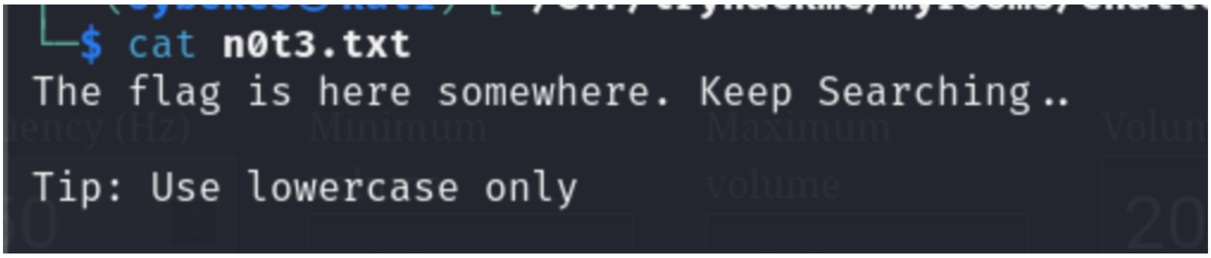
All these alphabets can be sent in Morse using standard timing. The "Latin" "ABC" (and includes accented characters and prosigns).

Use the microphone:

Or analyse an audio file containing Morse code:

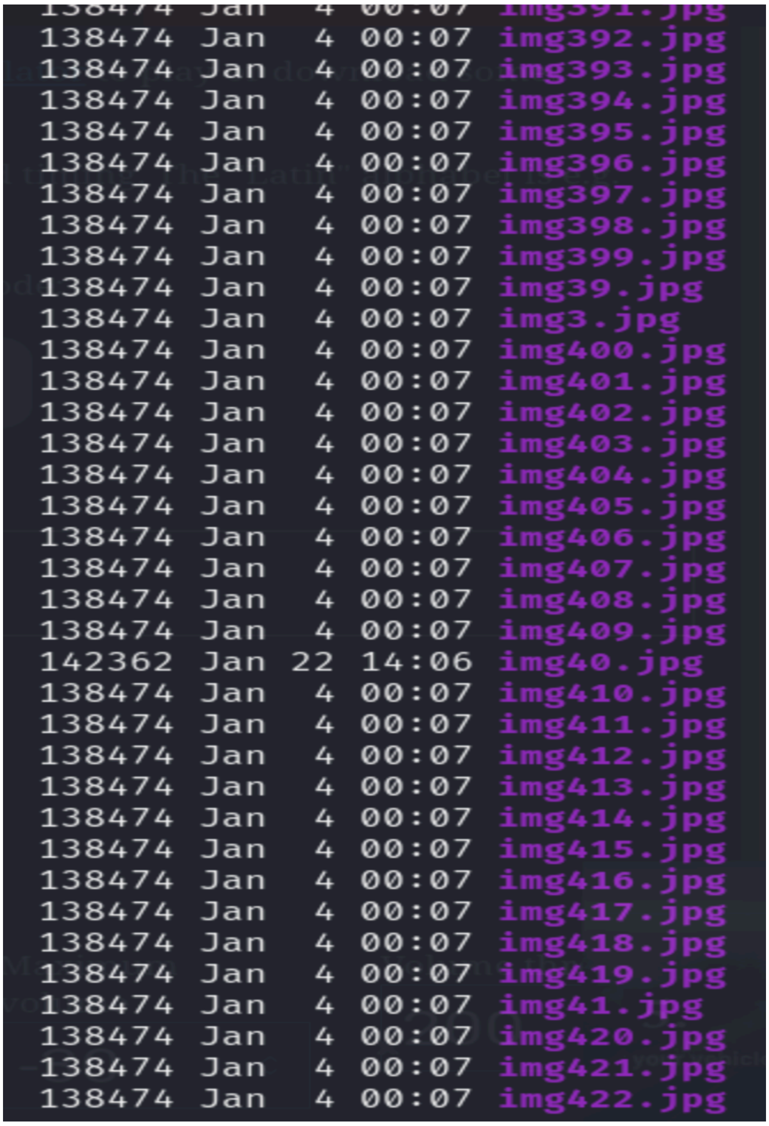
Filename: "key.wav"

MORSECODETOTHERESCUE!!



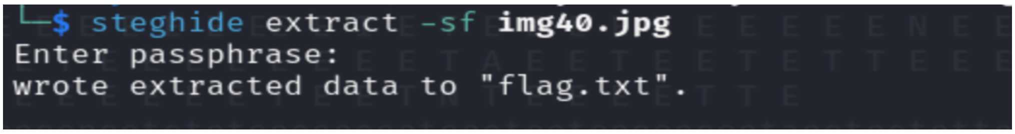
```
└─$ cat n0t3.txt
The flag is here somewhere. Keep Searching..

Tip: Use lowercase only
```

Hint is saying that we should use only lowercase letters bear in mind!

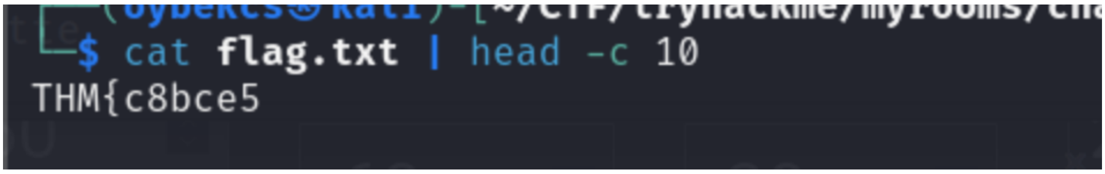


```
138474 Jan  4 00:07 img392.jpg
138474 Jan  4 00:07 img393.jpg
138474 Jan  4 00:07 img394.jpg
138474 Jan  4 00:07 img395.jpg
138474 Jan  4 00:07 img396.jpg
138474 Jan  4 00:07 img397.jpg
138474 Jan  4 00:07 img398.jpg
138474 Jan  4 00:07 img399.jpg
138474 Jan  4 00:07 img39.jpg
138474 Jan  4 00:07 img3.jpg
138474 Jan  4 00:07 img400.jpg
138474 Jan  4 00:07 img401.jpg
138474 Jan  4 00:07 img402.jpg
138474 Jan  4 00:07 img403.jpg
138474 Jan  4 00:07 img404.jpg
138474 Jan  4 00:07 img405.jpg
138474 Jan  4 00:07 img406.jpg
138474 Jan  4 00:07 img407.jpg
138474 Jan  4 00:07 img408.jpg
138474 Jan  4 00:07 img409.jpg
142362 Jan 22 14:06 img40.jpg
138474 Jan  4 00:07 img410.jpg
138474 Jan  4 00:07 img411.jpg
138474 Jan  4 00:07 img412.jpg
138474 Jan  4 00:07 img413.jpg
138474 Jan  4 00:07 img414.jpg
138474 Jan  4 00:07 img415.jpg
138474 Jan  4 00:07 img416.jpg
138474 Jan  4 00:07 img417.jpg
138474 Jan  4 00:07 img418.jpg
138474 Jan  4 00:07 img419.jpg
138474 Jan  4 00:07 img41.jpg
138474 Jan  4 00:07 img420.jpg
138474 Jan  4 00:07 img421.jpg
138474 Jan  4 00:07 img422.jpg
```

Except one file which is img40.jpg all files have the same size in bytes. We can use steghide to extract using the word we retrieved from wav file which is MORSECODETOTHERESCUE!! but in lowercase since it was mentioned.



```
└─$ steghide extract -sf img40.jpg
Enter passphrase:
wrote extracted data to "flag.txt".
```



```
┌──(oybekcs㉿kali)-[~/CTF/tryhackme/myrooms/cha
└─$ cat flag.txt | head -c 10
THM{c8bce5
```

We have the flag !

Thanks for taking time to read this walkthrough.