HACKY EASTER 2019

Contents

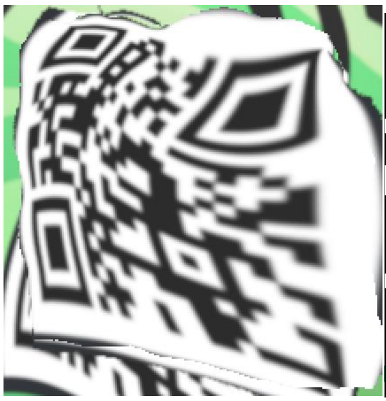
EASY

01: Twisted

Solution

- **TOOLS**: Gimp 2

Using the cage transform to select the QR code, the image was transformed as much as possible to show the QR code. The code was then recreated in excel:



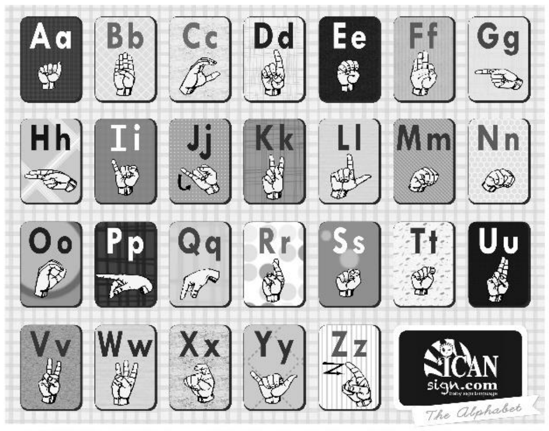
he19-Eihb-UUVw-nObm-lxaW

02: Just Watch

Solution

- **TOOLS**:
  - google
  - Gif splitter: https://ezgif.com/split

First, split the gif into component frames. There are 10 frames in total. These look like sign language:



The gif translates as: givemeasign

`he19-DwWd-aUU2-yVhE-SbaG`

03: Sloppy Encryption

Solution

- **Tools:** Python

The ruby code is a simple encryption algorithm. It takes an input, applies a mathematical algorithm (as cycle) then turns this into base64 encoding; all with a load of jumps forwards and back between asci and hex. We know the b64 output, so this code can be reversed to give the input.

The code is fairly straight forward to reverse, apart from one line:

```
x=ox.to_i(16)*['5'].cycle(101).to_a.join.to_i
```

This code takes a number summing it 101 times: sum(input x 10n)
e.g. input is 123, over 4 cycles:

```
4     1 2 3 0 0 0
3       1 2 3 0 0
2         1 2 3 0
1           1 2 3
      1 3 6 6 5 3
```

Reversing this is possible, if you work from the right had side and remove the lsb from the overall number.

This then gives a number which is divided by 5, then converted to hex, then converted to ASCII. Beware then /5. This gave me problems in python3, as it was rounding the number. So had to run in python2, to get message: **n00b_style_crypto**

he19-YPkZ-ZZpf-nbYt-6ZyD

```
import base64
import math

#this is the one I need to convert
string ="K7sAYzGlYx0kZyXIIPrXxK22DkU4Q+rTGfUk9i9vA60C/ZcQOSWNfJLTu4RpIBy/27yK5CBW+UrBhm0="
# REVERSE OF: b=Base64.encode64(c)
a = base64.b64decode(string)
# REVERSE OF: c=x.to_s(16).scan(/../).map(&:hex).map(&:chr).join
c=""
for x in range(0,len(a)):
    char = hex(a[x])[2:]
    if len(char) < 2:
        char = "0"+char
    c+=char
d = int(c,16)      #where d is the base 10
# REVERSE OF: x=ox.to_i(16)*['5'].cycle(101).to_a.join.to_i
cycle = 101
d_array = []        #break d into component digits
n = []
total = []
digits = []         # holds the new digits
```

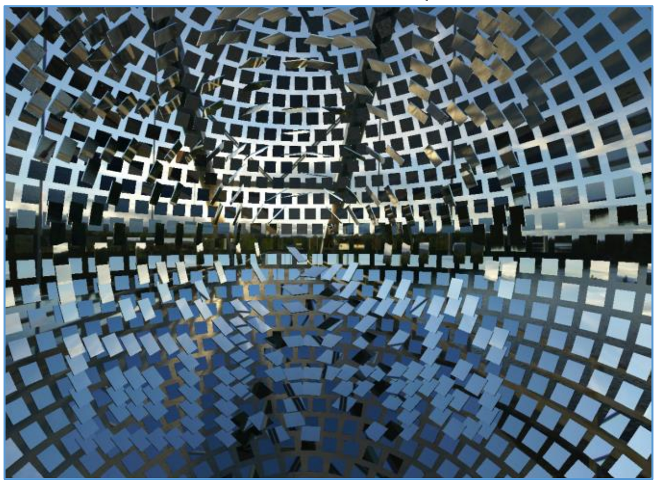
```python
for i in str(d):
    d_array.append(int(i))
    n.append(0)
    total.append(0)
running_total = 0                # holds the running sum of total[n]
pointer = len(d_array)-1         # this points to the digit currently being calculated
d_array.append(0)                # add this so we can work from right to left when calculating

while pointer > 0:
    total[pointer] = running_total
    n[pointer] = d_array[pointer] - running_total
    amount = d_array[pointer]
    while amount - running_total < 0:   #d_array[pointer+1] < 0:
        amount += 10
        d_array[pointer-1] -= 1
    n[pointer] = amount - running_total
    running_total += n[pointer]
    pointer -=1
#now add all digits[i] into a string
number = 0
power = 0
for i in range(len(n)-1,len(n)-1-cycle,-1):
    number += n[i]*(10**power)
    power += 1
number = int(number / 5) #note: this gives issues in python3, python2 number =
37495114530588664796246000071968314913903
hex_array = []
# REVERSE OF: ox='%#X'%h.to_i(16)
hex_nos = hex(int(number)).split("x")[-1]
for i in range(len(hex_nos),0,-2):
    hex_array.append(hex_nos[i-2:i])
# REVERSE OF: h=input.unpack('C'*input.length).collect{|x|x.to_s(16)}.join
hex_array.reverse()
for i in hex_array:
    print(chr(int(i,16)),end="")
```

04: Disco 02

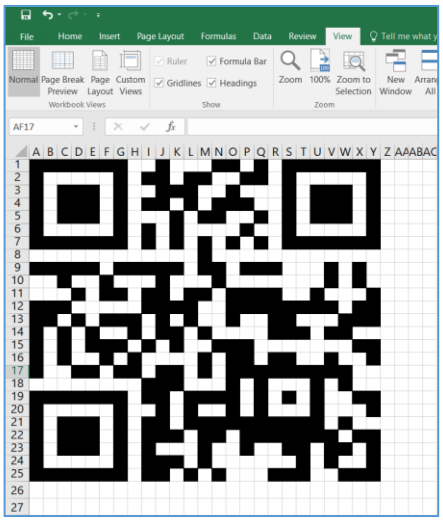
Solution

- **Tools:** excel

Zoom into the ball, and it's full of squares.



Basically I pivoted around the squares and transferred them to excel.



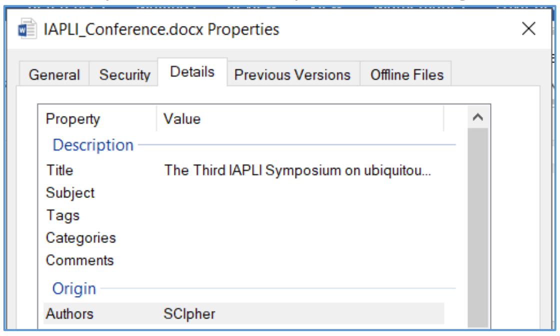
he19-r5pN-YIRp-2cyh-GWh8

05: Call for Papers

Solution

- **Tools:** google

The file metadata gives a hint to SCIpher. This is a cipher for hiding clues in academic papers.



IAPLI_Conference.docx Properties

General | Security | Details | Previous Versions | Offline Files

Property	Value
Description	
Title	The Third IAPLI Symposium on ubiquitou...
Subject	
Tags	
Categories	
Comments	
Origin	
Authors	SCIpher

Paste the document into an online decoder: https://pdos.csail.mit.edu/archive/scigen/scipher.html

Reveals:
https://hackyeaster.hacking-lab.com/hackyeaster/images/eggs/5e171aa074f390965a12fdc240.png

`he19-A6kG-rb9U-Iury-qv93`

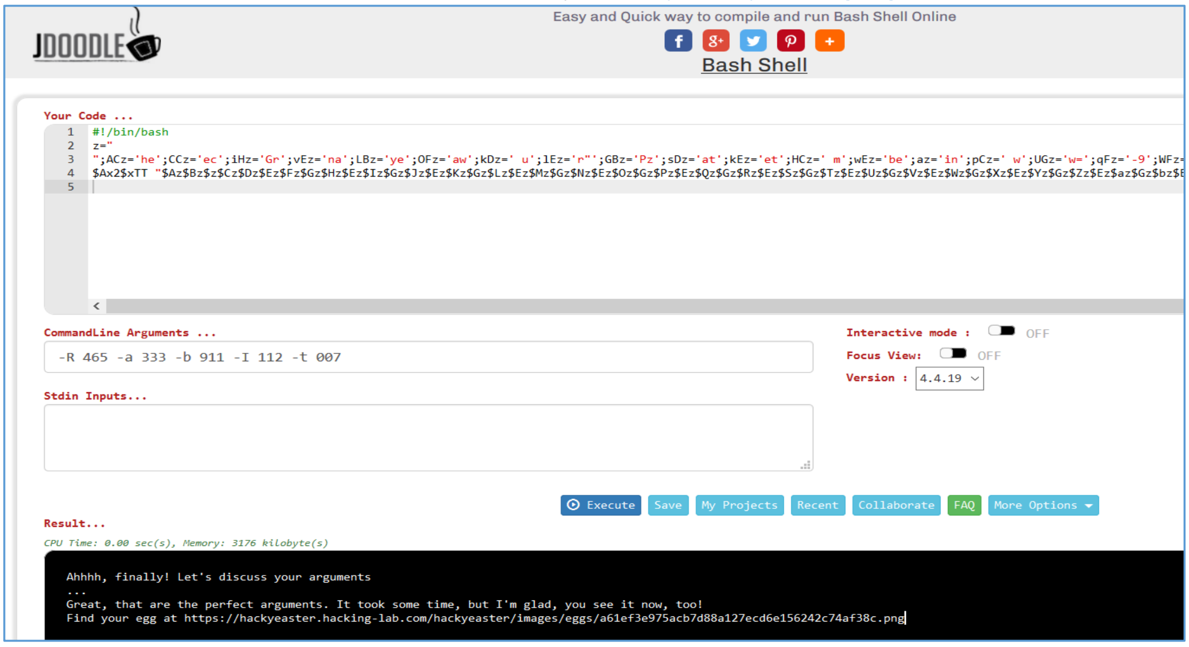
07: Shell we Argument

Solution

- **Tools:** Google

This .sh code provided is nested variables which once expanded reveal some proper bash code. Examining the bash shows you need 10 arguments, which are provided in the bash code as *-R 465 –a 333 –b 911 –I 112 –t 007*

Enter the .sh shell code into an online bash compiler: https://repl.it/languages/bash



Your Code ...

```
#!/bin/bash
z="
";ACz='he';CCz='ec';iHz='Gr';vEz='na';LBz='ye';OFz='aw';kDz=' u';lEz='r"';GBz='Pz';sDz='at';kEz='et';HCz=' m';wEz='be';az='in';pCz=' w';UGz='w=';qFz='-9';WFz=
$Ax2$xTT "$Az$Bz$z$Cz$Dz$Ez$Fz$Gz$Hz$Ez$Iz$Gz$Jz$Ez$Kz$Gz$Lz$Ez$Mz$Gz$Nz$Ez$Oz$Gz$Pz$Ez$Qz$Gz$Rz$Ez$Sz$Gz$Tz$Ez$Uz$Gz$Vz$Ez$Wz$Gz$Xz$Ez$Yz$Gz$Zz$Ez$az$Gz$bz$E
```

CommandLine Arguments ...

```
-R 465 -a 333 -b 911 -I 112 -t 007
```

Interactive mode : OFF

Focus View: OFF

Version : 4.4.19

Stdin Inputs...

Result...

CPU Time: 0.00 sec(s), Memory: 3176 kilobyte(s)

```
Ahhhh, finally! Let's discuss your arguments
...
Great, that are the perfect arguments. It took some time, but I'm glad, you see it now, too!
Find your egg at https://hackyeaster.hacking-lab.com/hackyeaster/images/eggs/a61ef3e975acb7d88a127ecd6e156242c74af38c.png
```

The egg can be found at: https://hackyeaster.hacking-lab.com/hackyeaster/images/eggs/a61ef3e975acb7d88a127ecd6e156242c74af38c.png

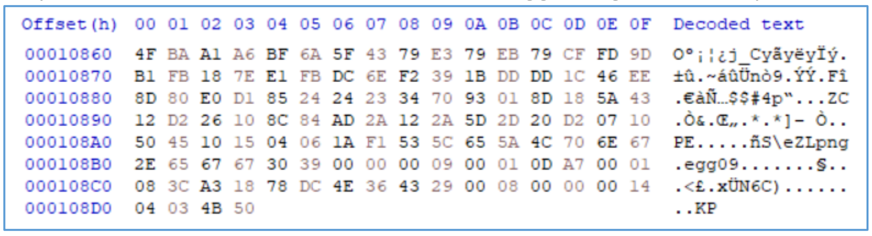
he19-Bxvs-Vno1-9l9D-49gX

09: rorriM rorriM

Solution

- **Tools:**
    - gimp
    - kali

The filename suggests this is a zip file, but in reverse. Opening the file in HxD shows that the end of the Hex is actually the start of the file. The file ends KP, suggesting it is a PK zip file.

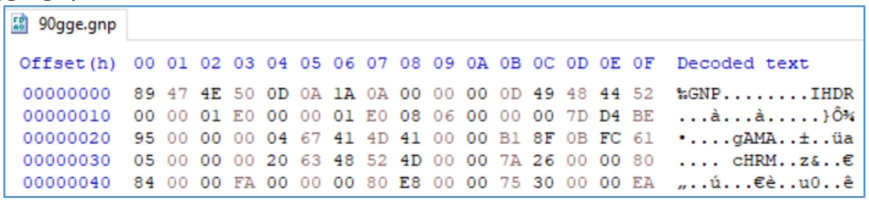


```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

00010860  4F BA A1 A6 BF 6A 5F 43 79 E3 79 EB 79 CF FD 9D  O°¡¦¿j_CyãyëyÏý.
00010870  B1 FB 18 7E E1 FB DC 6E F2 39 1B DD DD 1C 46 EE  ±û.~áûÜnò9.ÝÝ.Fî
00010880  8D 80 E0 D1 85 24 24 23 34 70 93 01 8D 18 5A 43  .€àÑ…$$#4p"...ZC
00010890  12 D2 26 10 8C 84 AD 2A 12 2A 5D 2D 20 D2 07 10  .Ò&.Œ„.*.*]- Ò..
000108A0  50 45 10 15 04 06 1A F1 53 5C 65 5A 4C 70 6E 67  PE.....ñS\eZLpng
000108B0  2E 65 67 67 30 39 00 00 00 09 00 01 0D A7 00 01  .egg09.......§..
000108C0  08 3C A3 18 78 DC 4E 36 43 29 00 08 00 00 00 14  .<£.xÜN6C)......
000108D0  04 03 4B 50                                      ..KP
```

So the entire file needs to be reversed.

1) reverse the file bytes and save it as a .zip:
```
< evihcra.piz xxd -p -c1 | tac | xxd -p -r > file.zip
```
2) extract “90gge.gnp” from the zip:
```
binwalk –Me file.zip
```

Now open 09gge.gnp in HxD.



90gge.gnp

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

00000000  89 47 4E 50 0D 0A 1A 0A 00 00 00 0D 49 48 44 52  ‰GNP........IHDR
00000010  00 00 01 E0 00 00 01 E0 08 06 00 00 00 7D D4 BE  ...à...à.....}Ô¾
00000020  95 00 00 00 04 67 41 4D 41 00 00 B1 8F 0B FC 61  •....gAMA..±..üa
00000030  05 00 00 00 20 63 48 52 4D 00 00 7A 26 00 00 80  .... cHRM..z&..€
00000040  84 00 00 FA 00 00 00 80 E8 00 00 75 30 00 00 EA  „..ú...€è..u0..ê
```

Whilst the file is correct, the start of the file is still wrong. Reverse GNP to PNG and save the file as egg09.png

In gimp:

- invert the egg: colours>invert
- rotate the egg though 180 degrees horizontally: image>transform>flip horizontally

he19-VFTD-kVos-DeL1-lATA

13 Symphony in Hex

Solution

- **Tools:** google for ascii codes: https://ascii.cl/

The notes are delimited to within each bar.
The hint tells you to :

- count quavers
- read semibreves

So reading each bar gives:

Stave 1:	4 8 4 1 4 3 4 B 5 F
Stave 2:	4 D 4 5 4 1 4 D 4
Stave 3:	1 4 4 5 5 5 5 3

As the notes are hex, group them together into 2 and convert to asci:

48 41 43 4B 5F 4D 45 5F 41 4D 41 44 45 55 53
H A C K _ M E _ A M A D E U S

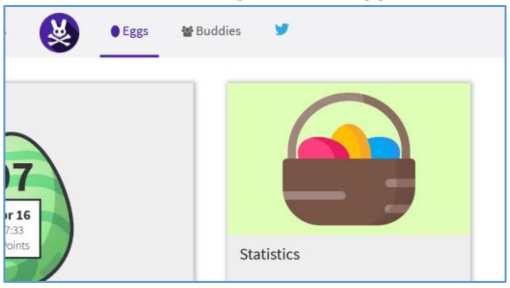
he19-7fEm-jj7g-gpt3-4Mdh

25 - HIDDEN EGG 1

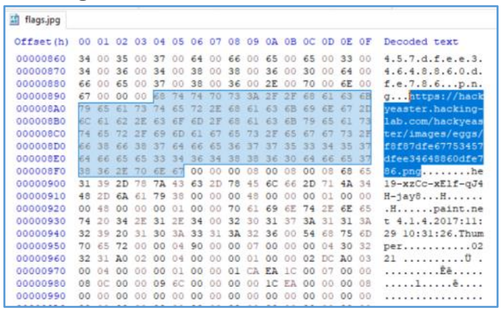
Solution

- **Tools:** HxD

Open up the “Eggs” page, and download the image of the egg basket (flags.jpg).



Open flags.jpg up in HxD and the flag is revealed:

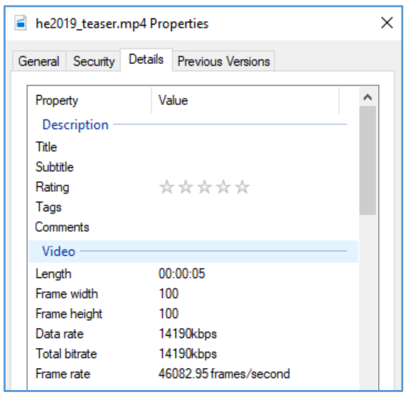


```
flags.jpg

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
00000860  34 00 35 00 37 00 64 00 66 00 65 00 65 00 33 00  4.5.7.d.f.e.e.3.
00000870  34 00 36 00 34 00 38 00 38 00 36 00 30 00 64 00  4.6.4.8.8.6.0.d.
00000880  66 00 65 00 37 00 38 00 36 00 2E 00 70 00 6E 00  f.e.7.8.6...p.n.
00000890  67 00 00 00 68 74 74 70 73 3A 2F 2F 68 61 63 6B  g...https://hack
000008A0  79 65 61 73 74 65 72 2E 68 61 63 6B 69 6E 67 2D  yeaster.hacking-
000008B0  6C 61 62 2E 63 6F 6D 2F 68 61 63 6B 79 65 61 73  lab.com/hackyeas
000008C0  74 65 72 2F 69 6D 61 67 65 73 2F 65 67 67 73 2F  ter/images/eggs/
000008D0  66 38 66 38 37 64 66 65 36 37 37 35 33 34 35 37  f8f87dfe67753457
000008E0  64 66 65 65 33 34 36 34 38 38 36 30 64 66 65 37  dfee34648860dfe7
000008F0  38 36 2E 70 6E 67 00 00 00 08 00 08 00 08 68 65  86.png........he
00000900  31 39 2D 78 7A 43 63 2D 78 45 6C 66 2D 71 4A 34  19-xzCc-xElf-qJ4
00000910  48 2D 6A 61 79 38 00 00 00 48 00 00 00 01 00 00  H-jay8...H......
00000920  00 48 00 00 00 01 00 00 70 61 69 6E 74 2E 6E 65  .H......paint.ne
00000930  74 20 34 2E 31 2E 34 00 32 30 31 37 3A 31 31 3A  t 4.1.4.2017:11:
00000940  32 39 20 31 30 3A 33 31 3A 32 36 00 54 68 75 6D  29 10:31:26.Thum
00000950  70 65 72 00 00 04 90 00 00 07 00 00 00 04 30 32  per...........02
00000960  32 31 A0 02 00 04 00 00 00 01 00 00 02 DC A0 03  21 .........Ü .
00000970  00 04 00 00 00 01 00 00 01 CA EA 1C 00 07 00 00  .........Êê.....
00000980  08 0C 00 00 09 6C 00 00 00 00 1C EA 00 00 00 08  .....l.....ê....
00000990  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

he19-xzCc-xElf-qJ4H-jay8

Teaser

Unzip the file and the mp4 properties tell us this file has a framerate of 46082.95 frames/second:


he2019_teaser.mp4 Properties

Property	Value
Description	
Title	
Subtitle	
Rating	☆☆☆☆☆
Tags	
Comments	
Video	
Length	00:00:05
Frame width	100
Frame height	100
Data rate	14190kbps
Total bitrate	14190kbps
Frame rate	46082.95 frames/second

1) ffmpeg to extract all the frames from the video gives us 230400 files:

```
ffmpeg.exe -i \Users\mark\Desktop\he2019_teaser.mp4 \Users\mark\Desktop\thumb\thumb%06d.jpg
```

2) OpenCV python library can be used to join these files. However the files vary slightly in size from 295KB to 298KB, which means OpenCV does struggle to concatentate them, as the files need to be the same size.

   Again, ffmpeg to the rescue. Resize each of the 230,400 jpegs to 90kb png file. This batch took about 6 hours.

```
for %%a in ("\Users\mark\Desktop\thumb\*.jpg") do ffmpeg -i "%%a" -s 1x1 "\Users\mark\Desktop\smaller\%%~na.png"
```

3) Now all 230,400 pngs need to be re-assembled. A short python script allows this. A bit of trial and error soon shows the final image is 480 x 480.

```python
import cv2
import numpy as np

#create a load of blank files for OpenCV to save to
img1 = cv2.imread('thumb000001.png')
img2 = cv2.imread("thumb000002.png")
vis = np.concatenate((img1, img2), axis=1)
for i in range(0,480):
    cv2.imwrite("output"+str(i)+".png", vis)
cv2.imwrite("final.png", vis)

#main code
for y in range (0,480):
    outfile = "output"+str(y)+".png"
    for x in range(0,480):
        #define the names of input file to concatenate
        infile = "thumb" + str(480+1+x+(480*y)) + ".png"

        #create files
        img1 = cv2.imread(outfile)
        img2 = cv2.imread(infile)
        img = np.concatenate((img1, img2), axis=1)
        cv2.imwrite(outfile, img)

#collate all the outfiles into the final teaser image
for y in range (0,480):
    img1 = cv2.imread("output0.png")
    img2 = cv2.imread("output1.png")
    vis = np.concatenate((img1, img2), axis=0)
    cv2.imwrite("final.png", vis)

    for final in range(0,y):
        img1 = cv2.imread("final.png")
        img2 = cv2.imread("output"+str(final)+".png")
        vis = np.concatenate((img1, img2), axis=0)
        cv2.imwrite("final.png", vis)
```

he19-th1s-isju-5tAt-Eazr