8900 Hacky Easter 2016 Teaser Challenge

by eash

On March 14, the third Hacky Easter competition is going to start!

In order to shorten the waiting time, we are providing a teaser challenge in advance. Download the presentation about Hacky Easter 2016. It shows information about the upcoming event, and gives a glimpse at the event's challenges.

An easter egg is hidden somewhere in the file. Can you find it? You'll need to find hidden content in the file, and perform some operations on it, in order to get the egg.

Requirements

The presentation file.

Goal

1. Find the hidden content in the presentation.
2. Perform the necessary operations on it. You don't need to be a super h4XOR for this!
3. HINT: When extracting things from the presentation, do it from the original file, and not from within PowerPoint!
4. Get the Easter egg

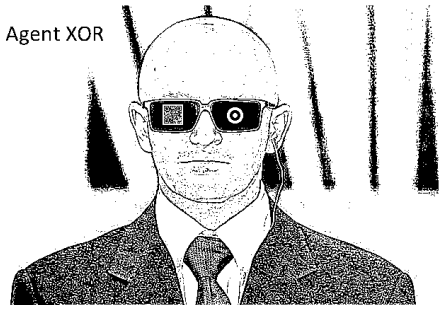
Ok, First of all, I downloaded the required file "HackyEaster2016.odp" from http://media.hacking-lab.com/hackyeaster/HackyEaster2016.odp.

Step 1.

Analyzing HackyEaster2016.odp with exiftool command show one more HINT.
"Initial-creator : Agent Xor"

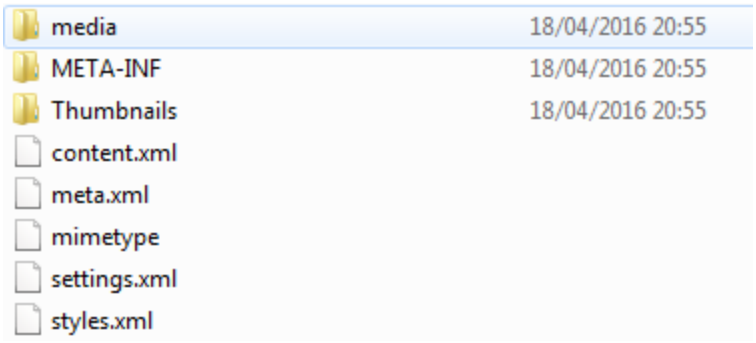
```
File Name                       : HackyEaster2016.odp
Directory                       : .
File Size                       : 5.0 MB
File Modification Date/Time     : 2016:02:15 16:45:28-05:00
File Access Date/Time           : 2016:03:14 09:03:59-04:00
File Inode Change Date/Time     : 2016:02:15 16:45:28-05:00
File Permissions                : rwxrwxrwx
File Type                       : ODP
MIME Type                       : application/vnd.oasis.opendocument.presentation
Generator                       : MicrosoftOffice/14.0 MicrosoftPowerPoint
Title                           : Hacky Easter 2016
Initial-creator                 : Agent Xor
Creator                         : Philipp Sieber
Creation-date                   : 2016:02:01 10:42:30Z
Date                            : 2016:02:02 10:12:37Z
Print-date                      : 2016:02:02 06:52:33Z
Editing-cycles                  : 36
Editing-duration                : PT0SDocument-statistic Paragraph-count: 43
Document-statistic Word-count   : 154
Preview PNG                     : (Binary data 8247 bytes, use -b option to extract)
```

This hint remembered me the Hackyeaster 2015 Challenge 17 named "Spot the Difference".



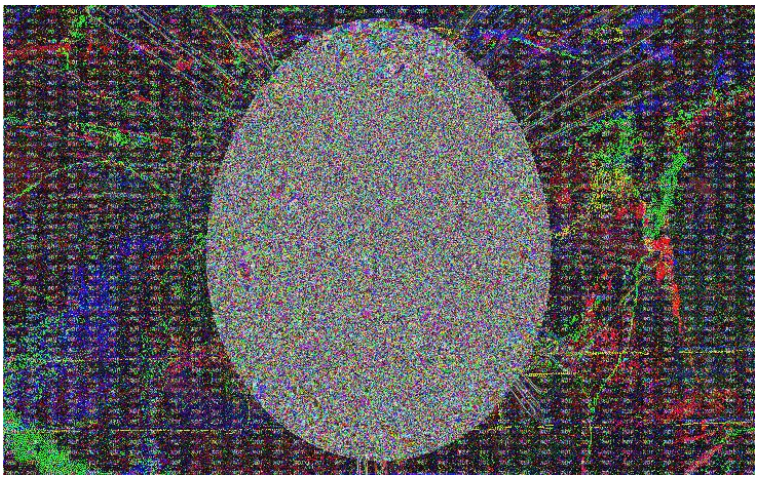
Step 2

Following the HINT number 3 of the Goals, I unzipped the HackyEaster2016.odp content instead off opening it with Powerpoint and get the content below.



Step 2

Quick search in the media folder show me the suspicious image “image36.png”.



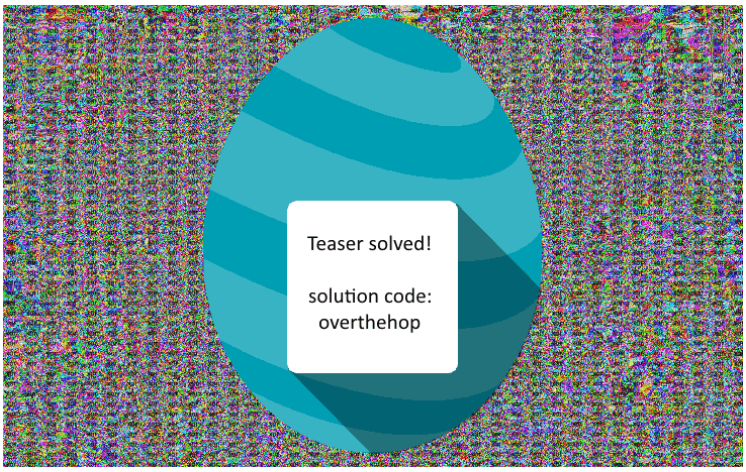
Step3

Now was crystal clear what I need to do. Perform a XOR operation with image36.png and other image to reveal the egg.

Here I spent some time performing a XOR of image36.png with other images up to realize that the trick was perform the XOR operation with all images of the challenges from image36.png up to image12.png cause all those images have the same size, 732x458 pixels.

Step 4

I wrote a script in python to perform the XOR operation and revel the egg below.



One more very creative Teaser challenge.
Congratulations to HL team.

Tks

Eash#

PNG XOR
#Coded by eash#

```
from PIL import Image

img1 = Image.open("media/image36.png").convert("RGBA")
img2 = Image.open("media/image35.png").convert("RGBA")

pixels1 = img1.load() # create the pixel map
pixels2 = img2.load() # create the pixel map

(w,h)=img1.size

# create new image to which we will write hidden image
outimg = Image.new( 'RGB',(732, 458))
pixels_out = outimg.load()

#First XOR
print "Performing XOR on Image 36 and 35"
for y in xrange(h):
    for x in xrange(w):
      r1, g1 , b1, a1 = pixels1[x,y]
      r2, g2 , b2, a2 = pixels2[x,y]
      r_check =  r1 ^ r2
      g_check =  g1 ^ g2
```

```
      b_check =  b1 ^ b2
      a_check =  a1 ^ a2
      pixels_out[x,y] = (r_check,g_check,b_check,a_check)

outimg.save("media/teaser_egg.png","png")

#Looping to XOR rest of the images
for i in range(34,11,-1):
    print "Performing XOR on Image" + str(i)
    img1 = Image.open("media/teaser_egg.png").convert("RGBA")
    img2 = Image.open("media/image"+str(i)+".png").convert("RGBA")
    pixels1 = img1.load() # create the pixel map
    pixels2 = img2.load() # create the pixel map
    for y in xrange(h):
        for x in xrange(w):
            r1, g1 , b1, a1 = pixels1[x,y]
            r2, g2 , b2, a2 = pixels2[x,y]
            r_check =  r1 ^ r2
            g_check =  g1 ^ g2
            b_check =  b1 ^ b2
            a_check =  a1 ^ a2
            pixels_out[x,y] = (r_check,g_check,b_check,a_check)
    outimg.save("media/teaser_egg.png","png")

print "Voila your Egg is done :) at media/teaser_egg.png"
-------------------------------------------------------------------------------------------------
```