baby 杯 CTF writeup

Frankss

包含题目：
WEB：第一题 baby_captcha
MISC

、

web

baby_captcha:

推测音频是脚本拼接，查看网页源码发现音频的 base64 正好可以被很多大段的 ICAg 和 AAAA

CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgI

分隔成正好 8 份，推测每一份代表一个验证码，听了三次后判断推测正确，在分割出的每一部分

2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FmlRbfwiXqqX5x8xekZjXDO3p
2F%2FsSq%2Bllq9uKpIDLaGipxynpvem5dSSKHq158oXIHlxYeo51JVgWm9XW%2Bxt%2Bv
2F7tGDAAAoGfkYhoRNwSIT46AwiWglAkxlmhElBIidjoDEJsDbI3WKoY%2FlleqJIgkk%2F%2BE%2FBujfU6kK
2BLZtZlxraa99NGXo9nu7r7e0auPN3LQsjQdrS7e5x6AzCNgxS6C9bsoBrlvTAillLHbGteW6beT7P%2BrTf6JVkl
2F8PT5ZL6Z%2FMellRE8ft6tI39uzkXut7LvxObVNm69iv77Lelrelbd1Zlr%2Buelmt4JVyHLUjeOotY
2FRQxv9rk32Uoqle7rI2UUUT%2F4RYT5IWq055zmn3I0sUtVv3bl7URNkvpfV3S
2FX669S69p9metPjNSOotLKGretdgpW51CLGWv07W9GPXt1XP19V6%2Fu%2B%2Flej%2F%2F%2F%2F%2F%2F%2F%2F%2F
2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F
2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F

任取中间一段作为匹配特征码即可，故伪代码

```
raw = mp3_bas64.split("ICAg.......ICAg")
code=""
for part in raw:
    for pat in pattern_dictionary:
        if pat in part:
            code+=pattern_dictionary[pat]
```

转换出验证码，写脚本爆破即可

MISC

babyLSB

zsteg -a 一把梭，发现

```
bl,g,msb,YX          text: "LoaYuBeMnehSieW{wohsftc\n.htaeD ot nwonknU\n.se
dalb dnasuoht a revo detaerc evah I\n.do"
```

反转

```
>>> "LoaYuBeMnehSieW{wohsftc\n.htaeD ot nwonknU\n.sedalb dnasuoht a revo detaerc evah I"[::-1]
'I have created over a thousand blades.\nUnknown to Death.\nctfshow{WeiShenMeBuYaoL'
```

得到

ctfshow{WeiShenMeBuYaoL 根据图片内容直接猜是不要 LSB，即

ctfshow{WeiShenMeBuYaoLSB} 提交发现 correct

天书奇谭 PLUS-misc

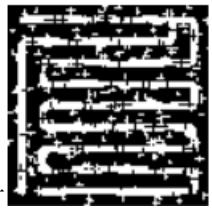
辨认九叠篆可以想象把文字拉直，那么前三个字文字特征比较明显，可以直接辨认出了 “人美歌”，根据出题人信息和最后一个字合理推测是阿狸“人美歌甜”，亦或原图片得到

参考 https://blog.csdn.net/liyuqian199695/article/details/53888254 的差分矩阵求均值，脚本

```python
import cv2
import numpy as np
from PIL import Image


def showpiclocation(img, findimg):
    l = []
    w = img.shape[1]
    h = img.shape[0]
    fw = findimg.shape[1]
    fh = findimg.shape[0]
    for now_h in range(0, h, 100):
        for now_w in range(0, w, 100):
            comp_tz = img[now_h:now_h + fh, now_w:now_w + fw, :] - findimg
            if abs(np.mean(comp_tz)) < 39:
                print("ok")
                print(now_h/100+1)
                print(now_w/100+1)
                l.append((now_w, now_h))
    return l


im = Image.open("solved.bmp")
tmp = im.crop((0, 0, 100, 100))
# tmp = im.crop((104, 0, 204, 100))
# tmp = im.crop((208, 0, 308, 100))
# tmp = im.crop((312, 0, 412, 100))
tmp.save('flag2.png')

img = cv2.imread('flag.png')
r = showpiclocation(img, cv2.imread('flag2.png'))
```

```
if r[0] is not None:
    for res in r:
        image_clip = img[res[1]:res[1] + 200, res[0]:res[0] + 200]
        cv2.namedWindow('img')
        cv2.imshow('img', image_clip)
        cv2.waitKey(0)
```

逐个尝试均值的阈值在 35 到 45 之间即可得到最优匹配，手动确认输出的图片即可

找到的图片  阈值限制`if abs(np.mean(comp_tz)) < 39:`

美丽的小姐姐

pngcheck 发现提前分离，应该是有未显示的部分

少有修改图片宽度的题目，先尝试直接修改图片高度，高度改很大也不会影响图片显示：

```
)1 4f  00 00  01 b8  08 06  00
)0 00  01 73  52 47  42 00  ae
57 41  4d 41  00 00  b1 8f  0b
```

(01B8=440)修改为

```
50 4e 47  0d 0a 1a 0a  00 00 00 0d  49 48 44
00 01 4f  00 00 03 b8  08 06 00 00  00 5e 68
00 00 00  01 73 52 47  42 00 ae ce  1c e9 00
04 67 41  4d 41 00 00  b1 8f 0b fc  61 05 00
```

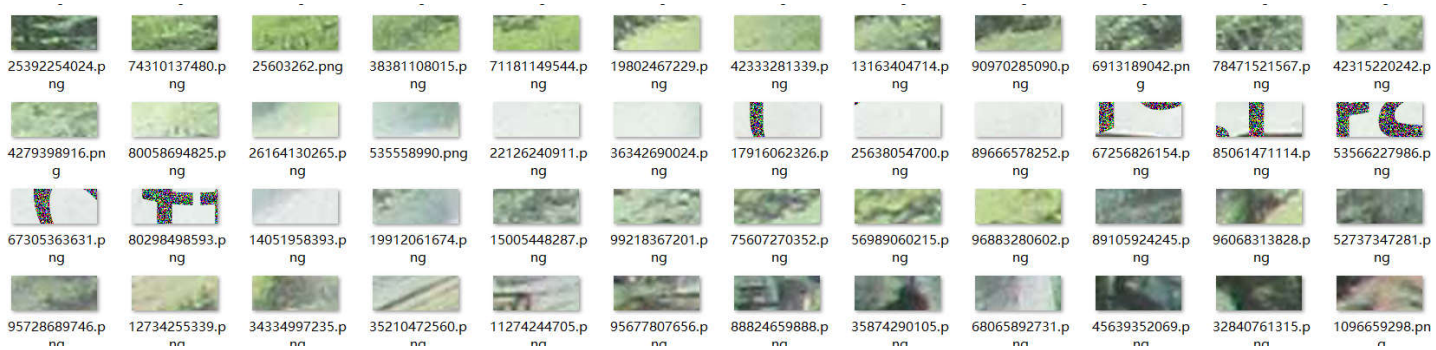
保存发现 flag

万里长城（非预期-根据文件生成时间排序）

题目给了一万张图片，gaps 是跑不动的。从数据量来看文件应该有规律，这应该不是一道暴力一天内可以搜出来的题。

遂尝试爆文件名的规律，统计每位数字出现次数，失败。

尝试按照最后编辑时间排序，发现结果竟然意外的整齐



于是用 os.path.getmtime(filePath) 获取精确到毫秒的修改时间，set()之后发现有 8000 余个结果，也就是只有 1000 张左右图片顺序可能左右是错误的，大胆根据生成时间生成图片。

在上边的预览中，相似图片大概每隔 100 张出现，推测原图宽是 100*99，生成脚本：

```python
import os
import cv2 as cv2
import numpy as np


def get_FileCreateTime(filePath):
    t = os.path.getmtime(filePath)
    return t


if __name__ == '__main__':
    d = os.listdir('random')
    c = []
    for i in d:
        c.append(str(get_FileCreateTime('random\\' + i)) + '--random\\' + i)
    c.sort()

    images = []
    for name in c:
        image = cv2.imread(name.split("--")[1])
        images.append(image)

    blank_image = np.zeros((40 * 100, 99 * 100, 3), np.uint8)
    for i in range(0, 100):
        for j in range(0, 100):
            blank_image[i * 40:i * 40 + 40, j * 99:j * 99 + 99] = images[i * 100 + j][0:40,
0:99]
    cv2.namedWindow('img')
    cv2.imshow('img', blank_image)
    cv2.waitKey(0)
    cv2.imwrite("flag.png", blank_image)
```

得到的结果如图

有些位置误差，但足够辨认出 flag。

五子棋

开两个窗口互博即可

不问天

将每个(5*k, 5*k)的像素点的值复制到(5*k-5, 5*k-5, 5*k, 5*k)的区域，可以看到提示

Pass: BVnumber

得到 binwalk 出压缩包的密码，解压后将 [^\x00-\xff]替换为 1，空格替换为 0，补全后 7 位一组转得到 flag