HACK@10

CAPTURE IF YOU CAN

HACK@10 CTF – CAPTURE IF YOU CAN

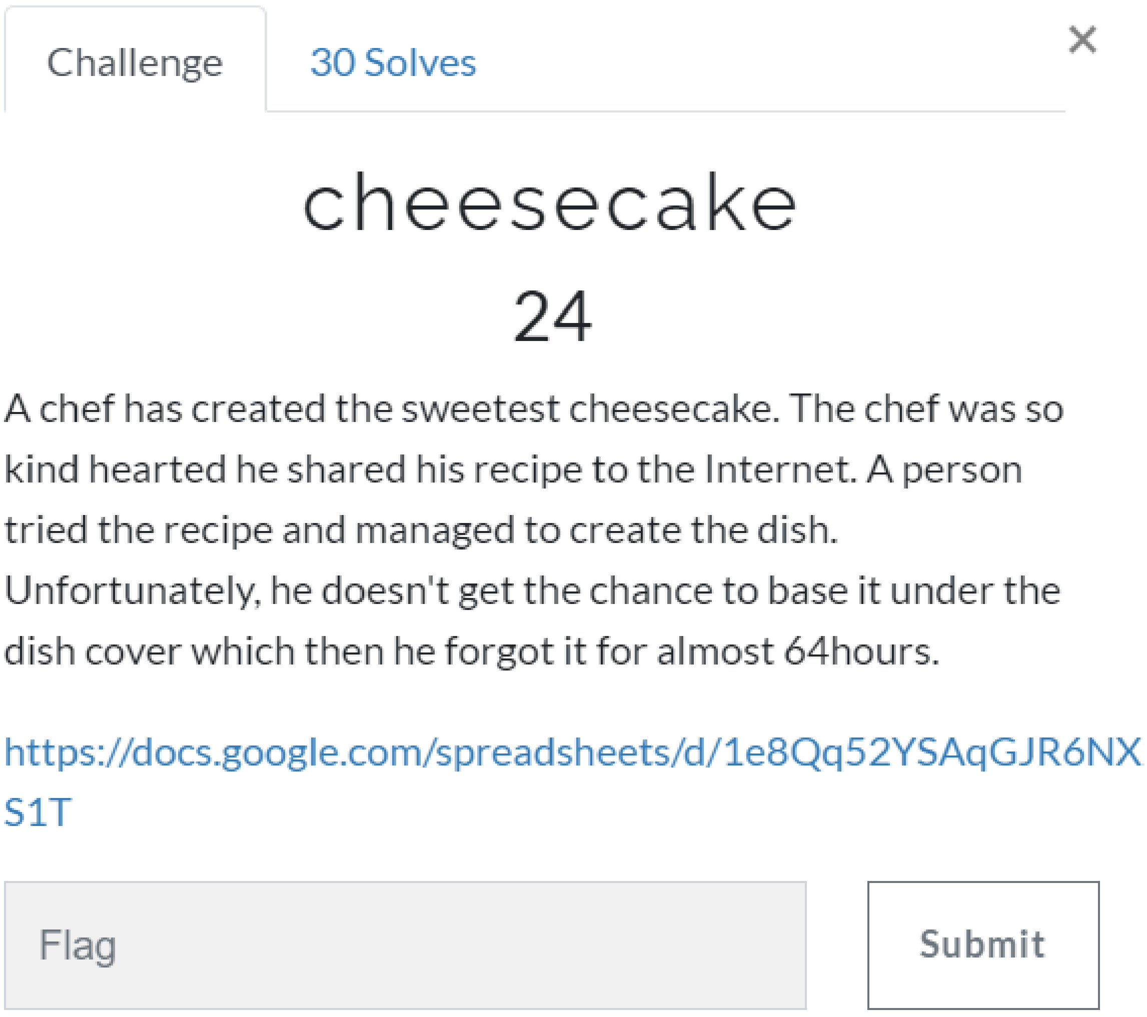
Write-ups

Team Name: x0rry

Team Leader: Wesley Wong Kee Han

Team Member: Lim Wei Xun, Ong Fo Seng

MISC: cheesecake



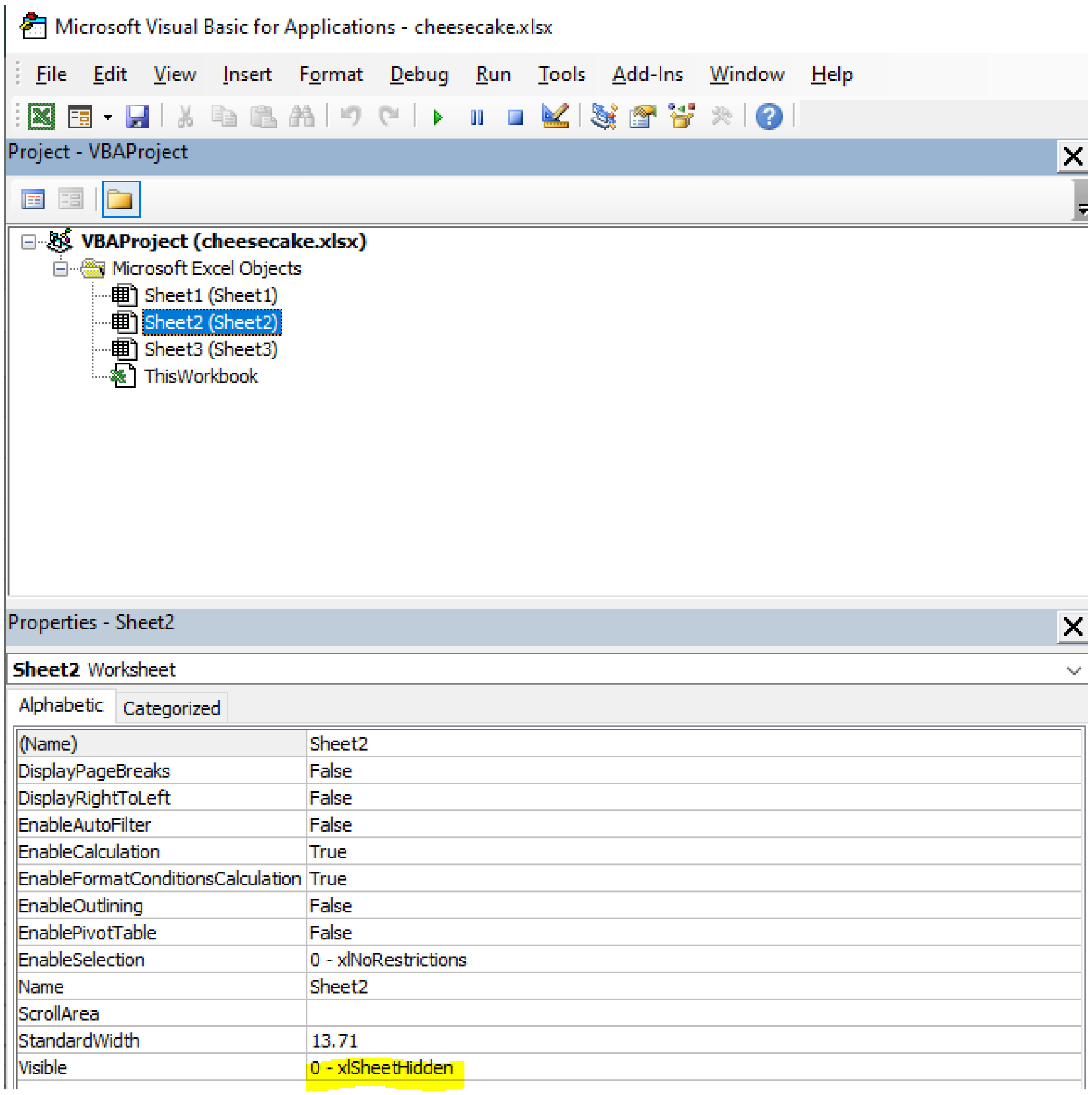
cheesecake

24

A chef has created the sweetest cheesecake. The chef was so kind hearted he shared his recipe to the Internet. A person tried the recipe and managed to create the dish. Unfortunately, he doesn't get the chance to base it under the dish cover which then he forgot it for almost 64hours.

https://docs.google.com/spreadsheets/d/1e8Qq52YSAqGJR6NXS1T

Download the Google Spread Sheet as excel.


Microsoft Visual Basic for Applications - cheesecake.xlsx

File Edit View Insert Format Debug Run Tools Add-Ins Window Help

Project - VBAProject

```
VBAProject (cheesecake.xlsx)
  Microsoft Excel Objects
    Sheet1 (Sheet1)
    Sheet2 (Sheet2)
    Sheet3 (Sheet3)
    ThisWorkbook
```

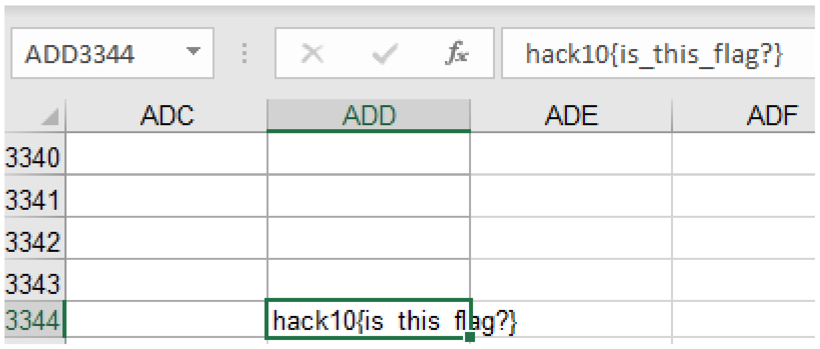
Properties - Sheet2

Sheet2 Worksheet

Alphabetic | Categorized

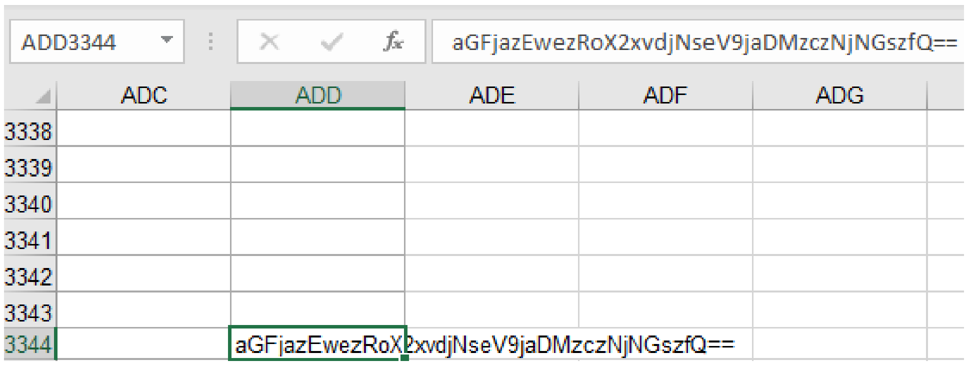
(Name)	Sheet2
DisplayPageBreaks	False
DisplayRightToLeft	False
EnableAutoFilter	False
EnableCalculation	True
EnableFormatConditionsCalculation	True
EnableOutlining	False
EnablePivotTable	False
EnableSelection	0 - xlNoRestrictions
Name	Sheet2
ScrollArea	
StandardWidth	13.71
Visible	0 - xlSheetHidden

ALT + F11 to enter Microsoft Visual Basic Editor. 2 sheets are found to be hidden with modified properties. Modify the attribute “visible” to “1”.


ADD3344 | hack10{is_this_flag?}

	ADC	ADD	ADE	ADF
3340				
3341				
3342				
3343				
3344		hack10{is_this_flag?}		

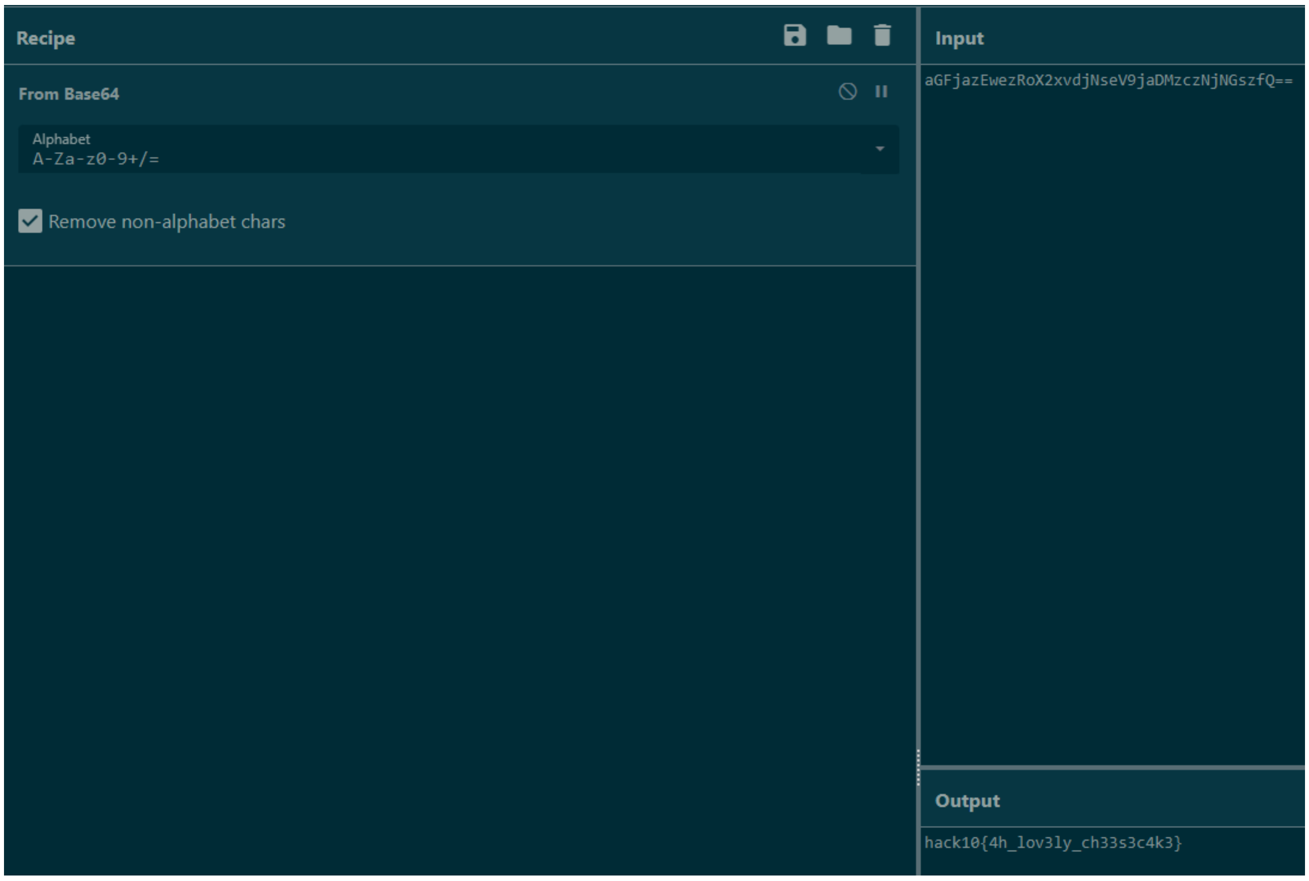
On Sheet2, we tried searching for flag by keyword “hack10”, but a fake flag is found instead.



```
ADD3344    aGFjazEwezRoX2xvdjNseV9jaDMzczNjNGszfQ==
```

	ADC	ADD	ADE	ADF	ADG
3338					
3339					
3340					
3341					
3342					
3343					
3344		aGFjazEwezRoX2xvdjNseV9jaDMzczNjNGszfQ==			

The title mentioned about 64 hours, which is a hint to base64. So we search for delimiter "==" instead in Sheet3.



From Base64
Alphabet
A-Za-z0-9+/=
✓ Remove non-alphabet chars

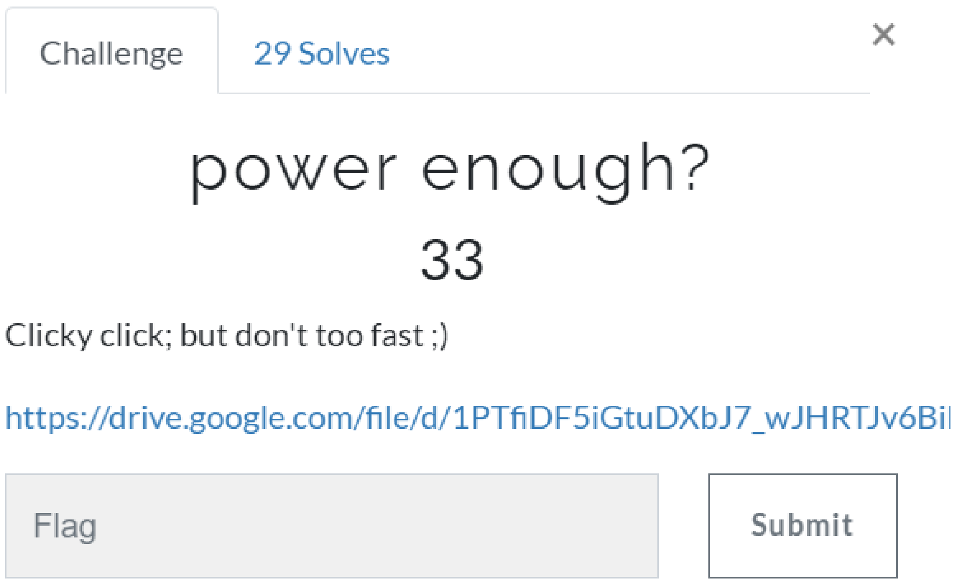
```
aGFjazEwezRoX2xvdjNseV9jaDMzczNjNGszfQ==
```

```
hack10{4h_lov3ly_ch33s3c4k3}
```

Decode the base64 string in CyberChef to obtain the final flag.

FLAG: hack10{4h_lov3ly_ch33s3c4k3}

MISC: power enough



Challenge

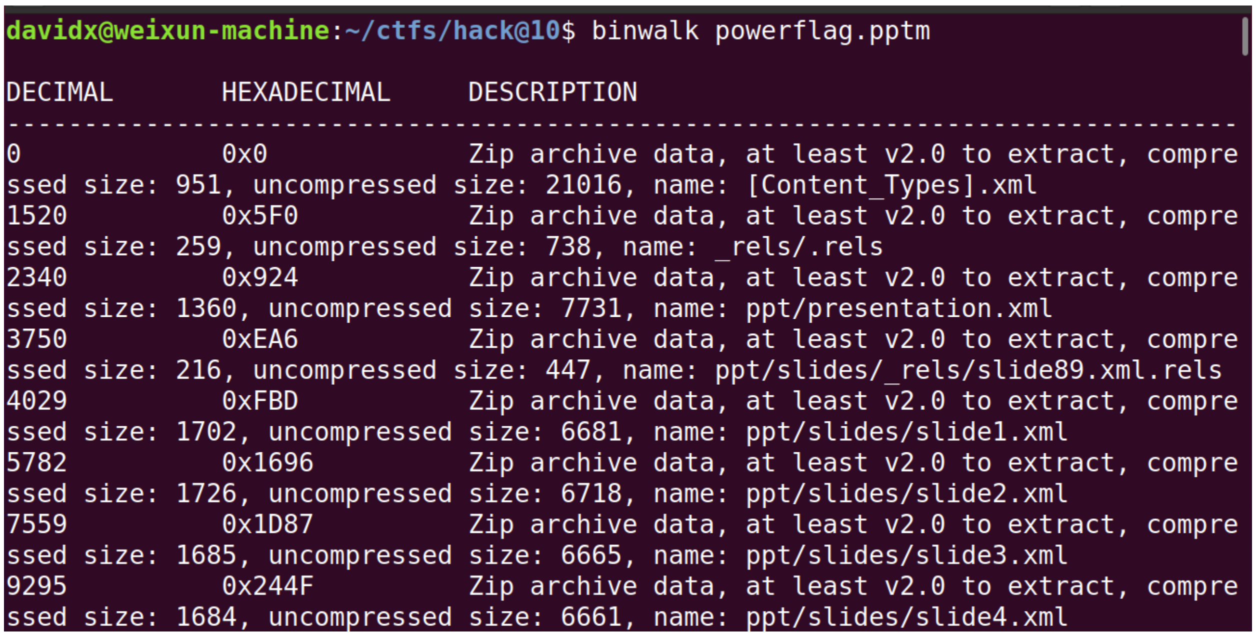
29 Solves

power enough?

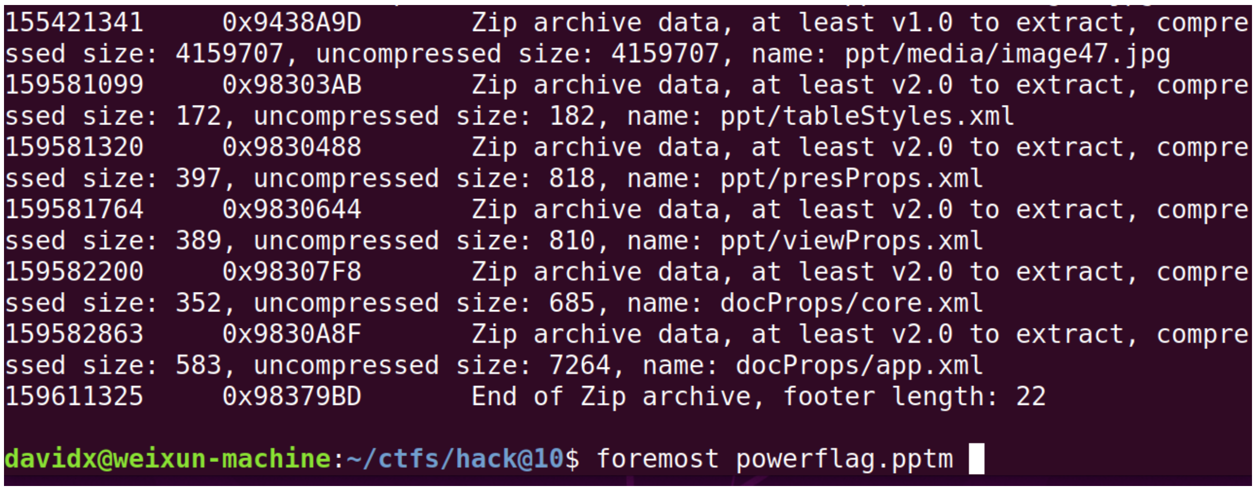
33

Clicky click; but don't too fast ;)

https://drive.google.com/file/d/1PTfiDF5iGtuDXbJ7_wJHRTJv6Bil

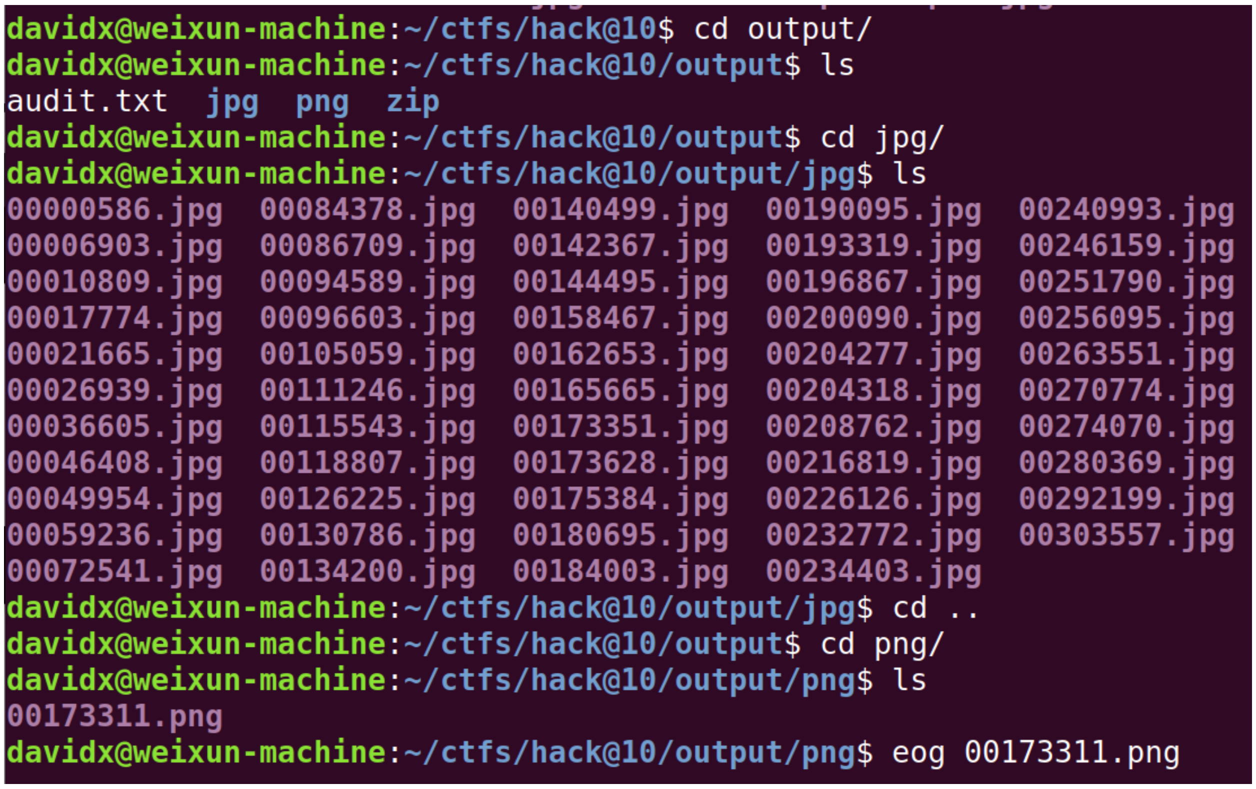


```
davidx@weixun-machine:~/ctfs/hack@10$ binwalk powerflag.pptm

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Zip archive data, at least v2.0 to extract, compre
ssed size: 951, uncompressed size: 21016, name: [Content_Types].xml
1520          0x5F0           Zip archive data, at least v2.0 to extract, compre
ssed size: 259, uncompressed size: 738, name: _rels/.rels
2340          0x924           Zip archive data, at least v2.0 to extract, compre
ssed size: 1360, uncompressed size: 7731, name: ppt/presentation.xml
3750          0xEA6           Zip archive data, at least v2.0 to extract, compre
ssed size: 216, uncompressed size: 447, name: ppt/slides/_rels/slide89.xml.rels
4029          0xFBD           Zip archive data, at least v2.0 to extract, compre
ssed size: 1702, uncompressed size: 6681, name: ppt/slides/slide1.xml
5782          0x1696          Zip archive data, at least v2.0 to extract, compre
ssed size: 1726, uncompressed size: 6718, name: ppt/slides/slide2.xml
7559          0x1D87          Zip archive data, at least v2.0 to extract, compre
ssed size: 1685, uncompressed size: 6665, name: ppt/slides/slide3.xml
9295          0x244F          Zip archive data, at least v2.0 to extract, compre
ssed size: 1684, uncompressed size: 6661, name: ppt/slides/slide4.xml
```



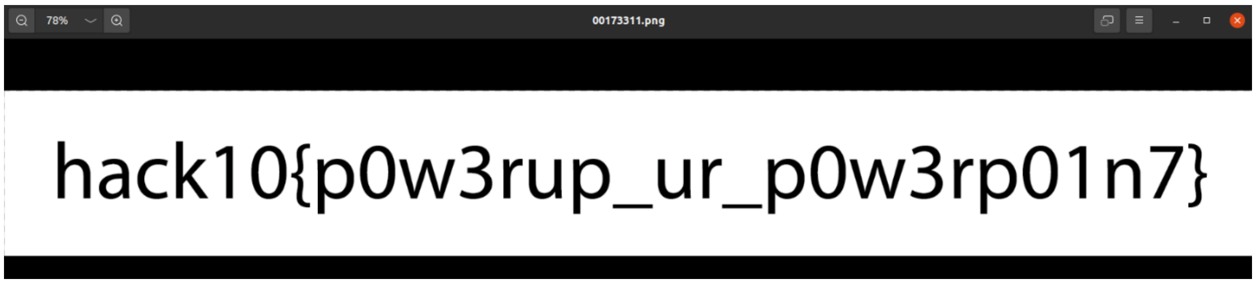
```
155421341     0x9438A9D       Zip archive data, at least v1.0 to extract, compre
ssed size: 4159707, uncompressed size: 4159707, name: ppt/media/image47.jpg
159581099     0x98303AB       Zip archive data, at least v2.0 to extract, compre
ssed size: 172, uncompressed size: 182, name: ppt/tableStyles.xml
159581320     0x9830488       Zip archive data, at least v2.0 to extract, compre
ssed size: 397, uncompressed size: 818, name: ppt/presProps.xml
159581764     0x9830644       Zip archive data, at least v2.0 to extract, compre
ssed size: 389, uncompressed size: 810, name: ppt/viewProps.xml
159582200     0x98307F8       Zip archive data, at least v2.0 to extract, compre
ssed size: 352, uncompressed size: 685, name: docProps/core.xml
159582863     0x9830A8F       Zip archive data, at least v2.0 to extract, compre
ssed size: 583, uncompressed size: 7264, name: docProps/app.xml
159611325     0x98379BD       End of Zip archive, footer length: 22

davidx@weixun-machine:~/ctfs/hack@10$ foremost powerflag.pptm
```

The .pptm file contains a large amount of embedded data as shown above using binwalk.



```
davidx@weixun-machine:~/ctfs/hack@10$ cd output/
davidx@weixun-machine:~/ctfs/hack@10/output$ ls
audit.txt  jpg  png  zip
davidx@weixun-machine:~/ctfs/hack@10/output$ cd jpg/
davidx@weixun-machine:~/ctfs/hack@10/output/jpg$ ls
00000586.jpg  00084378.jpg  00140499.jpg  00190095.jpg  00240993.jpg
00006903.jpg  00086709.jpg  00142367.jpg  00193319.jpg  00246159.jpg
00010809.jpg  00094589.jpg  00144495.jpg  00196867.jpg  00251790.jpg
00017774.jpg  00096603.jpg  00158467.jpg  00200090.jpg  00256095.jpg
00021665.jpg  00105059.jpg  00162653.jpg  00204277.jpg  00263551.jpg
00026939.jpg  00111246.jpg  00165665.jpg  00204318.jpg  00270774.jpg
00036605.jpg  00115543.jpg  00173351.jpg  00208762.jpg  00274070.jpg
00046408.jpg  00118807.jpg  00173628.jpg  00216819.jpg  00280369.jpg
00049954.jpg  00126225.jpg  00175384.jpg  00226126.jpg  00292199.jpg
00059236.jpg  00130786.jpg  00180695.jpg  00232772.jpg  00303557.jpg
00072541.jpg  00134200.jpg  00184003.jpg  00234403.jpg
davidx@weixun-machine:~/ctfs/hack@10/output/jpg$ cd ..
davidx@weixun-machine:~/ctfs/hack@10/output$ cd png/
davidx@weixun-machine:~/ctfs/hack@10/output/png$ ls
00173311.png
davidx@weixun-machine:~/ctfs/hack@10/output/png$ eog 00173311.png
```

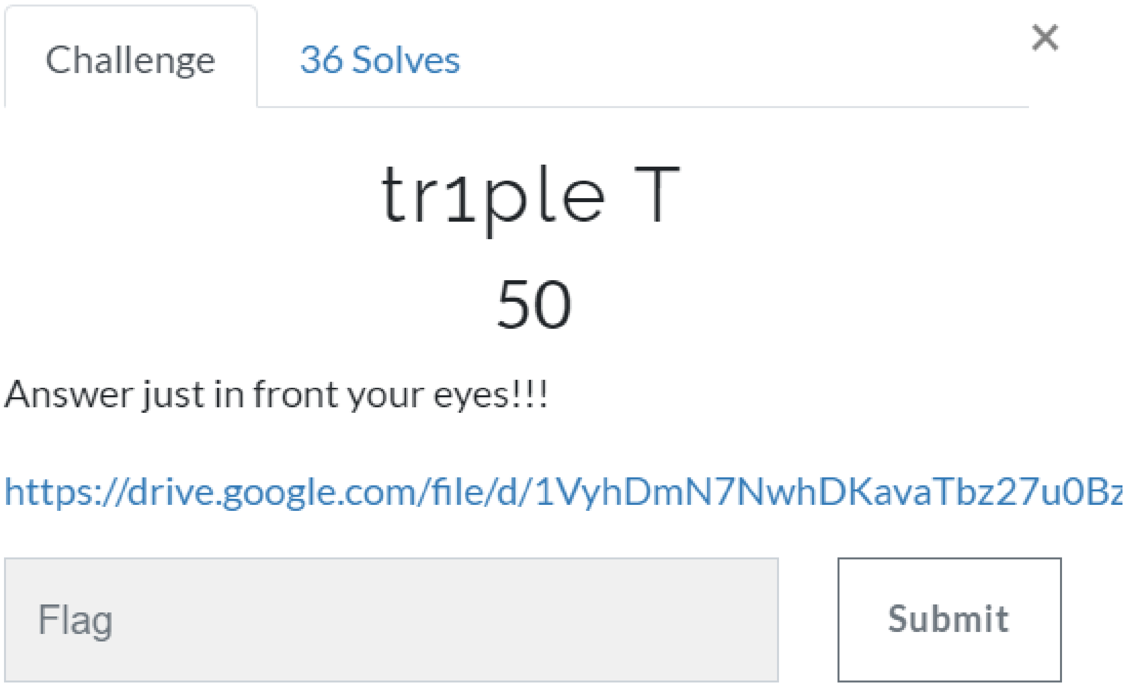
Most of the extracted images are the source image for the ppt background. The only standout image is with the .PNG extension.



hack10{p0w3rup_ur_p0w3rp01n7}

FLAG: **hack10{p0w3rup_ur_p0w3rp01n7}**

CRYPTO: tr1ple T


Challenge

36 Solves

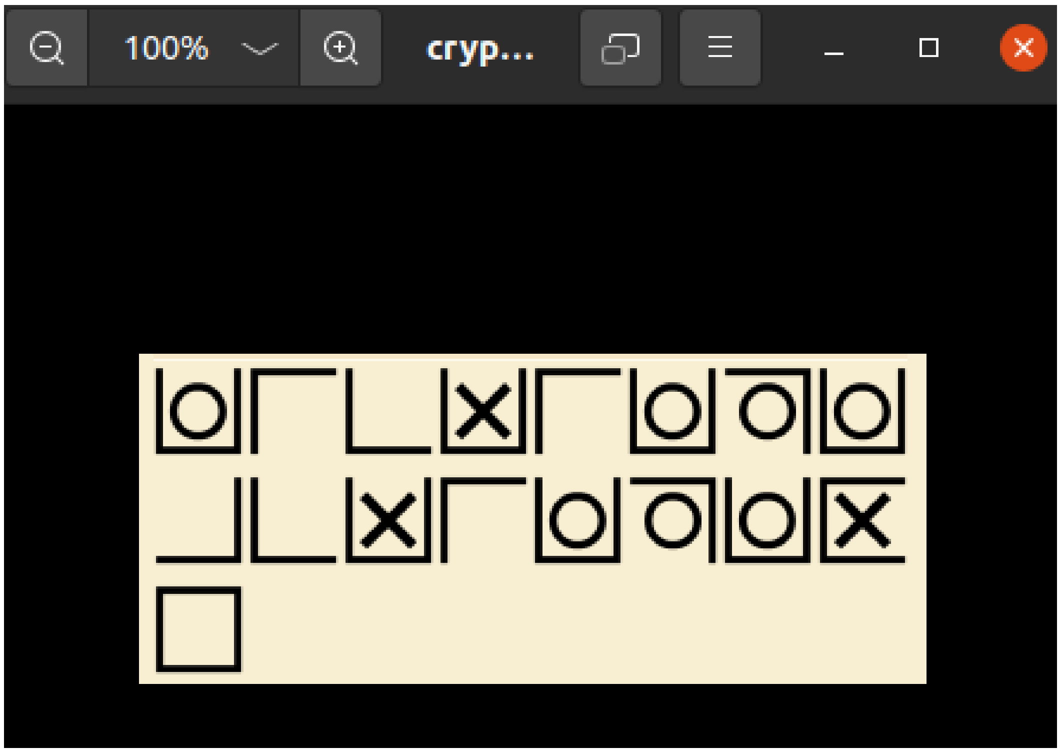
tr1ple T

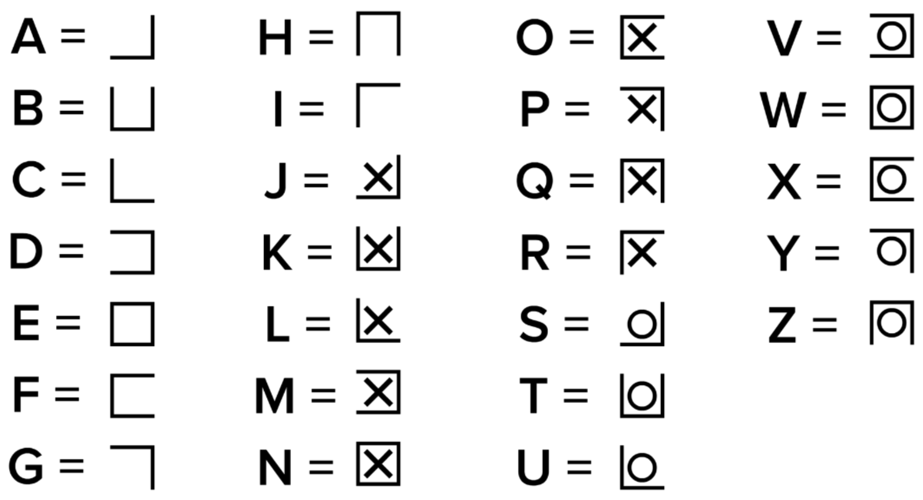
50

Answer just in front your eyes!!!

https://drive.google.com/file/d/1VyhDmN7NwhDKavaTbz27u0Bz

Triple T is a phrase that is hinting the term “Tic Tac Toe”. Hence, we attempt to decipher the tic tac toe images for letter representations.

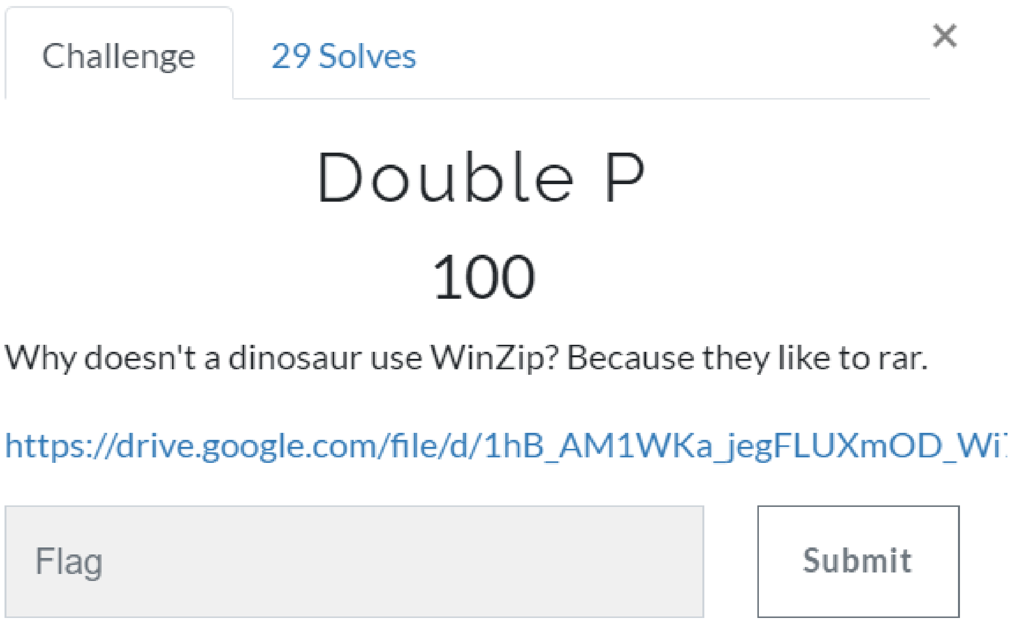




Use this link https://www.dcode.fr/tic-tac-toe-cipher to decrypt the message.

FLAG: hack10{TICKITYTACKITYTOE}

CRYPTO: Double P



Challenge 29 Solves

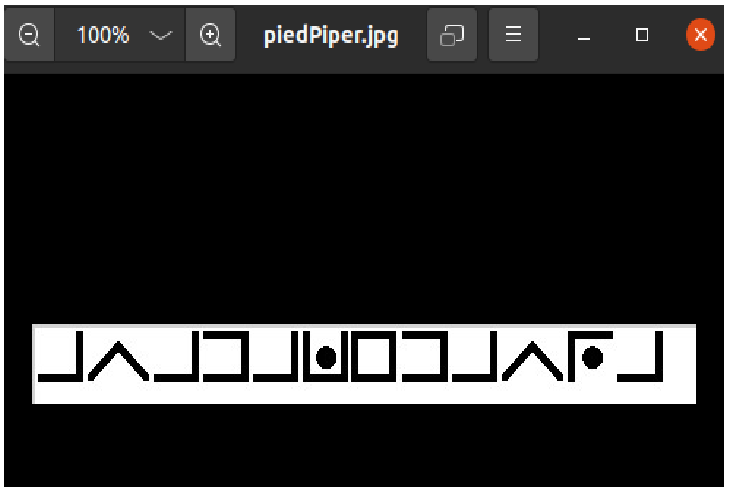
Double P

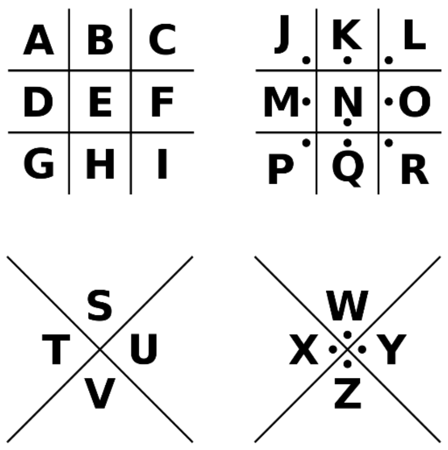
100

Why doesn't a dinosaur use WinZip? Because they like to rar.

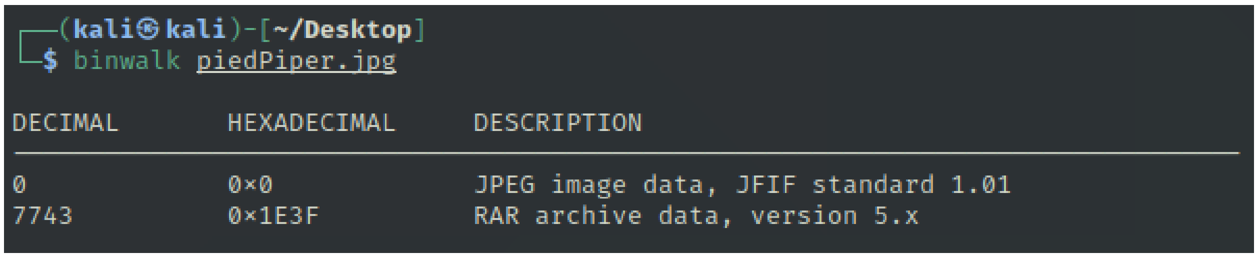
https://drive.google.com/file/d/1hB_AM1WKa_jegFLUXmOD_Wi

Double P is the acronym for "Pig Pen" encryption method. Therefore, we may refer to the characters list to decrypt the encryption.



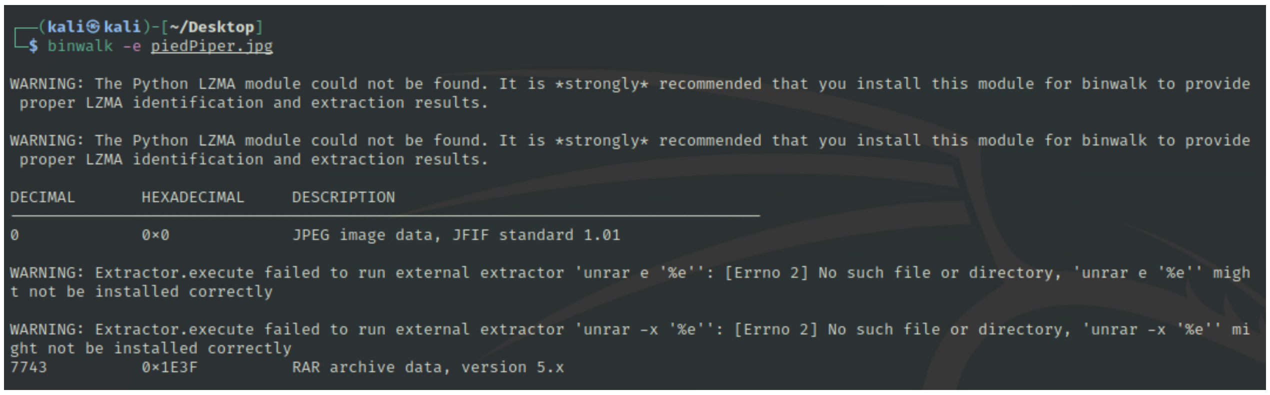


Password: AVADAKEDAVRA (we attempted to submit but that it is not the flag)



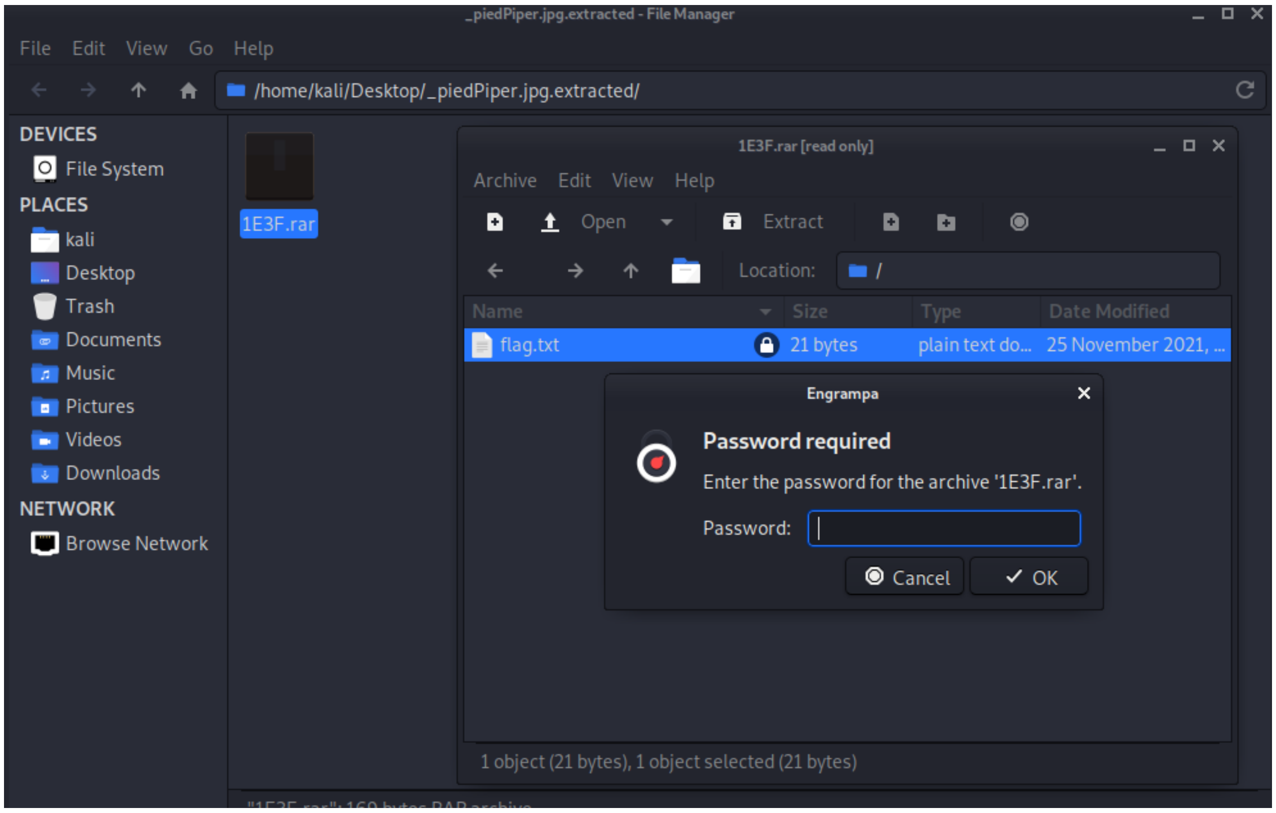
```
┌──(kali㉿kali)-[~/Desktop]
└─$ binwalk piedPiper.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
7743          0x1E3F          RAR archive data, version 5.x
```

We use binwalk to find other possible clues. We have found something interesting which is a RAR file embedded inside the .JPG file. After that, we add a -e flag to indicate the extraction function in binwalk.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ binwalk -e piedPiper.jpg

WARNING: The Python LZMA module could not be found. It is *strongly* recommended that you install this module for binwalk to provide
 proper LZMA identification and extraction results.

WARNING: The Python LZMA module could not be found. It is *strongly* recommended that you install this module for binwalk to provide
 proper LZMA identification and extraction results.

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01

WARNING: Extractor.execute failed to run external extractor 'unrar e '%e'': [Errno 2] No such file or directory, 'unrar e '%e'' migh
t not be installed correctly

WARNING: Extractor.execute failed to run external extractor 'unrar -x '%e'': [Errno 2] No such file or directory, 'unrar -x '%e'' mi
ght not be installed correctly
7743          0x1E3F          RAR archive data, version 5.x
```

After extraction, the RAR file had been extracted – 1E3F.rar. It requires a password which the string that we decrypted previously. It grants access and we got our flag.


_piedPiper.jpg.extracted - File Manager

/home/kali/Desktop/_piedPiper.jpg.extracted/

1E3F.rar [read only]

Name	Size	Type	Date Modified
flag.txt	21 bytes	plain text do...	25 November 2021, ...

Engrampa

Password required

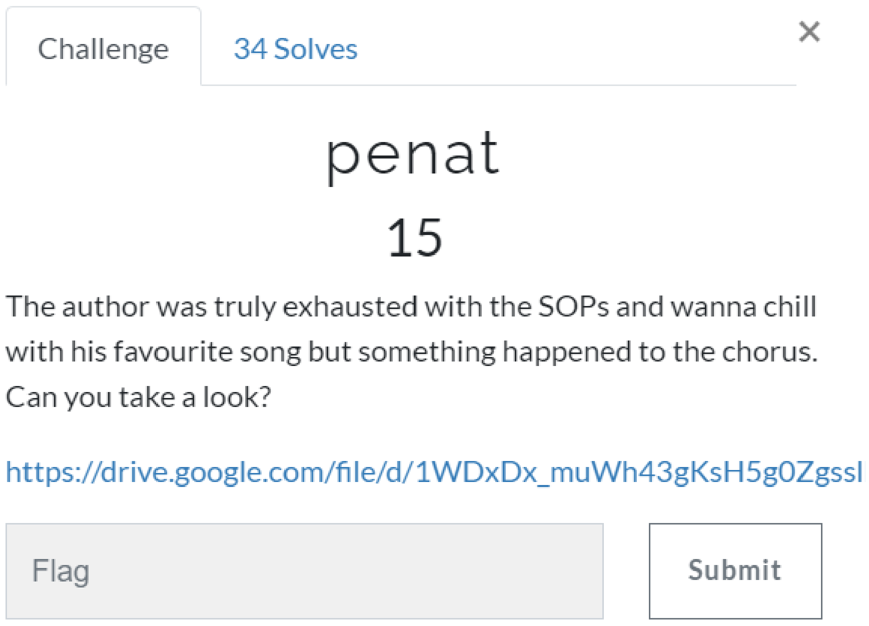
Enter the password for the archive '1E3F.rar'.

Password:

1 object (21 bytes), 1 object selected (21 bytes)

FLAG: hack10{y0u_f0uNd_m3!}

STEG: penat



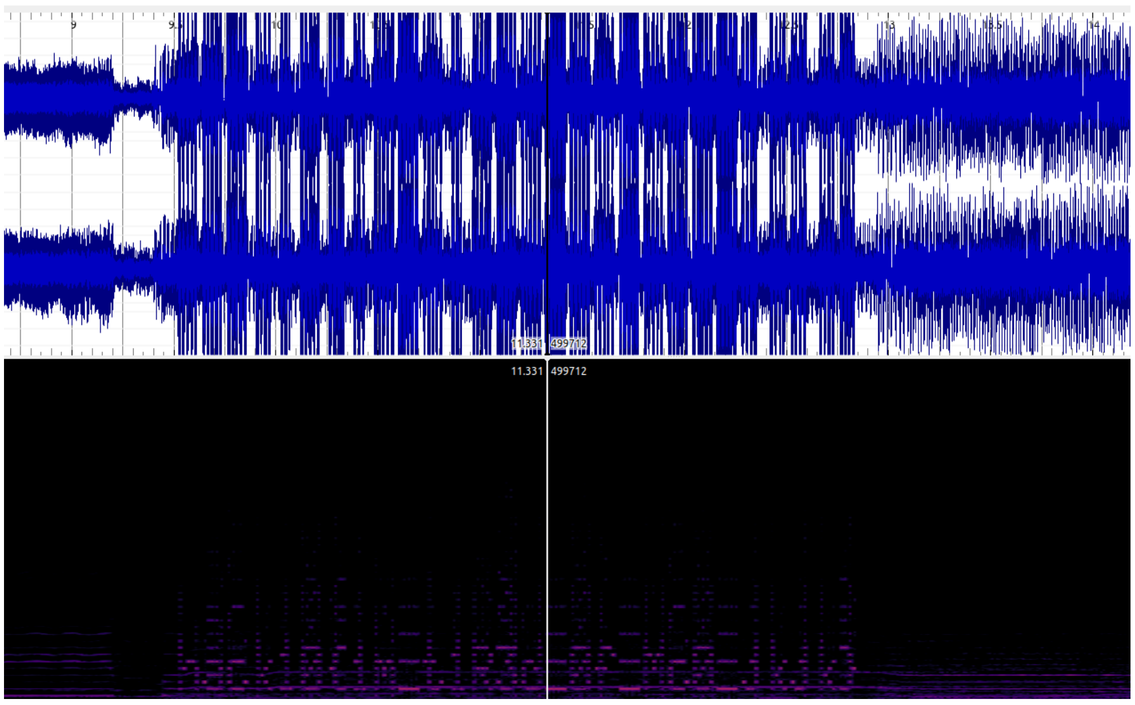
penat

15

The author was truly exhausted with the SOPs and wanna chill with his favourite song but something happened to the chorus. Can you take a look?

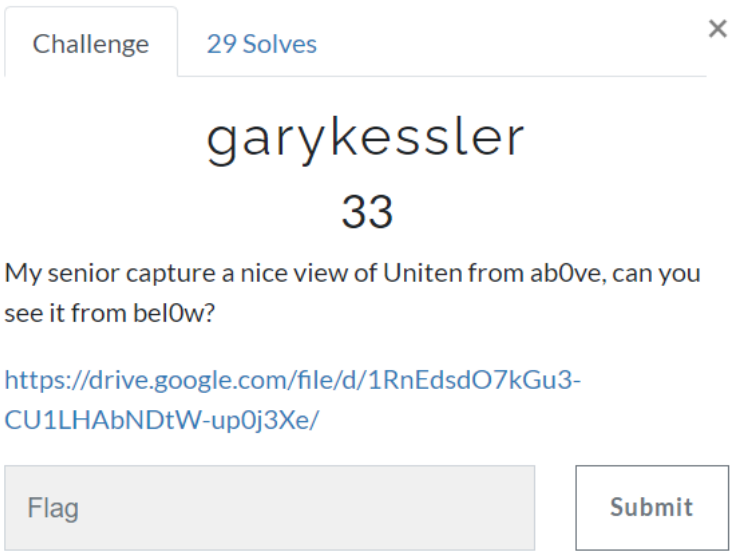
https://drive.google.com/file/d/1WDxDx_muWh43gKsH5g0Zgssl

The downloadable file is an audio file (.WAV). There is a screeching sound at the middle when audio is played. Therefore, we use sonic visualizer to inspect the audio by adding spectrogram. We found the hidden flag after adjusting to a suitable vision.



FLAG: hack10{1m_t1R3d_0f_Mc0_jkjk}

STEG: garykessler


Challenge | 29 Solves

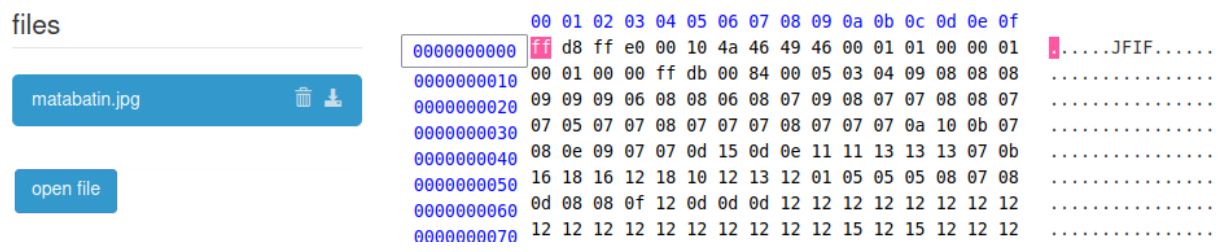
garykessler

33

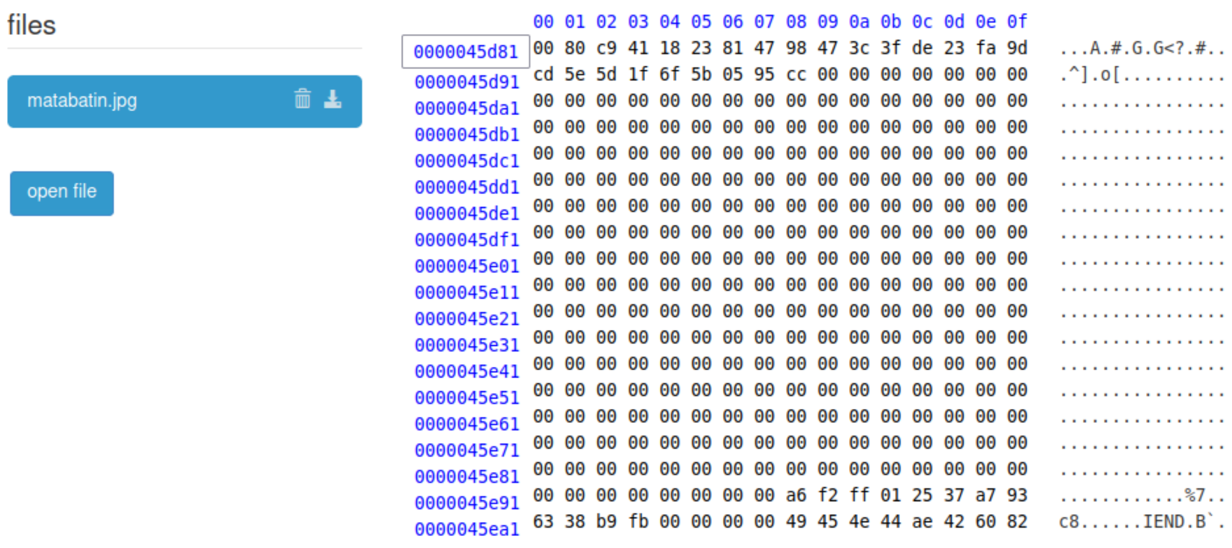
My senior capture a nice view of Uniten from ab0ve, can you see it from bel0w?

https://drive.google.com/file/d/1RnEdsdO7kGu3-CU1LHAbNDtW-up0j3Xe/

The downloadable file is an image file (.JPG). We look through the image file in hex editor and we found file signature mismatch at the header and trailer section.

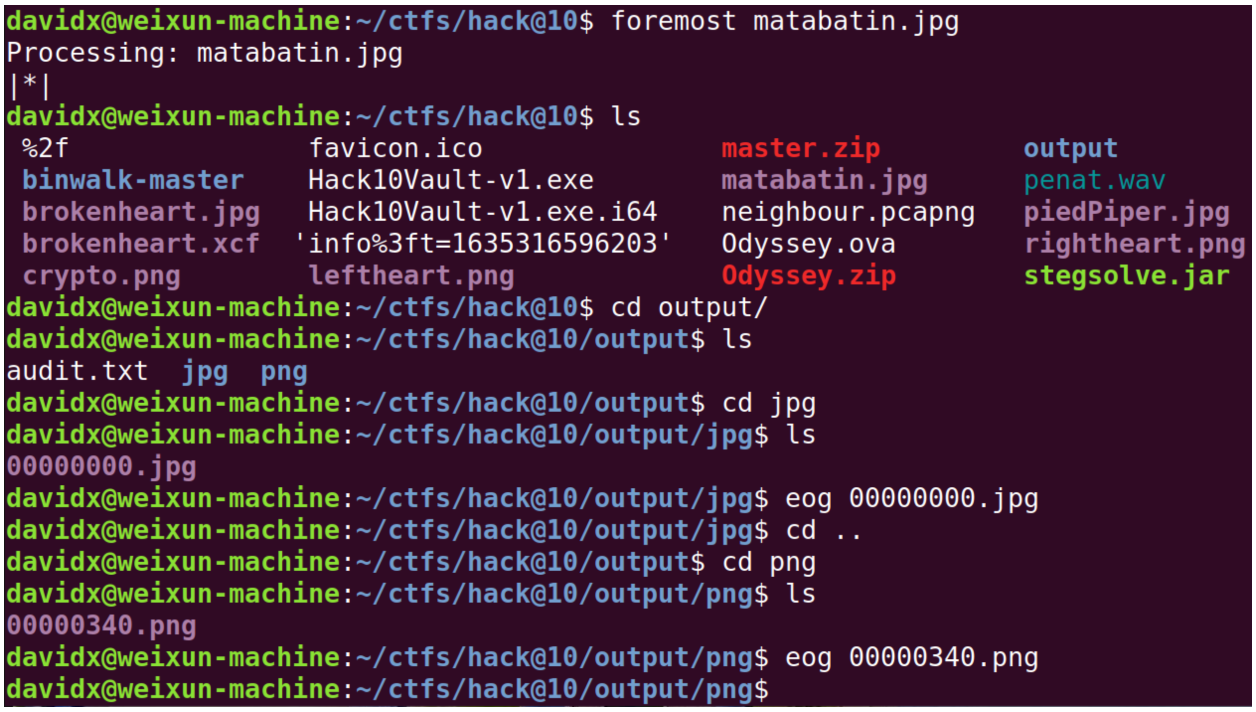


```
            00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
0000000000  ff d8 ff e0 00 10 4a 46 49 46 00 01 01 00 00 01   ......JFIF......
0000000010  00 01 00 00 ff db 00 84 00 05 03 04 09 08 08 08   ................
0000000020  09 09 09 06 08 08 06 08 07 09 08 07 07 08 08 07   ................
0000000030  07 05 07 07 08 07 07 07 08 07 07 07 0a 10 0b 07   ................
0000000040  08 0e 09 07 07 0d 15 0d 0e 11 11 13 13 13 07 0b   ................
0000000050  16 18 16 12 18 10 12 13 12 01 05 05 05 08 07 08   ................
0000000060  0d 08 08 0f 12 0d 0d 0d 12 12 12 12 12 12 12 12   ................
0000000070  12 12 12 12 12 12 12 12 12 12 15 12 15 12 12 12   ................
```



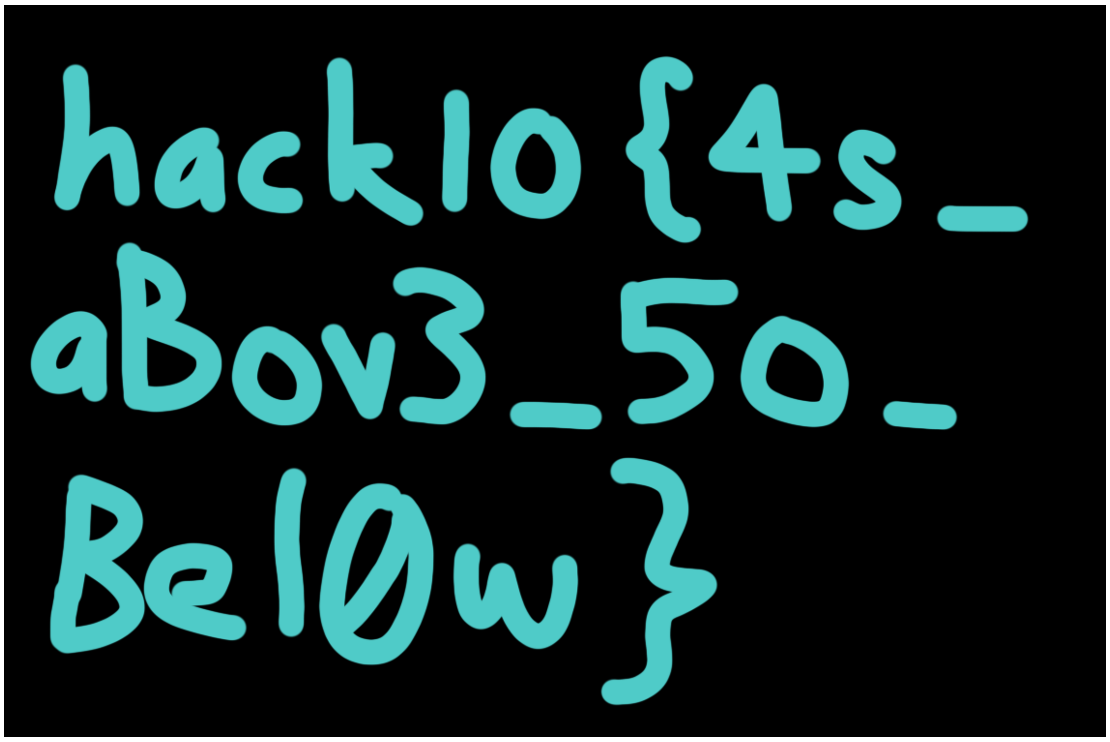
```
            00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
0000045d81  00 80 c9 41 18 23 81 47 98 47 3c 3f de 23 fa 9d   ...A.#.G.G<?.#..
0000045d91  cd 5e 5d 1f 6f 5b 05 95 cc 00 00 00 00 00 00 00   .^].o[..........
0000045da1  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000045db1  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000045dc1  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000045dd1  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000045de1  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000045df1  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000045e01  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000045e11  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000045e21  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000045e31  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000045e41  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000045e51  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000045e61  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000045e71  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000045e81  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000045e91  00 00 00 00 00 00 00 00 a6 f2 ff 01 25 37 a7 93   ............%7..
0000045ea1  63 38 b9 fb 00 00 00 00 49 45 4e 44 ae 42 60 82   c8......IEND.B`.
```

Therefore, we believe that the image file contains embedded data and can be carved out using foremost command.



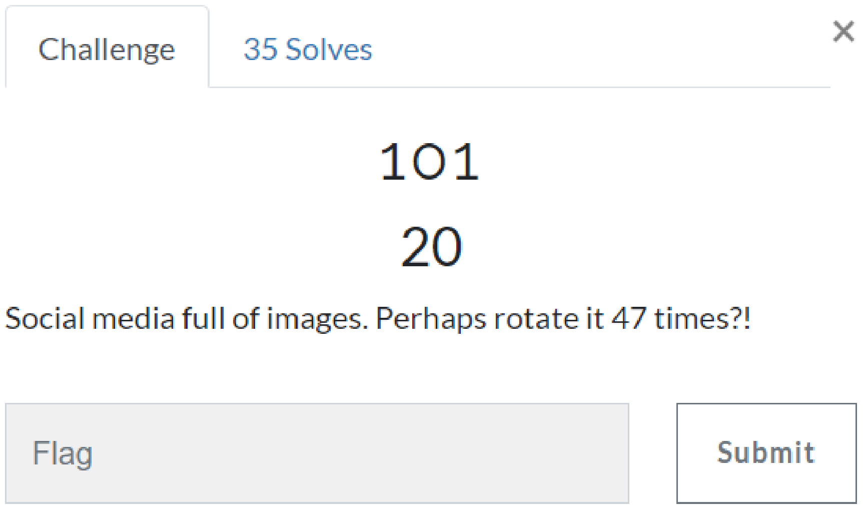
```
davidx@weixun-machine:~/ctfs/hack@10$ foremost matabatin.jpg
Processing: matabatin.jpg
|*|
davidx@weixun-machine:~/ctfs/hack@10$ ls
 %2f                 favicon.ico                 master.zip         output
 binwalk-master      Hack10Vault-v1.exe          matabatin.jpg      penat.wav
 brokenheart.jpg     Hack10Vault-v1.exe.i64      neighbour.pcapng   piedPiper.jpg
 brokenheart.xcf    'info%3ft=1635316596203'     Odyssey.ova        rightheart.png
 crypto.png          leftheart.png               Odyssey.zip        stegsolve.jar
davidx@weixun-machine:~/ctfs/hack@10$ cd output/
davidx@weixun-machine:~/ctfs/hack@10/output$ ls
audit.txt  jpg  png
davidx@weixun-machine:~/ctfs/hack@10/output$ cd jpg
davidx@weixun-machine:~/ctfs/hack@10/output/jpg$ ls
00000000.jpg
davidx@weixun-machine:~/ctfs/hack@10/output/jpg$ eog 00000000.jpg
davidx@weixun-machine:~/ctfs/hack@10/output/jpg$ cd ..
davidx@weixun-machine:~/ctfs/hack@10/output$ cd png
davidx@weixun-machine:~/ctfs/hack@10/output/png$ ls
00000340.png
davidx@weixun-machine:~/ctfs/hack@10/output/png$ eog 00000340.png
davidx@weixun-machine:~/ctfs/hack@10/output/png$
```

After the process is done, a suspicious .PNG file is retrieved, which is our final flag in that case.



FLAG: hack10{4s_aBov3_50_Bel0w}

OSINT: 101

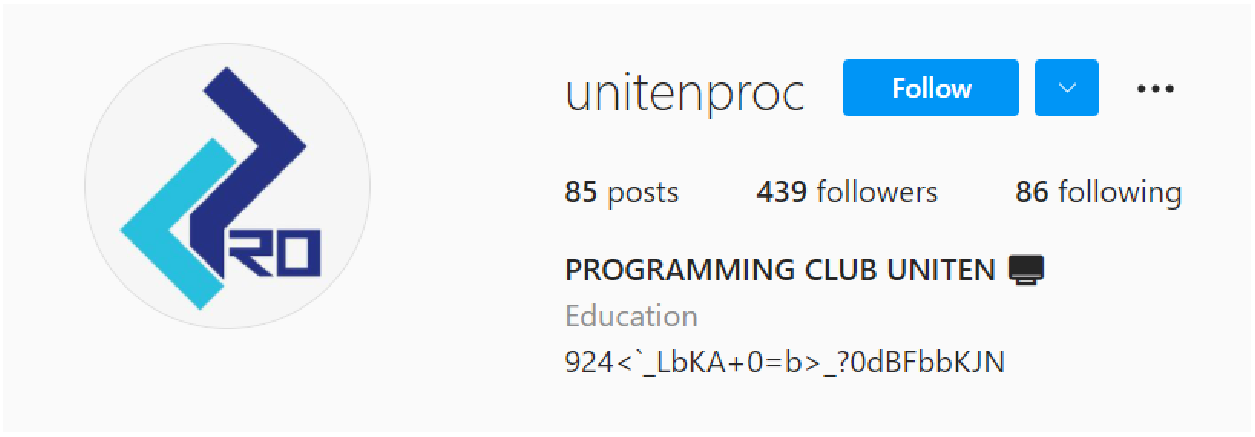


101

20

Social media full of images. Perhaps rotate it 47 times?!

According to the description, it is believed that can we obtain the ciphertext from one of the social media platforms of UniTen.

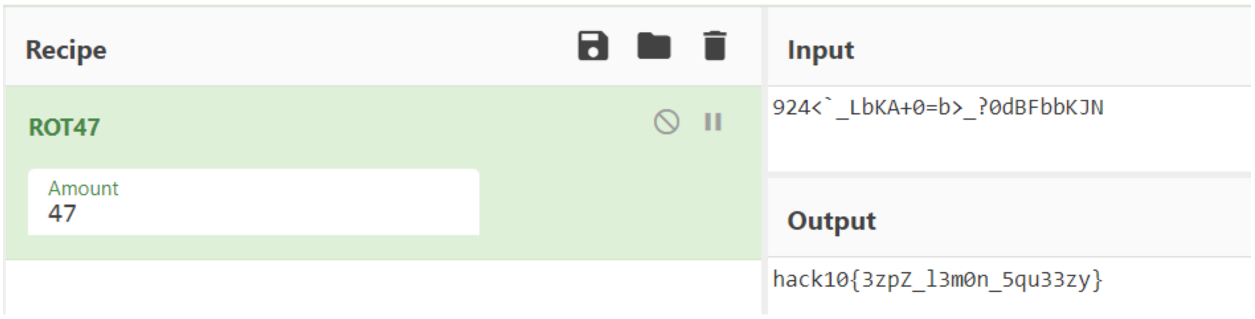


PROGRAMMING CLUB UNITEN

Education

924<`_LbKA+0=b>_?0dBFbbKJN

Based on the profile description and challenge hint, the cipher is processed by rot47. The flag is revealed after decrypting using CyberChef.



```
924<`_LbKA+0=b>_?0dBFbbKJN
```

```
hack10{3zpZ_l3m0n_5qu33zy}
```

FLAG: hack10{3zpZ_l3m0n_5qu33zy}