

GIF Tutorial

This tutorial uses Python to implement a GIF file analyser.

1 GIF Reader

In the first part of the tutorial we will read the first 16 bytes of GIF files, and analyse their values. For this use the following code to read the file (the `f.read(1)` statement reads one byte at a time):

```
import binascii
with open('timeline.gif', 'rb') as f:
    for x in range(0,15):
        byte_s = f.read(1)
        print byte_s,"[" ,binascii.hexlify(byte_s),"]",
```

Now download the following files, and analyse the first 16 bytes. What information can you gain about the file from these 16 bytes:

```
http://www.asecuritysite.com/log/cat01_with_hidden_text.gif
http://www.asecuritysite.com/log/mini.gif
http://asecuritysite.com/log/file01.gif
```

Normally files have a magic number at the start of the file. Access the following site, and determine the magic numbers for a range of graphics files:

<http://asecuritysite.com/forensics/magic>

Adobe Illustrator (AI):

Bitmap graphic (BMP):

JPEG graphic file (JPG):

JPEG 2000 graphic file (JP2):

GIF graphic file (GIF):

TIF graphic file (TIF):
PNG graphic file (PNG):
Photoshop Graphics (PSD):
Windows Meta File (WMF):
PDF Document (PDF):
Postscript File (PS):

Now modify the program so that it reads 16 bytes of a file and detects each of the graphic file formats given above. You can download samples from:

<http://asecuritysite.com/forensics/magic>

A starting point for a program might be:

```
import binascii

with open('timeline.gif', 'rb') as f:
    byte_s = f.read(16)
    if ((byte_s[0]=='G') and (byte_s[1]=='I')): print "It's a GIF"
```

2 GIF Analyser

We can use a Python library to read the full contents of a GIF file, and analyse its content. For this implement the Python code given in:

<http://www.asecuritysite.com/comms/gif>

```
import sys
import os
from PIL import Image

file1 = "mini.gif"

if (len(sys.argv)>1):
    file1=str(sys.argv[1])

im = Image.open(file1)

if im.tile[0][0] == "gif":
    tag, (x0, y0, x1, y1), offset, extra = im.tile[0]
```

```

im.size = x1-x0, y1-y0
im.tile = [(tag, (0, 0) + im.size, offset, extra)]
print "X0:\t",x0
print "Y0:\t",y0
print "X1:\t",x1
print "Y1:\t",y1
print "Offset:\t",offset
print "Extra:\t",extra
print "Image size:\t",im.size
print "Image title:\t",im.tile

nframes = 0
while im:
    nframes += 1
    try:
        im.seek( nframes )
    except EOFError:
        break;
print "No of frames:\t",nframes

```

Modify the code so that it imports the files specified in Table 1, and from this complete the table.

Table 1: GIF Analysis

Image	Pixel width and height	First colour in the table (3 byte hex value)	Number of frames
http://www.asecuritysite.com/log/cat01_with_hidden_text.gif			
http://www.asecuritysite.com/log/mini.gif			
http://asecuritysite.com/log/file01.gif			

3 Colour convertor

The main colour formats are RGB, YUV/YCbCr and CMYK. One conversion for JPEG images is:

$$\begin{aligned}
 Y &= R * 0.29900 + G * 0.58700 + B * 0.11400 \\
 Cb &= R * -0.16874 + G * -0.33126 + B * 0.50000 + 128 \\
 Cr &= R * 0.50000 + G * -0.41869 + B * -0.08131 + 128 \\
 \\
 R &= Y + \hspace{10em} + (Cr - 128) * 1.40200 \\
 G &= Y + (Cb - 128) * -0.34414 + (Cr - 128) * -0.71414 \\
 B &= Y + (Cb - 128) * 1.77200
 \end{aligned}$$

Implement a Python program in which the user can enter values of YCbCr, and the program converts to the equivalent RGB value. Once complete, modify the program so that it will convert from RGB into YCbCr.

Outline six of your conversions:

In the following code, we separate the Red, Green and Blue values, and can display their intensity:

http://asecuritysite.com/comms/face_yuv

Copy the Python code, and download some of the faces included on the page.

What observations can you make on the intensity levels of red, green and blue on the faces: