【愚公系列】2022年01月 攻防世界-进阶题-MISC-76(warmup)

原创

愚公搬代码 已于 2022-01-30 17:06:48 修改 8504 收藏 1
分类专栏： # CTF-攻防世界-MISC 文章标签： 前端 安全 web安全
于 2022-01-29 22:40:19 首次发布

本文链接：https://blog.csdn.net/aa2528877987/article/details/122738636
版权

CTF-攻防世界-MISC 专栏收录该内容
98 篇文章 0 订阅
订阅专栏

文章目录

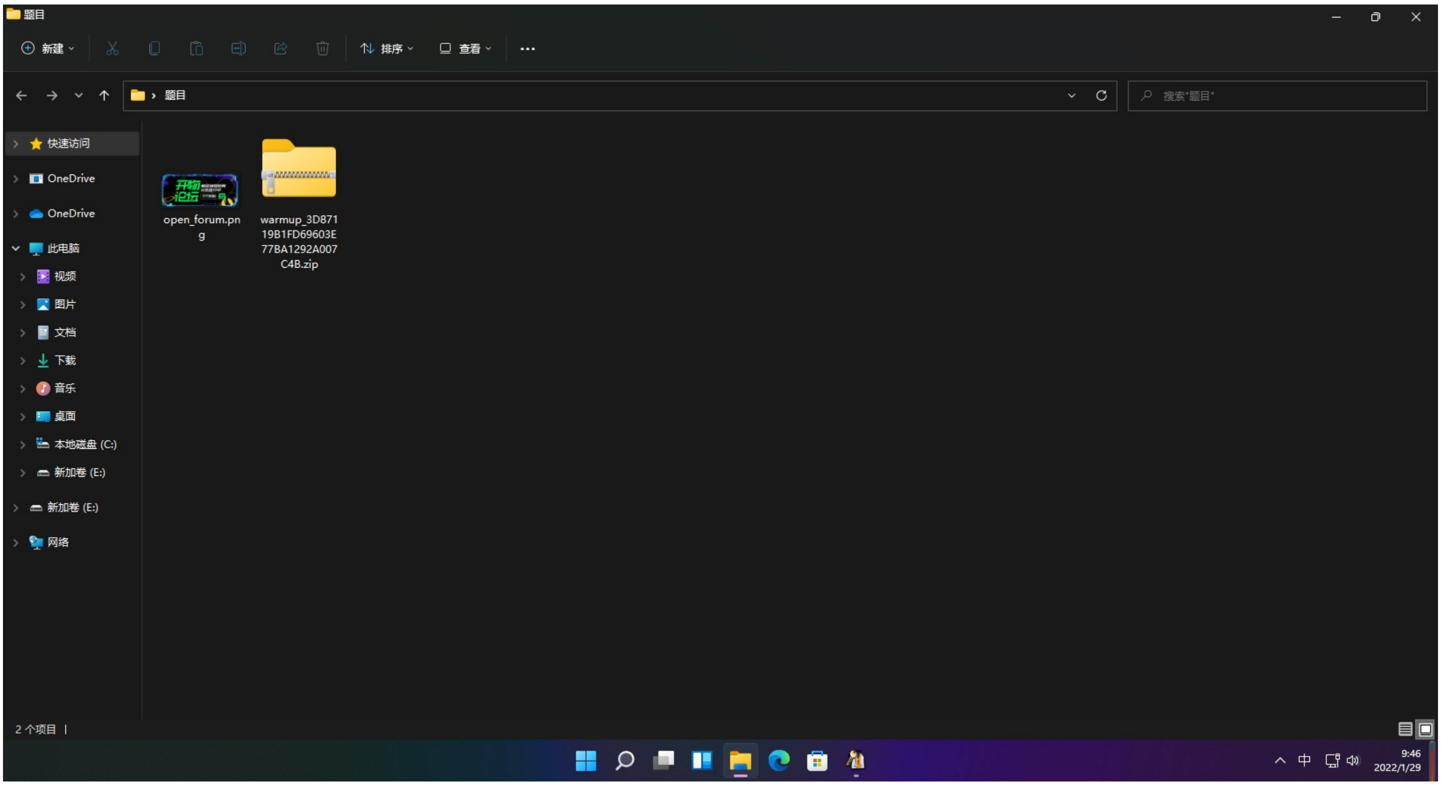
一、warmup

题目链接：https://adworld.xctf.org.cn/task/task_list?type=misc&number=1&grade=1&page=4

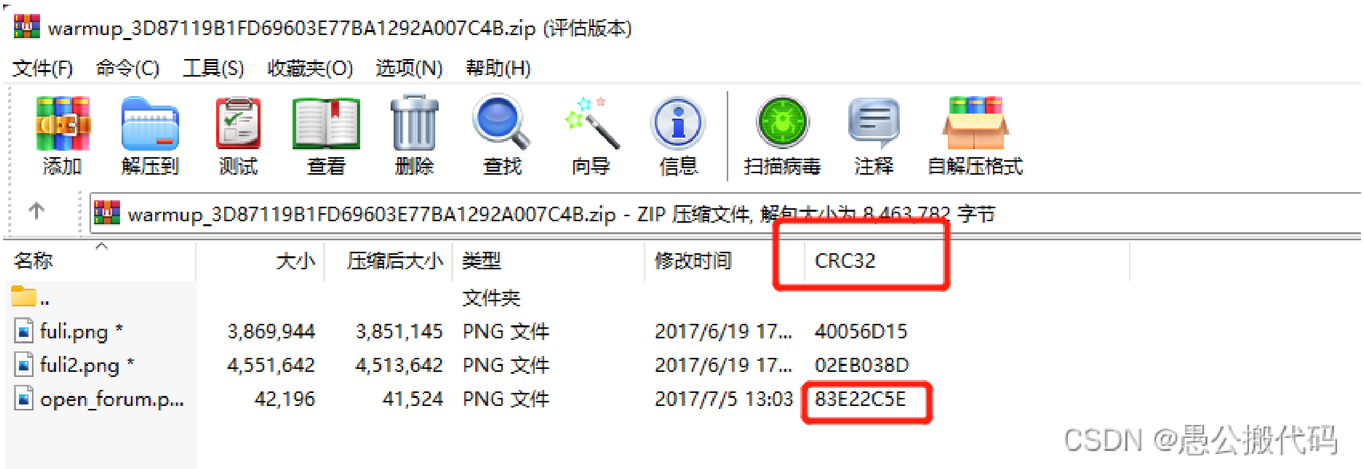
二、答题步骤

1.下载附件

拿到题目，发现有一个图片和一个压缩包，尝试打开压缩包发现需要密码，因此猜测图片即为压缩包的明文。

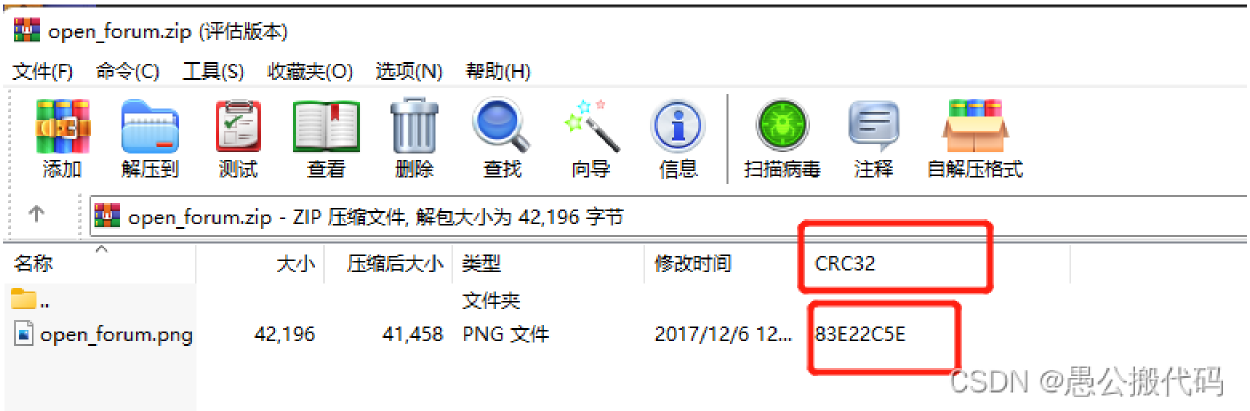


用WINRAR压缩png图片, 进行两个raar进行明文crc验证，用WINRAR软件打开两个文件对比cec32


warmup_3D87119B1FD69603E77BA1292A007C4B.zip (评估版本)
文件(F) 命令(C) 工具(S) 收藏夹(O) 选项(N) 帮助(H)
添加
解压到
测试
查看
删除
查找
向导
信息
扫描病毒
注释
自解压格式

warmup_3D87119B1FD69603E77BA1292A007C4B.zip - ZIP 压缩文件, 解包大小为 8,463,782 字节

名称	大小	压缩后大小	类型	修改时间	CRC32
..			文件夹		
fuli.png *	3,869,944	3,851,145	PNG 文件	2017/6/19 17...	40056D15
fuli2.png *	4,551,642	4,513,642	PNG 文件	2017/6/19 17...	02EB038D
open_forum.p...	42,196	41,524	PNG 文件	2017/7/5 13:03	83E22C5E


open_forum.zip (评估版本)
文件(F) 命令(C) 工具(S) 收藏夹(O) 选项(N) 帮助(H)
添加
解压到
测试
查看
删除
查找
向导
信息
扫描病毒
注释
自解压格式

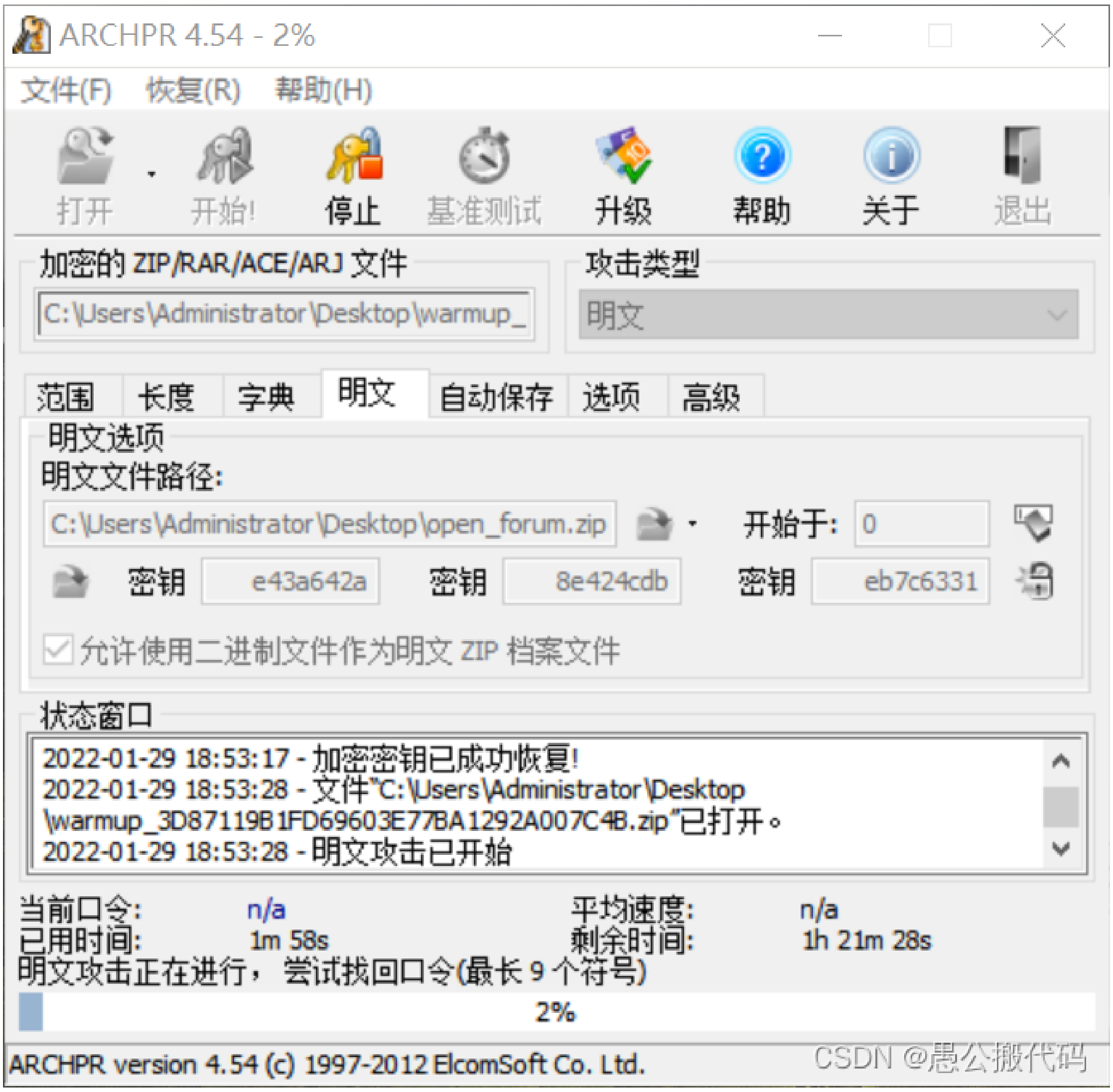
open_forum.zip - ZIP 压缩文件, 解包大小为 42,196 字节

名称	大小	压缩后大小	类型	修改时间	CRC32
..			文件夹		
open_forum.png	42,196	41,458	PNG 文件	2017/12/6 12...	83E22C5E

发现crc32一样可以进行明文工具，因为此处要使用ARCHPR进行压缩包的破解工作

2.ARCHPR

使用ARCHPR进行明文攻击



加密的 ZIP/RAR/ACE/ARJ 文件: C:\Users\Administrator\Desktop\warmup_

攻击类型: 明文

明文选项

明文文件路径: C:\Users\Administrator\Desktop\open_forum.zip

开始于: 0

密钥 e43a642a 密钥 8e424cdb 密钥 eb7c6331

☑ 允许使用二进制文件作为明文 ZIP 档案文件

状态窗口

```
2022-01-29 18:53:17 - 加密密钥已成功恢复!
2022-01-29 18:53:28 - 文件"C:\Users\Administrator\Desktop
\warmup_3D87119B1FD69603E77BA1292A007C4B.zip"已打开。
2022-01-29 18:53:28 - 明文攻击已开始
```

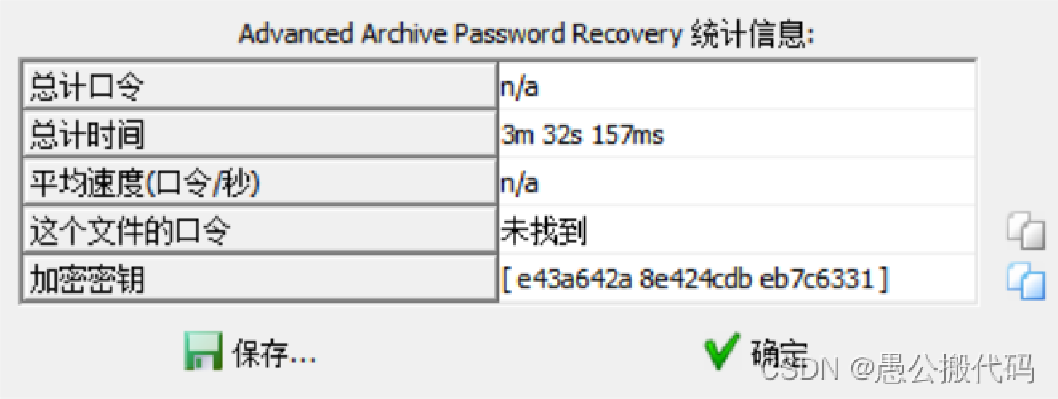
当前口令: n/a 平均速度: n/a

已用时间: 1m 58s 剩余时间: 1h 21m 28s

明文攻击正在进行, 尝试找回口令(最长 9 个符号)

2%

ARCHPR version 4.54 (c) 1997-2012 ElcomSoft Co. Ltd.

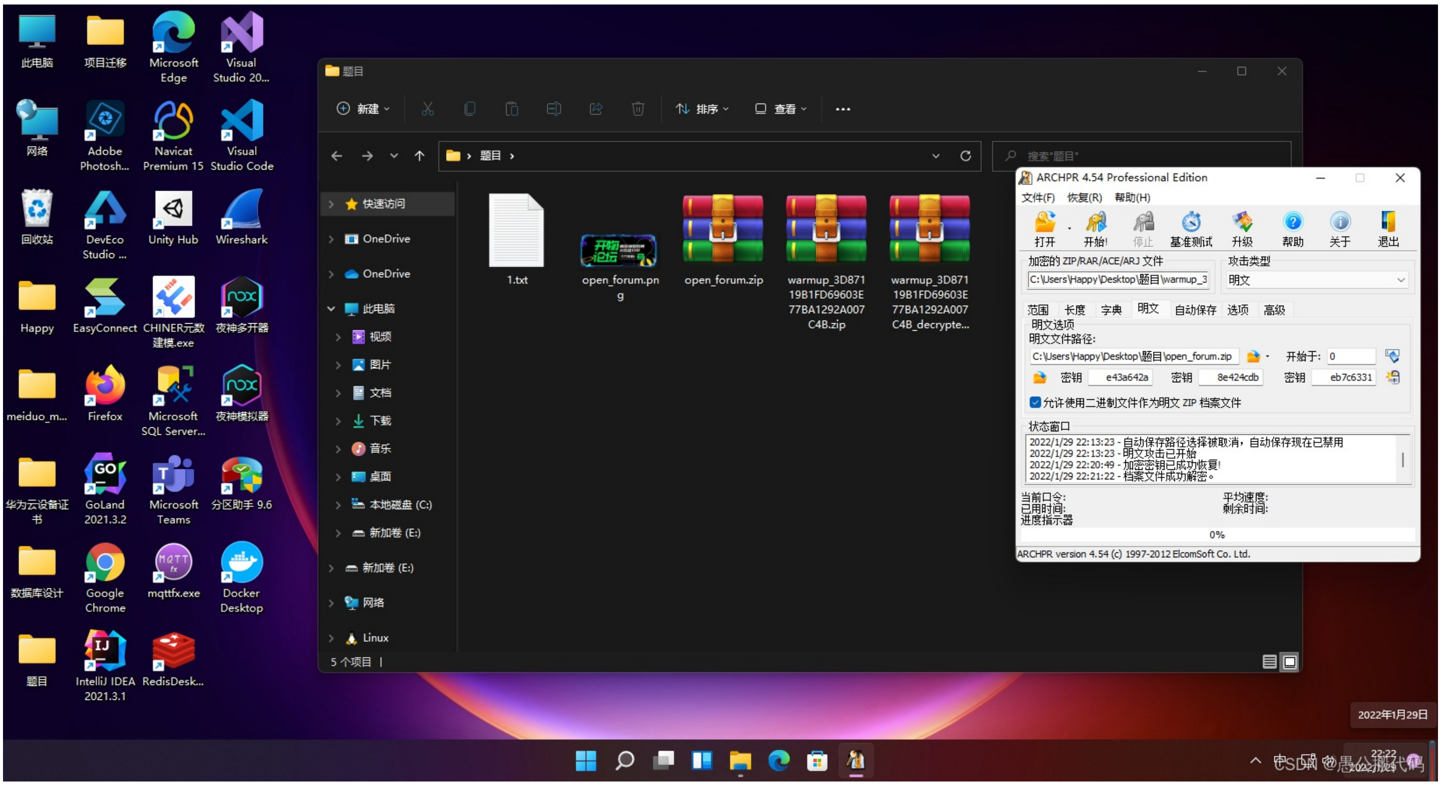


Advanced Archive Password Recovery 统计信息:

总计口令	n/a
总计时间	3m 32s 157ms
平均速度(口令/秒)	n/a
这个文件的口令	未找到
加密密钥	[e43a642a 8e424cdb eb7c6331]

```
Advanced Archive Password Recovery 统计信息:
加密的 ZIP/RAR/ACE/ARJ 文件: C:\Users\Administrator\Desktop\warmup_3D87119B1FD69603E77BA1292A007C4B.zip
总计口令: n/a
总计时间: 3m 32s 157ms
平均速度(口令/秒): n/a
这个文件的口令 : 未找到
加密密钥: [ e43a642a 8e424cdb eb7c6331 ]
```

解密文件会出现在目录下



ARCHPR 4.54 Professional Edition

加密的 ZIP/RAR/ACE/ARJ 文件: C:\Users\Happy\Desktop\题目\warmup_3

攻击类型: 明文

明文文件路径: C:\Users\Happy\Desktop\题目\open_forum.zip

开始于: 0

密钥 e43a642a 密钥 8e424cdb 密钥 eb7c6331

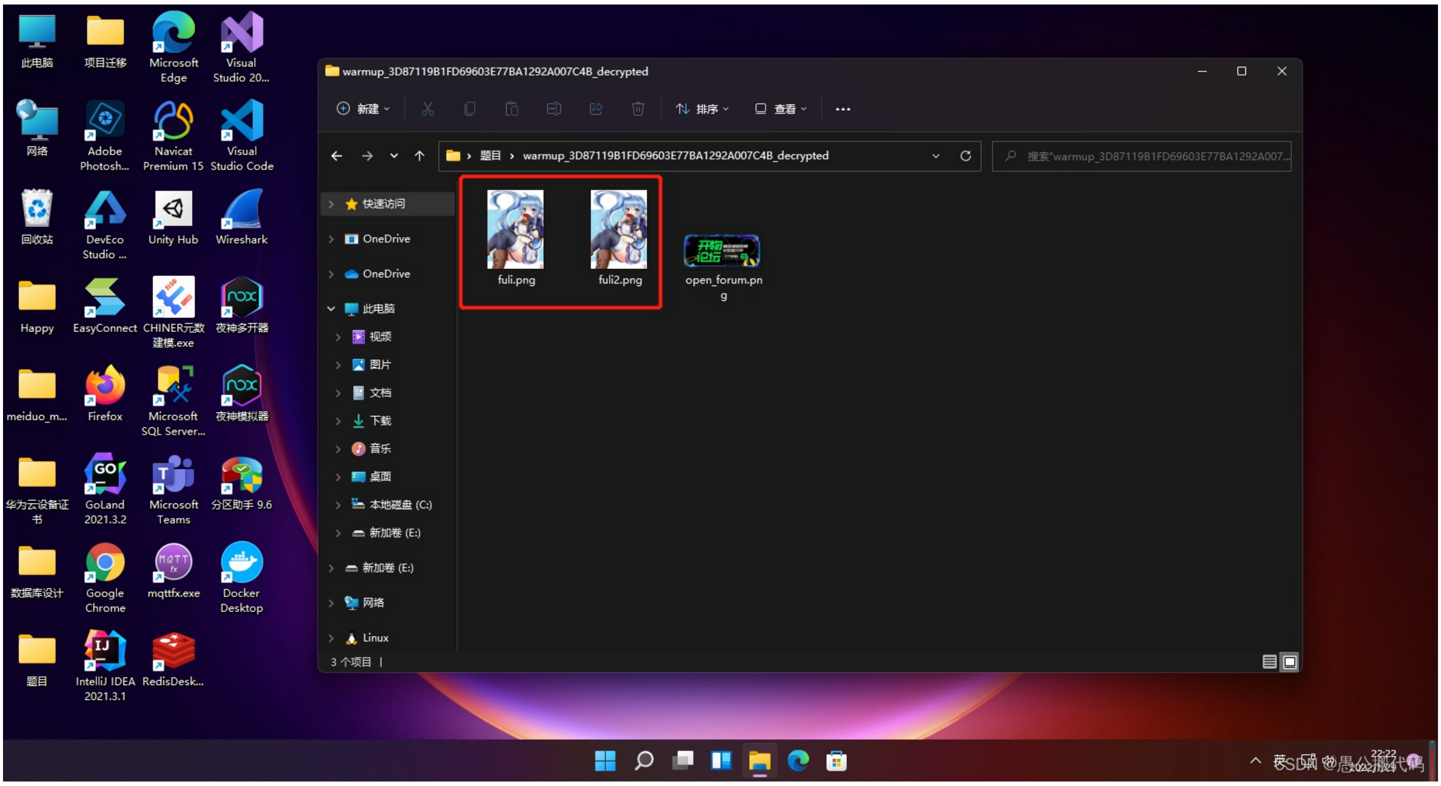
☑ 允许使用二进制文件作为明文 ZIP 档案文件

状态窗口

```
2022/1/29 22:13:23 - 自动保存路径选择被取消，自动保存现在已禁用
2022/1/29 22:13:23 - 明文攻击已开始
2022/1/29 22:20:49 - 加密密钥已成功恢复!
2022/1/29 22:21:22 - 档案文件成功解密。
```

ARCHPR version 4.54 (c) 1997-2012 ElcomSoft Co. Ltd.

题目
1.txt
open_forum.png
open_forum.zip
warmup_3D87119B1FD69603E77BA1292A007C4B.zip
warmup_3D87119B1FD69603E77BA1292A007C4B_decrypte...
5 个项目
2022年1月29日



warmup_3D87119B1FD69603E77BA1292A007C4B_decrypted
fuli.png
fuli2.png
open_forum.png
3 个项目

3.盲水印

盲水印脚本bwm.py

```
#!/usr/bin/env python
# -*- coding: utf8 -*-

import sys
import random
import random

cmd = None
debug = False
seed = 20160930
oldseed = False
alpha = 3.0

if __name__ == '__main__':
    if '-h' in sys.argv or '--help' in sys.argv or len(sys.argv) < 2:
        print ('Usage: python bwm.py <cmd> [arg...] [opts...]')
        print ('  cmds:')
        print ('    encode <image> <watermark> <image(encoded)>')
        print ('           image + watermark -> image(encoded)')
        print ('    decode <image> <image(encoded)> <watermark>')
        print ('           image + image(encoded) -> watermark')
        print ('  opts:')
        print ('    --debug,          Show debug')
        print ('    --seed <int>,     Manual setting random seed (default is 20160930)')
        print ('    --oldseed         Use python2 random algorithm.')
        print ('    --alpha <float>,  Manual setting alpha (default is 3.0)')
        sys.exit(1)
    cmd = sys.argv[1]
    if cmd != 'encode' and cmd != 'decode':
        print ('Wrong cmd %s' % cmd)
        sys.exit(1)
    if '--debug' in sys.argv:
        debug = True
        del sys.argv[sys.argv.index('--debug')]
    if '--seed' in sys.argv:
        p = sys.argv.index('--seed')
        if len(sys.argv) <= p+1:
            print ('Missing <int> for --seed')
            sys.exit(1)
        seed = int(sys.argv[p+1])
        del sys.argv[p+1]
        del sys.argv[p]
    if '--oldseed' in sys.argv:
        oldseed = True
        del sys.argv[sys.argv.index('--oldseed')]
    if '--alpha' in sys.argv:
        p = sys.argv.index('--alpha')
        if len(sys.argv) <= p+1:
            print ('Missing <float> for --alpha')
            sys.exit(1)
        alpha = float(sys.argv[p+1])
        del sys.argv[p+1]
        del sys.argv[p]
    if len(sys.argv) < 5:
        print ('Missing arg...')
        sys.exit(1)
    fn1 = sys.argv[2]
    fn2 = sys.argv[3]
    fn3 = sys.argv[4]

import cv2
import numpy as np
import matplotlib.pyplot as plt

# OpenCV是以(BGR)的顺序存储图像数据的
```

```python
# 而Matplotlib是以(RGB)的顺序显示图像的
def bgr_to_rgb(img):
    b, g, r = cv2.split(img)
    return cv2.merge([r, g, b])

if cmd == 'encode':
    print ('image<%s> + watermark<%s> -> image(encoded)<%s>' % (fn1, fn2, fn3))
    img = cv2.imread(fn1)
    wm = cv2.imread(fn2)

    if debug:
        plt.subplot(231), plt.imshow(bgr_to_rgb(img)), plt.title('image')
        plt.xticks([]), plt.yticks([])
        plt.subplot(234), plt.imshow(bgr_to_rgb(wm)), plt.title('watermark')
        plt.xticks([]), plt.yticks([])

    # print img.shape # 高, 宽, 通道
    h, w = img.shape[0], img.shape[1]
    hwm = np.zeros((int(h * 0.5), w, img.shape[2]))
    assert hwm.shape[0] > wm.shape[0]
    assert hwm.shape[1] > wm.shape[1]
    hwm2 = np.copy(hwm)
    for i in range(wm.shape[0]):
        for j in range(wm.shape[1]):
            hwm2[i][j] = wm[i][j]

    if oldseed: random.seed(seed,version=1)
    else: random.seed(seed)
    m, n = list(range(hwm.shape[0])), list(range(hwm.shape[1]))
    if oldseed:
        random.shuffle(m,random=random.random)
        random.shuffle(n,random=random.random)
    else:
        random.shuffle(m)
        random.shuffle(n)

    for i in range(hwm.shape[0]):
        for j in range(hwm.shape[1]):
            hwm[i][j] = hwm2[m[i]][n[j]]

    rwm = np.zeros(img.shape)
    for i in range(hwm.shape[0]):
        for j in range(hwm.shape[1]):
            rwm[i][j] = hwm[i][j]
            rwm[rwm.shape[0] - i - 1][rwm.shape[1] - j - 1] = hwm[i][j]

    if debug:
        plt.subplot(235), plt.imshow(bgr_to_rgb(rwm)), \
            plt.title('encrypted(watermark)')
        plt.xticks([]), plt.yticks([])

    f1 = np.fft.fft2(img)
    f2 = f1 + alpha * rwm
    _img = np.fft.ifft2(f2)

    if debug:
        plt.subplot(232), plt.imshow(bgr_to_rgb(np.real(f1))), \
            plt.title('fft(image)')
        plt.xticks([]), plt.yticks([])
    img_wm = np.real(_img)

    assert cv2.imwrite(fn3, img_wm, [int(cv2.IMWRITE_JPEG_QUALITY), 100])

    # 这里计算下保存前后的(溢出)误差
    img_wm2 = cv2.imread(fn3)
    sum = 0
    for i in range(img_wm.shape[0]):
        for j in range(img_wm.shape[1]):
            for k in range(img_wm.shape[2]):
                sum += np.power(img_wm[i][j][k] - img_wm2[i][j][k], 2)
    miss = np.sqrt(sum) / (img_wm.shape[0] * img_wm.shape[1] * img_wm.shape[2]) * 100
    print ('Miss %s%% in save' % miss)

    if debug:
        plt.subplot(233), plt.imshow(bgr_to_rgb(np.uint8(img_wm))), \
            plt.title('image(encoded)')
        plt.xticks([]), plt.yticks([])

    f2 = np.fft.fft2(img_wm)
    rwm = (f2 - f1) / alpha
    rwm = np.real(rwm)

    wm = np.zeros(rwm.shape)
    for i in range(int(rwm.shape[0] * 0.5)):
        for j in range(rwm.shape[1]):
            wm[m[i]][n[j]] = np.uint8(rwm[i][j])
    for i in range(int(rwm.shape[0] * 0.5)):
        for j in range(rwm.shape[1]):
            wm[rwm.shape[0] - i - 1][rwm.shape[1] - j - 1] = wm[i][j]

    if debug:
        assert cv2.imwrite('_bwm.debug.wm.jpg', wm)
        plt.subplot(236), plt.imshow(bgr_to_rgb(wm)), plt.title(u'watermark')
        plt.xticks([]), plt.yticks([])

    if debug:
        plt.show()

elif cmd == 'decode':
    print ('image<%s> + image(encoded)<%s> -> watermark<%s>' % (fn1, fn2, fn3))
    img = cv2.imread(fn1)
    img_wm = cv2.imread(fn2)

    if debug:
        plt.subplot(231), plt.imshow(bgr_to_rgb(img)), plt.title('image')
        plt.xticks([]), plt.yticks([])
        plt.subplot(234), plt.imshow(bgr_to_rgb(img_wm)), plt.title('image(encoded)')
        plt.xticks([]), plt.yticks([])

    if oldseed: random.seed(seed,version=1)
    else: random.seed(seed)
    m, n = list(range(int(img.shape[0] * 0.5))), list(range(img.shape[1]))
    if oldseed:
        random.shuffle(m,random=random.random)
        random.shuffle(n,random=random.random)
    else:
        random.shuffle(m)
        random.shuffle(n)
    f1 = np.fft.fft2(img)
    f2 = np.fft.fft2(img_wm)

    if debug:
        plt.subplot(232), plt.imshow(bgr_to_rgb(np.real(f1))), \
            plt.title('fft(image)')
        plt.xticks([]), plt.yticks([])
        plt.subplot(235), plt.imshow(bgr_to_rgb(np.real(f1))), \
            plt.title('fft(image(encoded))')
        plt.xticks([]), plt.yticks([])

    rwm = (f2 - f1) / alpha
    rwm = np.real(rwm)

    if debug:
        plt.subplot(233), plt.imshow(bgr_to_rgb(rwm)), \
            plt.title('encrypted(watermark)')
        plt.xticks([]), plt.yticks([])

    wm = np.zeros(rwm.shape)
    for i in range(int(rwm.shape[0] * 0.5)):
        for j in range(rwm.shape[1]):
            wm[m[i]][n[j]] = np.uint8(rwm[i][j])
    for i in range(int(rwm.shape[0] * 0.5)):
        for j in range(rwm.shape[1]):
            wm[rwm.shape[0] - i - 1][rwm.shape[1] - j - 1] = wm[i][j]
    assert cv2.imwrite(fn3, wm)

    if debug:
        plt.subplot(236), plt.imshow(bgr_to_rgb(wm)), plt.title(u'watermark')
        plt.xticks([]), plt.yticks([])

    if debug:
        plt.show()
```

保存requirements.txt文件

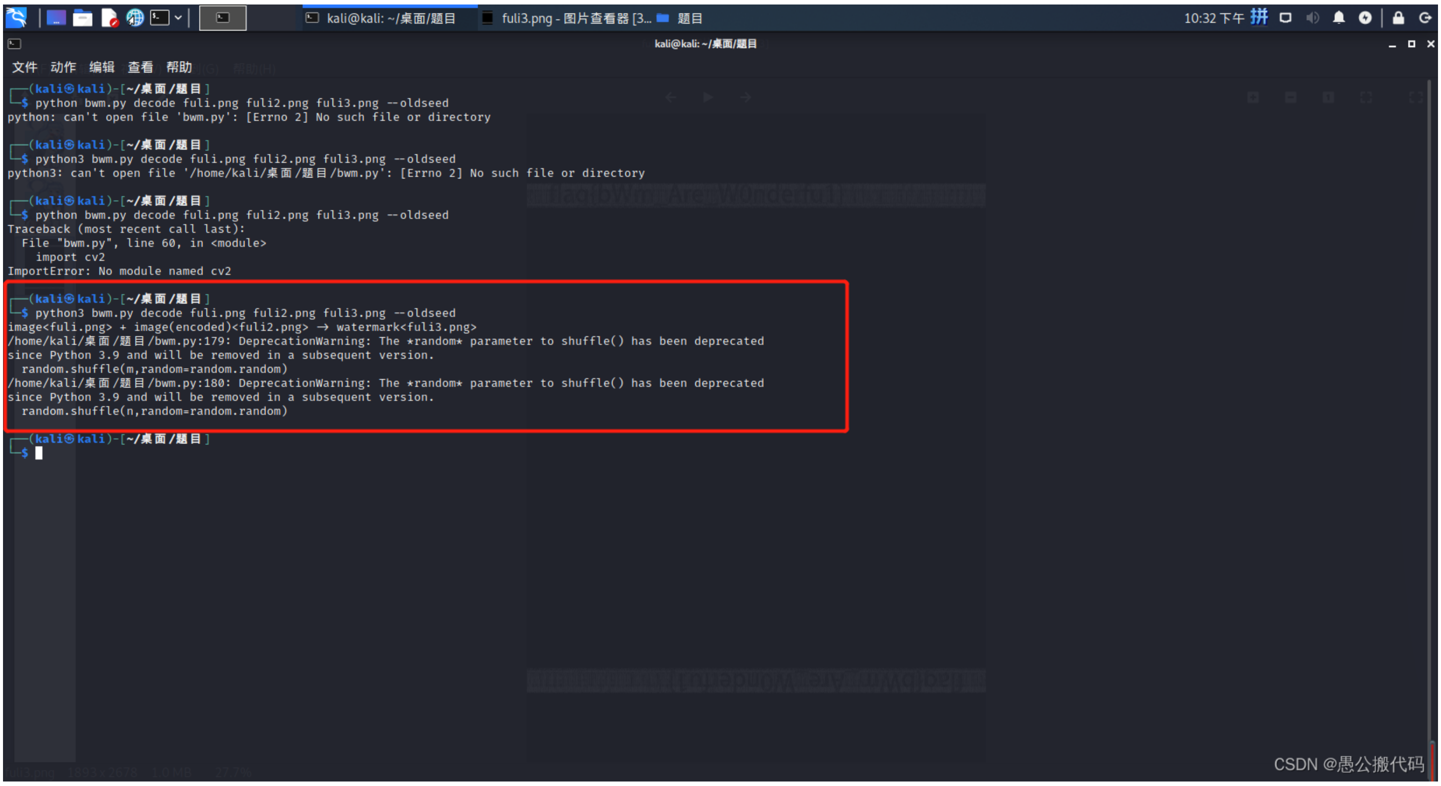
```
opencv-python==4.2.0.34
matplotlib==2.1.1
```

执行命令安装对应包

```
pip install -r requirements.txt
```

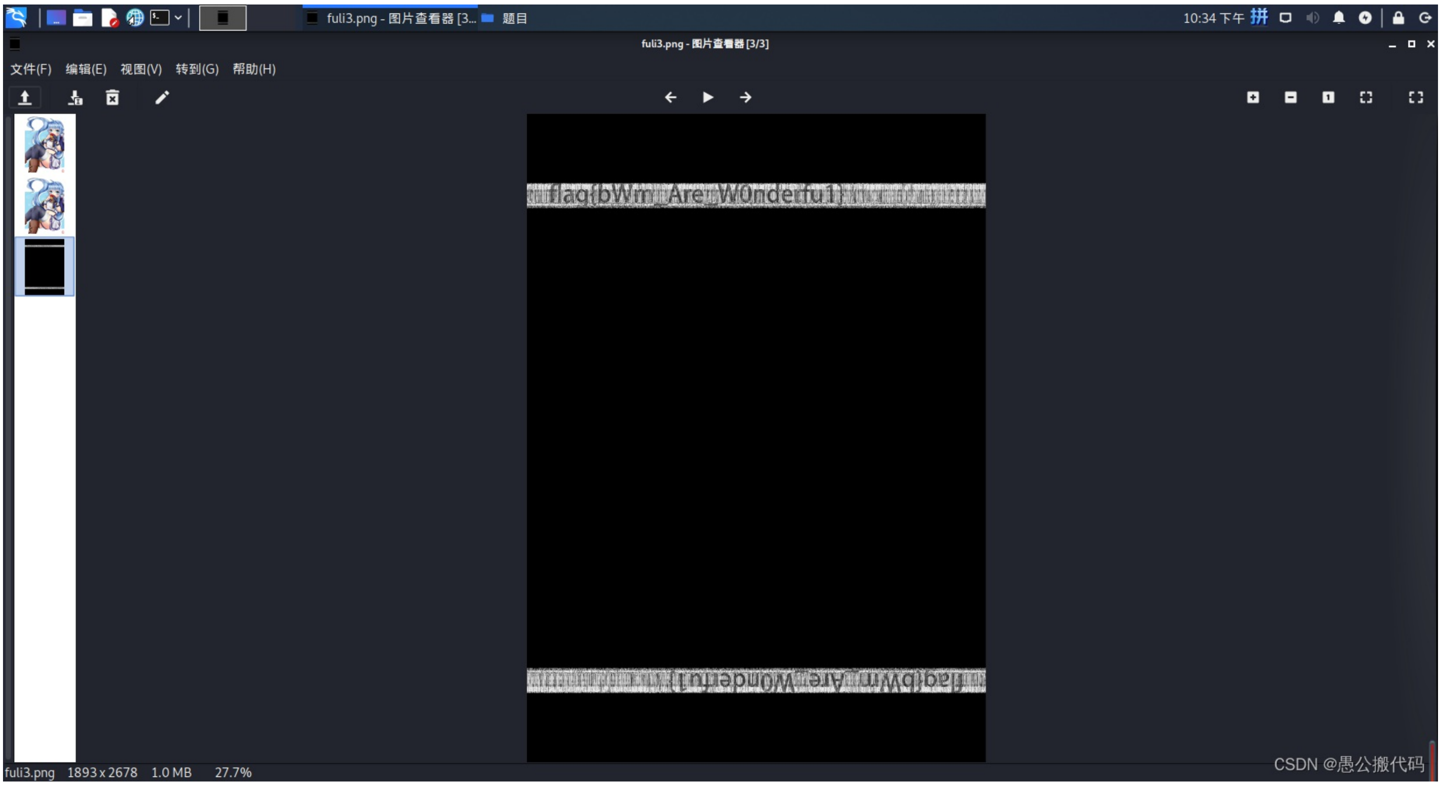
提取图中的盲水印

```
python3 bwm.py decode day1.png day2.png day3.png --oldseed
```



```
┌──(kali㉿kali)-[~/桌面/题目]
└─$ python bwm.py decode fuli.png fuli2.png fuli3.png --oldseed
python: can't open file 'bwm.py': [Errno 2] No such file or directory

┌──(kali㉿kali)-[~/桌面/题目]
└─$ python3 bwm.py decode fuli.png fuli2.png fuli3.png --oldseed
python3: can't open file '/home/kali/桌面/题目/bwm.py': [Errno 2] No such file or directory

┌──(kali㉿kali)-[~/桌面/题目]
└─$ python bwm.py decode fuli.png fuli2.png fuli3.png --oldseed
Traceback (most recent call last):
  File "bwm.py", line 60, in <module>
    import cv2
ImportError: No module named cv2

┌──(kali㉿kali)-[~/桌面/题目]
└─$ python3 bwm.py decode fuli.png fuli2.png fuli3.png --oldseed
image<fuli.png> + image(encoded)<fuli2.png> → watermark<fuli3.png>
/home/kali/桌面/题目/bwm.py:179: DeprecationWarning: The *random* parameter to shuffle() has been deprecated
since Python 3.9 and will be removed in a subsequent version.
  random.shuffle(m,random=random.random)
/home/kali/桌面/题目/bwm.py:180: DeprecationWarning: The *random* parameter to shuffle() has been deprecated
since Python 3.9 and will be removed in a subsequent version.
  random.shuffle(n,random=random.random)

┌──(kali㉿kali)-[~/桌面/题目]
└─$ 
```

4.得到图片



fuli3.png - 图片查看器 [3...
题目
10:34 下午
fuli3.png - 图片查看器 [3/3]
文件(F) 编辑(E) 视图(V) 转到(G) 帮助(H)
flag{bWm_Are_W0nderfu1}
fuli3.png 1893x2678 1.0 MB 27.7%
CSDN @愚公搬代码

falg为：`flag{bWm_Are_W0nderfu1}`

总结

- 盲水印
- 明文攻击
- crc32碰撞