



BUUCTF MISC

原创

[zh0u9527](#)  于 2021-09-27 22:19:37 发布  101  收藏

文章标签: [python](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/Hello_super/article/details/120518708

版权

BUUCTF MISC

[\[GXYCTF2019\]gakki](#)

1. 下载附件拿到一张图片，使用binwalk进行分离，得到一个压缩包。

```
root@kali:~# binwalk wolaopo.jpg
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	JPEG image data, JFIF standard 1.01
30	0x1E	TIFF image data, big-endian, offset of first image directory: 8
137448	0x218E8	RAR archive data, version 4.x, first volume type: MAIN_HEAD

CSDN @Hello_super

2. 使用ARCPHR压缩包爆破工具进行爆破，成功拿到密码。



3. 在压缩包中拿到flag.txt文件，发现大量乱序字符，使用Python脚本工具进行统计。

```
flag.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
#2V0VI_05X$GygD3*g@gYurMGim#1=)D_@Z(JcRevlyGq&N-dgPH8XXSGL{@9}zVmlmxv1vEwbqr)ea!
YMI2IznoV_bMrXLbwFrgaiQYfsVN14weObXp)(ybmXjXuTkFuj1pG54!mij1)
{41gKmFL&Zgeho01PPEwE=r*csndRof$X7JBJ=CaNRGMjLY_-GiqlDHWaVk-XZ*8ID5!kLb(OH%8u2LtQXX3QV
{1Lh)LyGF#kpV$}GXRKla)u(pw(&ggmYU82HLWhJgngOjhwofkqC(Hi)g!
GXrY6=UQGvaeOlrvG*jkGjgGRTY78OI$w0&tzZ1t)z#_c^t8GrskRcz9YKE_)4B(U
$r3qUcCwz4BVq92&0UBaWg#e23&oZ)G(zll=(k=^YTIZrQkryM6oW!#-0*{X1oiX4Zwi#jhOUm*aM{NFX-
s=j2M*$B_EMkF(R=QufYYVIOHmNGaDST0e)}w4q8{l(NY)BGCWKiGiM0(o$jPW@b!LeQbRM!k$8H
$5z7JhE4alHM-LsAn_PSSg_=lkHmGGok$A$Wrkd^yD9KT#zF-ByEJx-!lg3cZPAv{SkP7zult3NOZ)Kf-
Xah)%x3X4kx{SdoYB#icdYmB_T3rggCts^EcZl_R^w-B-B5H=4fGRx-lkH59BoB!
CSDN @Hello_super
```

```

# -*- coding:utf-8 -*-
#Author: mochu7
alphabet = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#$%^&*()_+- =\\{\}\[\]"
strings = open('./flag.txt').read()

result = {}
for i in alphabet:
    counts = strings.count(i)
    i = '{0}'.format(i)
    result[i] = counts

res = sorted(result.items(),key=lambda item:item[1],reverse=True)
for data in res:
    print(data)

for i in res:
    flag = str(i[0])
    print(flag[0],end="")

```

[ACTF新生赛2020]base64隐写

1. 下载附件，得到一个文本文件，里面全是base64加密的数据。

*ComeOn!.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```

STJsdVkyeDFaR1U4YVc5emRISmxZVzArQ2c9PQ1 =
STJsdVkyeDFaR1U4YzNSeWFXNW5QZ289DQ = =
STJsdVkyeDFaR1U4WTNOMGNtbHVaejRLDV = =
STJsdVkyeDFaR1U4WTNOMFpHbHZQZ289DQ = =
STJSbFptbHVau0J0WVhodUIEazVPUW89Dd = =
Q2c9PQ1 =
ZFhOcGJtY2dibUZ1WlhOd1IXTmxJSE4wWkRzSw1 =
Q2c9PQ0 =
WTJoaGNpUmpZVnR0VWhodVhTd2dZMkpiYldGNGJsMDdDZz09DU = =
Q2c9PQ1 =
YVc1MEIHTnRjQ2h6ZEhKcGJtY2dZU3dnYzNSeWFXNW5JR2lwQ2c9PQ2 =
ZXdvPQ1 =
GUNRZ3UuIkQ1L2V4YVc1NSeWFXNW5QZ289DQ = =

```

CSDN @Hello_super

2. 编写Python脚本进行解密。

flag.txt *	17	29	文本文档	20'
secret.txt *	19	33	文本文档	20:

3. 使用winhex打开图片，在文件尾部发现flag.txt文件字样，于是开心的使用binwalk命令进行提取，但是发现咋么也提取不了，傻了！为啥？

```

root@kali:~# binwalk woo.jpg

DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0         JPEG image data, JFIF standard 1.01
16733       0x415D      End of Zip archive, footer length: 22

root@kali:~# binwalk -e woo.jpg

DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0         JPEG image data, JFIF standard 1.01
16733       0x415D      End of Zip archive, footer length: 22

```

4. 通过仔细查看文件的hex值，发现那个文件头并不完整。

```

560  32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 2222222222222222
576  32 32 32 32 32 32 32 32 32 32 32 32 32 03 04 2222222222222222
592  14 00 00 00 08 00 CB A2 82 4F D8 30 C5 B0 11 00 00 00 00 00 00 00
608  00 00 11 00 00 00 08 00 00 00 66 6C 61 67 2E 74 78 74 0A 00 20 00 00 00
624  78 74 2B C9 C8 2C 56 00 A2 92 8C 54 85 B4 9C C4 xt+ÉÈ, V ç'GT... 'œÄ
640  74 3D 00 50 4B 01 02 14 00 14 00 00 00 08 00 CB t= PK
656  A2 82 4F D8 30 C5 B0 11 00 00 00 11 00 00 00 08 ç,øø0Å°
672  00 24 00 00 00 00 00 00 00 20 00 00 00 00 00 00 $
688  00 66 6C 61 67 2E 74 78 74 0A 00 20 00 00 00 00 flag.txt
704  00 01 00 18 00 01 02 2B 25 0B A9 D5 01 1D 7B 6F +% @ö {o
720  54 0B A9 D5 01 79 58 D8 1C 0B A9 D5 01 50 4B 05 T @ö yXø @ö PK
736  06 00 00 00 00 01 00 01 00 5A 00 00 00 37 00 00 z 7
752  00 00 00

```

50 4B 03 04 为zip文件的头，我们在这里将50 4B补上。

```

.6560  32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 2222222222222222
.6576  32 32 32 32 32 32 32 32 32 32 32 32 32 32 00 00 2222222222222222
.6592  03 04 14 00 00 00 08 00 CB A2 82 4F D8 30 C5 B0 00 00 00 00 00 00 00 00
.6608  11 00 00 00 11 00 00 00 08 00 00 00 66 6C 61 67 78 74 0A 00 20 00 00 00
.6624  2E 74 78 74 2B C9 C8 2C 56 00 A2 92 8C 54 85 B4 .txt+ÉÈ, V ç'GT... '
.6640  9C C4 74 3D 00 50 4B 01 02 14 00 14 00 00 00 08 œÄt= PK
.6656  00 CB A2 82 4F D8 30 C5 B0 11 00 00 00 11 00 00 08 Èç,øø0Å°
.6672  00 08 00 24 00 00 00 00 00 00 00 20 00 00 00 00 $
.6688  00 00 00 66 6C 61 67 2E 74 78 74 0A 00 20 00 00 00 flag.txt
.6704  00 00 00 01 00 18 00 01 02 2B 25 0B A9 D5 01 1D +% @ö
.6720  7B 6F 54 0B A9 D5 01 79 58 D8 1C 0B A9 D5 01 50 {oT @ö yXø @ö P
.6736  4B 05 06 00 00 00 00 01 00 01 00 5A 00 00 00 37 K z 7
.6752  00 00 00 00 00

```

在03上右击->粘贴0字节->输入2回车即可！

```

0016560  32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 2222222222222222
0016576  32 32 32 32 32 32 32 32 32 32 32 32 32 32 50 4B 2222222222222222
0016592  03 04 14 00 00 00 08 00 CB A2 82 4F D8 30 C5 B0 Èç,øø0Å°
0016608  11 00 00 00 11 00 00 00 08 00 00 00 66 6C 61 67 flag
0016624  2E 74 78 74 2B C9 C8 2C 56 00 A2 92 8C 54 85 B4 .txt+ÉÈ, V ç'GT... '

```

0016640	9C C4 74 3D 00 50 4B 01 02 14 00 14 00 00 00 08	œÄt= PK
0016656	00 CB A2 82 4F D8 30 C5 B0 11 00 00 00 11 00 00	Ëc,OØ0Å°
0016672	00 08 00 24 00 00 00 00 00 00 00 20 00 00 00 00	\$
0016688	00 00 00 66 6C 61 67 2E 74 78 74 0A 00 20 00 00	flag.txt
0016704	00 00 00 01 00 18 00 01 02 2B 25 0B A9 D5 01 1D	+% @Û
0016720	7B 6F 54 0B A9 D5 01 79 58 D8 1C 0B A9 D5 01 50	{oT @Û yXØ @Û P
0016736	4B 05 06 00 00 00 00 01 00 01 00 5A 00 00 00 37	K z 7
0016752	00 00 00 00 00	

右击我们添加的00,编辑->...->写入即可

CSDN @Hello_super

5. 然后将图片文件的后缀名改为zip，解压，成功拿到flag.txt文件，但是那个并不是真正的flag。再结合题目说的明文攻击，我们将拿到flag.txt文件压缩为zip格式，使用ARCHPR爆破攻击进行攻击。

简单加密

1. 下载附件得到一段密文和加密脚本。

```
==jEgWTn8kJrRyRFBuKJLuzH1LmDTAzs
```

```
function encode( $str = '' ){
    $strrev = strrev( $str );
    $string = '';
    for( $i=0; $i < strlen($strrev);$i++){
        $char = substr( $strrev, $i, 1 );
        $ordChar = ord( $char ) + 1;
        $char = chr( $ordChar );
        $string = $string.$char;
    }

    $string = base64_encode( $string );
    $string = strrev( $string );
    $string = str_rot13( $string );
    return $string;
}
```

2. 其实这个加密脚本挺简单的，但是由于本人基础薄弱，还是花了好几个小时才写出来，但是想起来还是值得。

```
function decode( $str = '' ){
    $strrev = strrev($str);
    $strrev = str_rot13($strrev);
    $strrev = base64_decode($strrev);
    $string = '';
    for ( $i=0; $i < strlen($strrev);$i++){
        $char = substr( $strrev, $i, 1 );
        $ordChar = ord( $char ) - 1;
        $char = chr( $ordChar );
        $string = $string.$char;
    }
    return $string;
}
```

```
// echo encode("123");
echo decode( str: "==jEgWTn8k JrRyRFBuKJLuzH1LmDTAzs"); # ==jEgWTn8k JrRyRFBuKJLuz

decode()

test.php (1) x
H:\phpstudy_pro\Extensions\php\php7.3.4nts\php.exe H:\phpstudy_pro\WWW\question\Code
}bEB54QgWXwMGHCxk{ga1F
Process finished with| exit code 0 CSDN @Hello_super
```

使用python的中切面倒置字符串即可：

```
>>>string = '}bEB54QgWXwMGHCxk{ga1F'
>>> print(string[::-1]);
Flag{kxCHGMwXWgQ45BEb}
```

黑客帝国

1. 下载附件，得到一个文件，使用winhex查看发现是zip文件，解压得到一个文本文件，文件的内容全是十六进制的，且文件的开头为52617221，初步判断这是一个rar文件，使用python将十六进制以二进制形式写入另一个文件。

```
import binascii

content = ''
with open('resource.txt') as file_obj:
    content = file_obj.read()

out=open('res.txt','wb')
out.write(binascii.unhexlify(content))
out.close()
```


2. 将拿到的文件再次解压，但是需要密码，使用爆破工具进行掩码爆破，成功拿到密码 3690。
3. 打开文件的时候，发现是一张无法正常显示的png图片，开始使用winhex查看的时候也没有啥发现，看了大佬的wp才知道原来这个是一个jpg文件，jpg图片的文件头被换成了png的文件头导致无法正常显示。

```
729ec4d72da9599a308c64fe...
Offset 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
00000000 89 50 4E 47 00 10 4A 46 49 46 00 01 01 01 00 48 PNG JFIF H
00000016 00 48 00 00 FF DB 00 43 00 02 01 01 02 01 01 02 H yU c
00000032 02 02 02 02 02 02 02 03 05 03 03 03 03 03 06 04
00000048 04 03 05 07 06 07 07 07 06 07 07 08 09 0B 09 08
00000064 08 0A 08 07 07 0A 0D 0A 0A 0B 0C 0C 0C 0C 07 09 png文件头 yU c
00000080 0E 0F 0D 0C 0E 0B 0C 0C 0C FF DB 00 43 01 02 02
00000096 02 03 03 03 06 03 03 06 0C 08 07 08 0C 0C 0C 0C
00000112 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C
00000128 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C
00000144 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C FF C0 yÀ
00000160 00 11 08 03 06 04 00 03 01 22 00 02 11 01 03 11 "
00000176 01 FF C4 00 1F 00 00 01 05 01 01 01 01 01 01 00 yÄ
00000192 00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 09 CSDN @Hello_super
00000208 0A 0B FF C4 00 B5 10 00 02 01 03 03 02 04 03 05 yÄ u
```

```
Offset 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
00000000 FF D8 4A 46 49 46 00 01 01 01 00 48 00 48 00 00 JFIF H H
00000016 FF DB 00 43 00 02 01 01 02 01 01 02 02 02 02 02 yU c
00000032 02 02 02 03 05 03 03 03 03 03 06 04 04 03 05 07
00000048 06 07 07 07 06 07 07 08 09 0B 09 08 08 0A 08 07 jpg文件头
00000064 07 0A 0D 0A 0A 0B 0C 0C 0C 0C 07 09 0E 0F 0D 0C
00000080 0E 0B 0C 0C 0C FF DB 00 43 01 02 02 02 03 03 03
00000096 06 03 03 06 0C 08 07 08 0C 0C 0C 0C 0C 0C 0C 0C yU c
```

改好之后，图片就能正常显示了，成功拿到了flag。



[GUET-CTF2019]KO

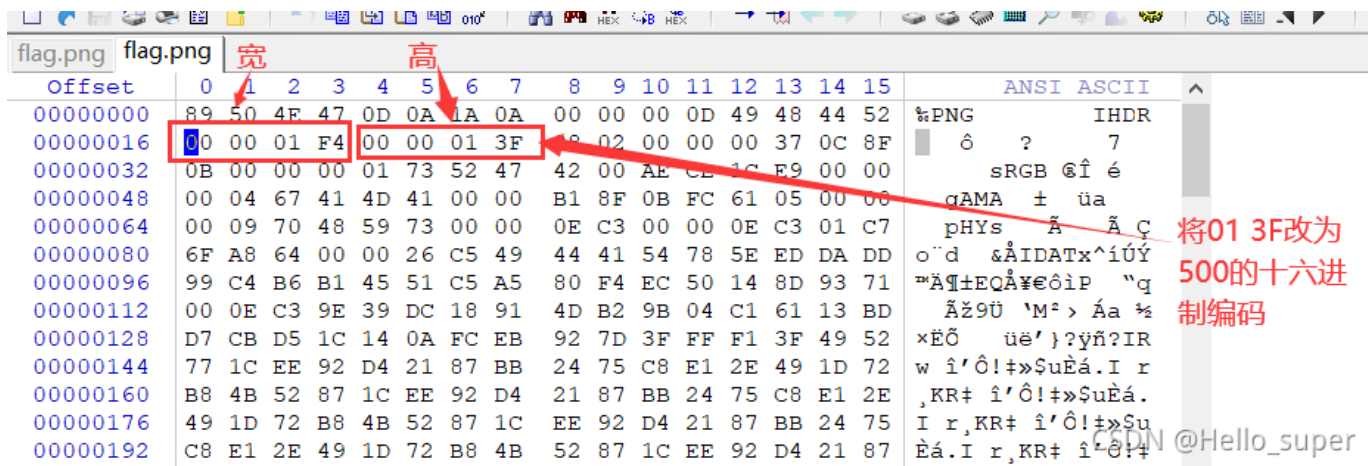
Ook!在线解密

[MRCTF2020]ezmisc

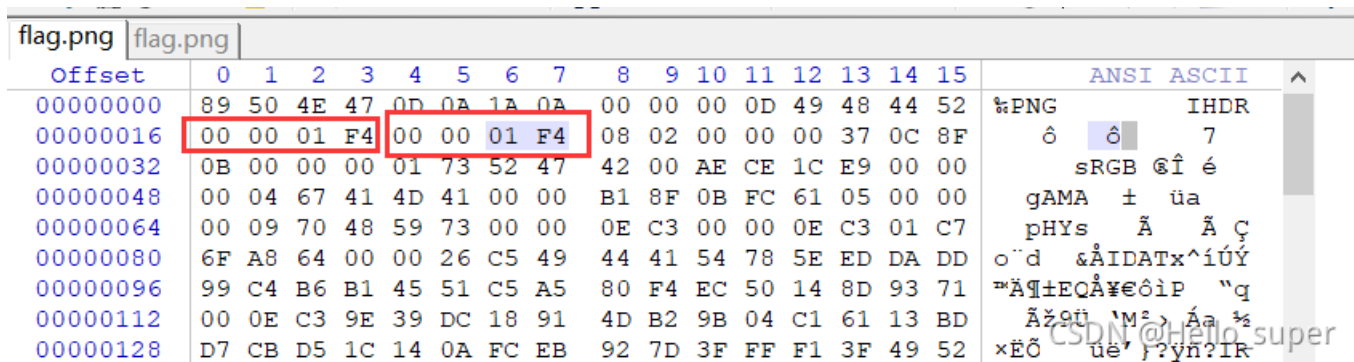
1. 下载附件得到一张图片，刚开始打开图片的时候就发现图片的像素有点不对500x319。想着图片的高度是不是被篡改过。



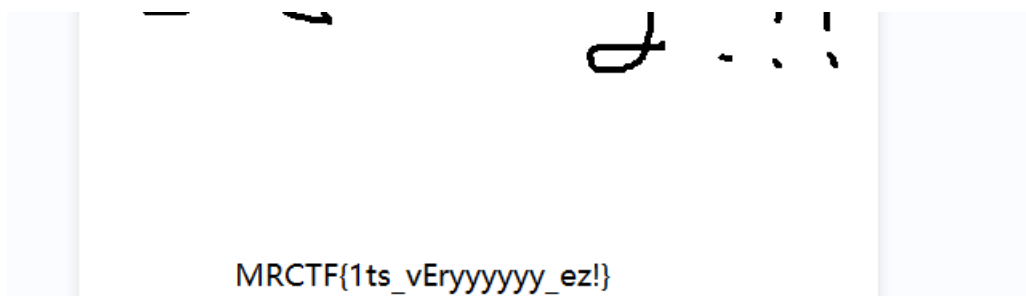
2. 使用winhex打开，修改图片的高，注意点，既然图片的宽度是500，那我们就把高度也改为500，500对应的十六进制编码为1f4。



将01 3F改为500的十六进制编码



保存，在图片的底部成功显示出flag。



下载附件得到一个文本文件。

题目: caesar

描述: gmbhjtdbftbs

flag格式: XXX 明文

提交: 直接提交明文 (小写)

CSDN @Hello_super

当时一点头绪都没有,看了大佬的博客才知道,直接使用脚本进行解密。

```
def change(c, i):
    num = ord(c)
    if (num >= 33 and num <= 126):
        num = 33 + (num + i - 33) % (94) # 126-33=93
    return chr(num)

def kaisa_jiAme(string, i):
    string_new = ''
    for s in string:
        string_new += change(s, i)
    print(string_new)
    return string_new

# 本题有种暴力解密感觉
def kaisa_jiEmi(string):
    for i in range(0, 94):
        print('第' + str(i + 1) + '种可能:', end=' ')
        # 区别在于 string 是该对象原本就是字符串类型, 而 str() 则是将该对象转换成字符串类型。
        kaisa_jiAme(string, i)

# 你要知道input输入的数据类型都是string
def main():
    print('请输入操作, 注意不是平常26种:')
    choice = input('1: 恺撒加密, 2: 凯撒穷举解密. 请输入1或2: ')
    if choice == '1':
        string = input('请输入需要加密字符串: ')
        num = int(input('请输入需要加密的KEY: '))
        kaisa_jiAme(string, num)
    elif choice == '2':
        string = input('请输入需要解密字符串: ')
        kaisa_jiEmi(string)
    else:
        print('输入错误, 请重试')
        main()

if __name__ == '__main__':
    main()
```

撒凯密码解密 ×

第93种可能: ek tnr b dr q

第94种可能: flagiscaesar

Process finished with exit code 0