Happy_puzzle

发表于 2021-01-04 分类于 Challenge， 2019， UNCTF， Misc
Challenge | 2019 | UNCTF | Misc | Happy_puzzle

点击此处获得更好的阅读体验

题目考点

- *PNG格式*

PNG便携式网络图形是一种无损压缩的位图片形格式，其设计目的是试图替代GIF和TIFF文件格式，同时增加一些GIF文件格式所不具备的特性。PNG使用从LZ77派生的无损数据压缩算法，一般应用于JAVA程序、网页或S60程序中，原因是它压缩比高，生成文件体积小。

解题过程

下载题目到本地，打开压缩包发现很多data文件如下图所示：

由文件夹中的info.txt可知，这些data数据块是由图片格式为400 X 400的png图片拆卡得到的，分析`*.data (10240 * N + 5214)`，推测这些data是IDAT 数据块，编写脚本将数据块组合到一起，部分脚本如下图所示：

exp.py

```
#!/usr/bin/env python2
# -*- coding:utf-8 -*-
"""
    Author : Virink <virink@outlook.com>
    Date   : 2019/08/28, 18:00
"""
import os
import sys
import binascii
import zlib
OUTPUT = 'puzzle'
def bin2hex(data):
    return binascii.b2a_hex(data)
def hex2bin(data):
    return binascii.a2b_hex(data)
def dec2bin(data, l=1):
    l = l / 2
    if l == 4:
        return hex2bin("%08x" % int(data))
    else:
        return hex2bin("%02x" % int(data))
def bin2dec(data):
    return int(bin2hex(data), 16)
def crc32(chunkType, chunkData):
    return dec2bin(binascii.crc32(chunkType + chunkData), 8)
def genIHDR(w, h):
    width = dec2bin(w, 8)
    height = dec2bin(h, 8)
    bits = dec2bin(8)
    color_type = dec2bin(2)
    compr_method = filter_method = interlace_method = dec2bin(0)
    chunkData = width+height+bits+color_type + \
        compr_method+filter_method+interlace_method
    res = dec2bin(len(chunkData), 8)+b'IHDR' + \
        chunkData+crc32(b'IHDR', chunkData)
    print([res])
    return res
def genIDAT(data):
    _c = zlib.crc32(b'IDAT'+data)
    if _c < 0:
        _c = ~_c ^ 0xffffffff
    _crc = dec2bin(_c, 8)
    return dec2bin(len(data), 8) + b'IDAT' + data + _crc
def merge_png(width, height, names, output="tmp.png"):
    header = hex2bin("89504E470D0A1A0A")
    ihdr = genIHDR(width, height)
    idat = []
    for name in names:
        f=open("%s/%s" % (OUTPUT, name),'rb')
        data = f.read()
        idat.append(genIDAT(data))
        f.close()
    idat = b''.join(idat)
    iend = hex2bin("00000000" + "49454E44" + "AE426082")
    with open(output, 'wb') as f:
        f.write(header+ihdr+idat+iend)
if __name__ == '__main__':
    fs = ["blczioav.data", "ciaoxptf.data", "csizrgxn.data", "dwelszrk.data", "fhnkotmb.data", "fkjhepcs.data", "gpiuezjw.data", "hbctmwqj.data", "jlxphwfm.data", "jrbiznkl
          "mrxtfkzj.data", "oaeqnubi.data", "pyusgabf.data", "rnydeiho.data", "tihzkoyu.data", "uilqywot.data", "uozjmdnl.data", "wgkapjbh.data", "xufbyndk.data", "xufnmac
    merge_png(400, 400, ['yvxmeawg.data',
                         'rnydeiho.data','uozjmdnl.data',
                         "fhnkotmb.data",'jlxphwfm.data',
                         'yscijlzx.data','ciaoxptf.data',
                         'blczioav.data','jtxsbevz.data',
                         'lstjobzi.data','pyusgabf.data',
                         'wgkapjbh.data','xufbyndk.data',
                         'csizrgxn.data','oaeqnubi.data',
                         'gpiuezjw.data','tihzkoyu.data',
                         'hbctmwqj.data','ycqzmbrw.data',
                         'fkjhepcs.data','kczwtlrd.data',
                         'dwelszrk.data','uilqywot.data',
                         'xufnmacj.data','jrbiznkl.data',
                         'mrxtfkzj.data'], "%s.png" % "flag")
    # for f in fs:
    #     merge_png(400, 400, ['yvxmeawg.data',
    #                          'rnydeiho.data','uozjmdnl.data',
    #                          "fhnkotmb.data",'jlxphwfm.data',
    #                          'yscijlzx.data','ciaoxptf.data',
    #                          'blczioav.data','jtxsbevz.data',
    #                          'lstjobzi.data','pyusgabf.data',
    #                          'wgkapjbh.data','xufbyndk.data',
    #                          'csizrgxn.data','oaeqnubi.data',
    #                          'gpiuezjw.data','tihzkoyu.data',
    #                          'hbctmwqj.data','ycqzmbrw.data',
    #                          'fkjhepcs.data','kczwtlrd.data',
    #                          'dwelszrk.data','uilqywot.data',
    #                          'xufnmacj.data','jrbiznkl.data',
    #                          'mrxtfkzj.data',f], "%s.png" % f)
```

最后逐个数据块测试 HEADER + IHDR + IDAT1 [+IDAT2...]，详细exp.py，一个一个测试可以看到已经拼出得图像，如下图所示，名称为yvxmeawg.data，在第一张的基础上往后去试第二张，以此类推：

最后复原完成的效果如下图所示

Flag

```
unctf{312bbd92c1b291e1827ba519326b6688}
```

- **本文作者：** CTFHub
- **本文链接：** https://writeup.ctfhub.com/Challenge/2019/UNCTF/Misc/a5g4y76Ns3Pf1n4pZcspfN.html
- **版权声明：** 本博客所有文章除特别声明外，均采用 BY-NC-SA 许可协议。转载请注明出处！

Challenge # Misc # 2019 # UNCTF
EasyBox
Think