

2017湖湘杯网络安全大赛Writeup

转载

普通网友 于 2017-12-15 09:36:14 发布 4226 收藏 4
分类专栏: [CTF](#)



[CTF 专栏收录该内容](#)

10 篇文章 0 订阅
订阅专栏

大赛分为初赛、复赛、决赛。初赛150道理论题，前300名进入复赛，复赛为夺旗赛（CTF）竞赛模式。

湖湘杯网络安全大赛相关链接: [【传送门】](#)

复赛为web渗透，反编译，破解，加密&解密 一共15道题。（以下是我们队伍答题的writeup）

Web200

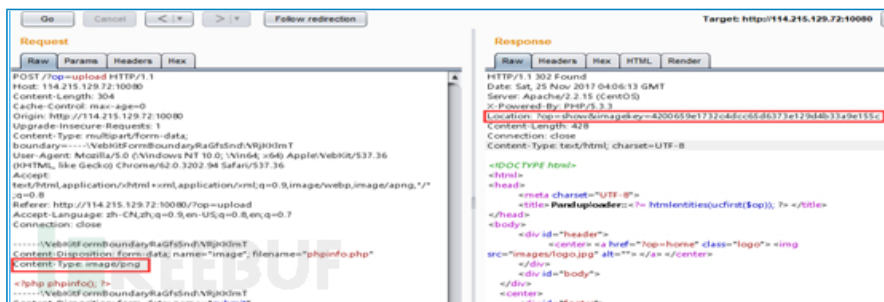
题目: 简简单单的上传，没有套路。

答题地址:

<http://118.190.87.135:10080/>

<http://114.215.129.72:10080/>

<http://118.190.86.101:10080/>



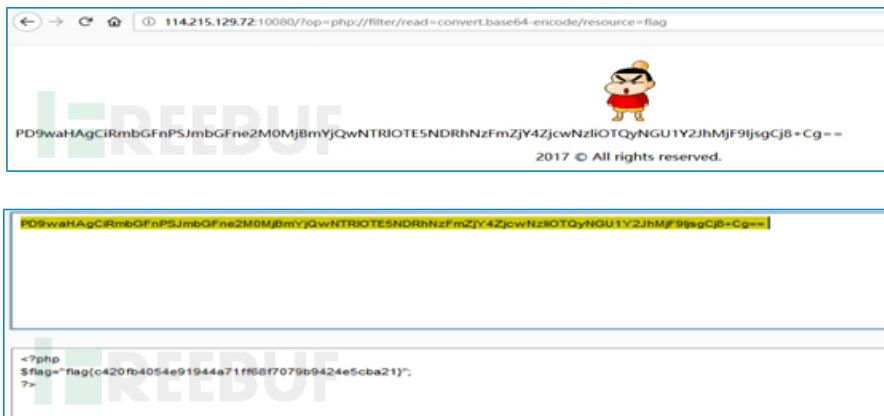
首先是在网站进行文件上传。修改Content-Type可绕过上传，这个题真的很想吐槽



最后经过各种尝试，get一枚文件包含。想到php://filter

<http://114.215.129.72:10080/?op=php://filter/read=convert.base64-encode/resource=flag>

get flag



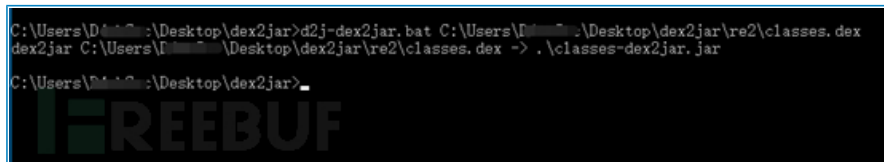
简单的android

题目：关于Android Crack的基本操作，对你来说SoEasy

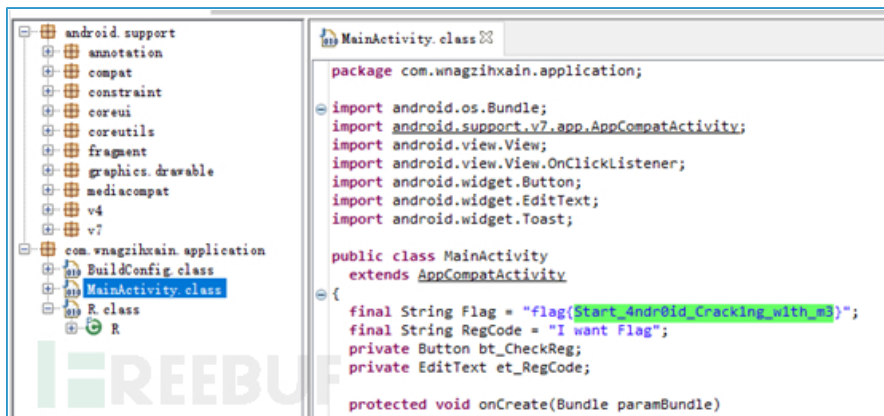
过程：

将apk格式改成zip解压出来

使用d2x2jar反汇编拿到源码



使用工具jd-gui查看源码



直接拿到flag。

Re4newer

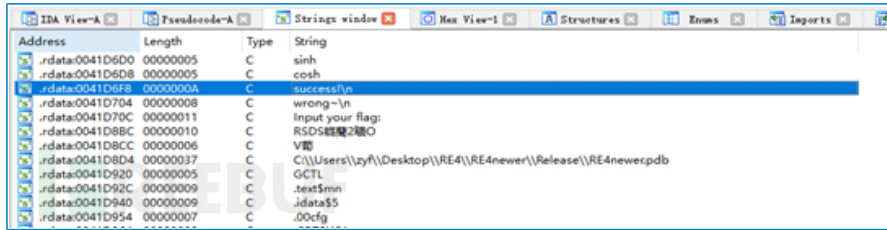
题目：逆向基本操作。

过程：

首先查壳，UPX，脱壳工具可以直接脱掉。



载入ida分析，shift+F12搜索字符串，发现字符串（success）



双击进去后，F5定位到关键代码



分析上述代码，取出41D740 等地址中的_DWORD与0x22异或即可得到flag

脚本及执行结果如下



流量分析

题目：



过程：

使用wireshark打开下载的流量包，直接导出HTTP对象，发现flag.zip文件

分组	主机名	内容类型	大小	文件名
6455				
6485				
6081	2967 qex.f.360.cn	multipart/form-data	1267 bytes	qexquery
6732	2969 qex.f.360.cn	application/octet-stream	197 bytes	qexquery
5129	3100 qex.f.360.cn	multipart/form-data	1291 bytes	qexquery
7155	3102 qex.f.360.cn	application/octet-stream	197 bytes	qexquery
6362	3202 192.168.199.203	application/zip	54 kB	flag.zip
12452	3262 qex.f.360.cn	multipart/form-data	1267 bytes	qexquery
12592	3264 qex.f.360.cn	application/octet-stream	197 bytes	qexquery
17270	3316 xjd.com		1440 bytes	exsites?spread_type=2&ad_ids=1555:!
6162	3323 xjd.com		1393 bytes	exsites?spread_type=2&ad_ids=1555:!
6254	3362 px.3.cn	application/json	392 bytes	mgets?skuids=_J_10201585618_J_10253
6456	3368 im-xjd.com	image/gif	0 bytes	LwkcuuJ3iBk9ZM0zrk9BIB0AD0bExed-
	3379 im-xjd.com	image/gif	0 bytes	OygAmpMkUT4LS0S0GoY4RGSf5R1BC

打开flag.zip 发现内容为rgb值的ce.txt文件（rgb值还原图像相关代码）

经过调试图片高宽和方向，代码如下：

```

from PIL import Image

f = open('/Users/SPY/Desktop/ce.txt','r')

length = 111
width = 887
pic = Image.new("RGB", (length, width))

for y in range(0,width):
    for x in range(0,length):
        l = f.next().split(',')
        pic.putpixel([x,y], (int(l[0]),int(l[1]),int(l[2])))

pic.transpose(Image.FLIP_LEFT_RIGHT).transpose(Image.ROTATE_90).show()

```

转换成图片后，get flag



Web300

题目：拿个shell就给flag。

答题地址：

<http://114.215.71.135:10080/>

<http://114.215.133.202:10080/>

http://118.190.77.141:10080

```

<?php
ini_set("display_errors", "On");
error_reporting(E_ALL | E_STRICT);
if(!isset($_GET['content'])){
    show_source(__FILE__);
    die();
}

function rand_string( $length ) {
    $chars = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
    $size = strlen( $chars );
    $str = '';
    for( $i = 0; $i < $length; $i++ ) {
        $str .= $chars[ rand( 0, $size - 1 ) ];
    }
    return $str;
}

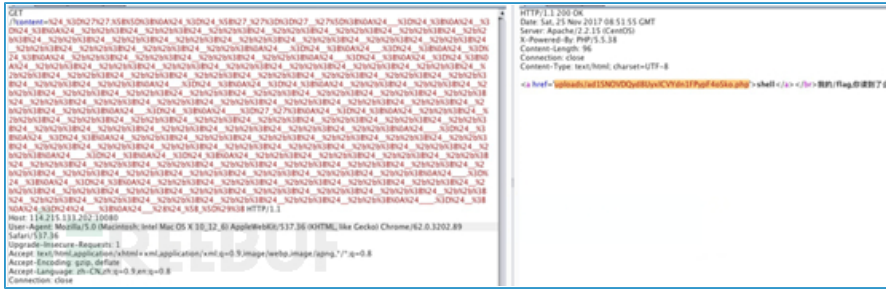
$data = $_GET['content'];
$black_char = array('a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z');
foreach ($black_char as $b) {
    if (strpos($data, $b) !== false) {
        die("关键字WAF");
    }
}

```

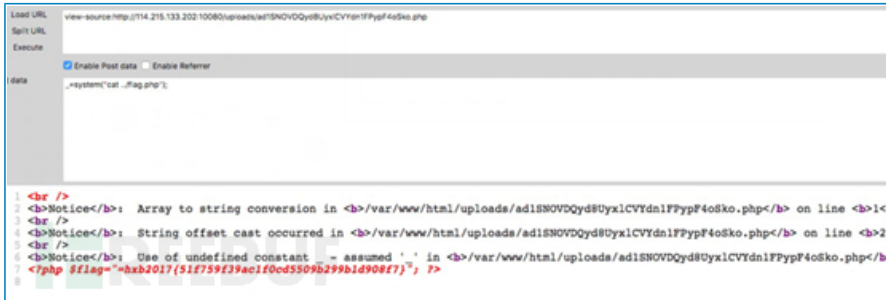
访问地址，直接给出源代码，过滤做的很全，并且只允许GET方式提交

参考文章：<https://www.leavesongs.com/PENETRATION/webshell-without-alphanum.html>

最后构造出不被拦截的payload, 进行url编码提交



如图上, 成功getshell (cat ./flag.php)



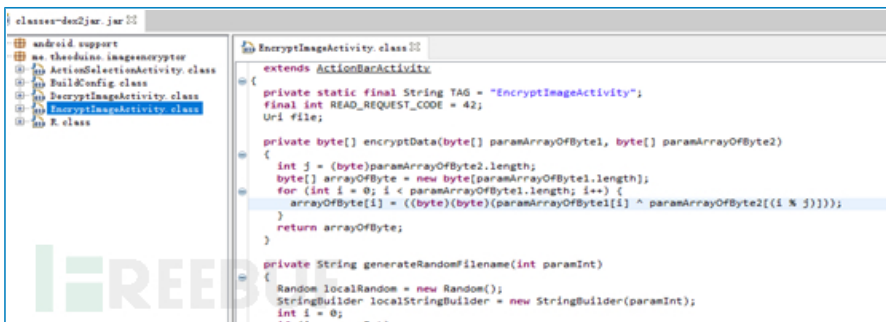
Encryptor.apk

题目:



过程:

通过分析源码, 发现程序需要输入一个密码和一张图片, 程序将图片与密码的MD5值逐位循环异或



直接给出解题代码


```

1 png_file = open("/Users/SPY/Desktop/own.png","rb")
2 png_h = png_file.read(16)
3 png_file.close()
4
5 flag_file = open("/Users/SPY/Desktop/flag.encrypted","rb")
6 flag_h = flag_file.read(16)
7 flag_file.close()
8
9 secret = []
10 for i in xrange(16):
11     secret.append(ord(png_h[i])^ord(flag_h[i]))
12
13 decrypt_file = open("/Users/SPY/Desktop/flag.png","wb")
14 with open("/Users/SPY/Desktop/flag.encrypted","rb") as f:
15     while True:
16         data = f.read(16)
17         if not data:
18             break
19         for i in xrange(len(data)):
20             decrypt_file.write(chr(ord(data[i])^secret[i]))
21     decrypt_file.close()

```

Flag{all_encryption_is_equal_but_some_are_More_equal_than_others}

MISC 300

题目:

MISC Misc300

分值: 300 类型: MISC 已解决

题目: 无

[附件下载](#)

flag格式: flag(ABC)仅填写ABC即可

过程:

下载后查看是一个pkl文件, 用pythonpickle模块打开, 发现如下

```

['The black pixels of a b/w image are at', (15, 200), (21, 300), (23, 310), (24, 314), (25, 300), (25, 318), (25, 431), (25, 441), (25, 451), (26, 310), (26, 320), (26, 429), (26, 439), (27, 449), (27, 459), (27, 315), (27, 325), (27, 423), (27, 433), (27, 443), (27, 453), (28, 303), (28, 313), (28, 323), (28, 333), (28, 419), (28, 429), (28, 439), (28, 449), (28, 459), (29, 304), (29, 314), (29, 324), (29, 334), (29, 417), (29, 427), (29, 454), (30, 128), (30, 138), (30, 311), (30, 333), (30, 343), (30, 414), (30, 424), (30, 456), (30, 471), (31, 130), (31, 309), (31, 349), (31, 414), (31, 460), (31, 474), (32, 129), (32, 343), (32, 353), (32, 363), (32, 411), (32, 401), (32, 471), (32, 481), (33, 135), (33, 346), (33, 356), (33, 366), (33, 411), (33, 464), (33, 474), (33, 484), (34, 305), (34, 318), (34, 352), (34, 362), (34, 372), (34, 382), (34, 411), (34, 478), (34, 485), (35, 307), (35, 317), (35, 327), (35, 337), (35, 347), (35, 357), (35, 379), (35, 405), (35, 466), (35, 484), (36, 306), (36, 316), (36, 326), (36, 336), (36, 346), (36, 356), (36, 400), (36, 466), (37, 125), (37, 307), (37, 327), (37, 337), (37, 347), (37, 357), (37, 398), (37, 438), (37, 486), (38, 305), (38, 334), (38, 346), (38, 356), (38, 366), (38, 396), (38, 425), (38, 435), (38, 448), (38, 487), (39, 306), (39, 375), (39, 402), (39, 431), (39, 441), (39, 485), (40, 137), (40, 375), (40, 394), (40, 417), (40, 427), (40, 437), (40, 447), (40, 487), (41, 306), (41, 383), (41, 414), (41, 424), (41, 434), (41, 444), (41, 485), (42, 305), (42, 396), (42, 415), (42, 425), (42, 444), (42, 478), (43, 137), (43, 397), (43, 417), (43, 427), (43, 447), (44, 128), (44, 394), (44, 414), (44, 442), (44, 479), (45, 306), (45, 397), (45, 417), (45, 446), (45, 484), (46, 389), (46, 409), (46, 442), (46, 487), (47, 390), (47, 409), (47, 48), (48, 137), (48, 394), (48, 430), (48, 485), (49, 387), (49, 485), (49, 456), (50, 131), (50, 391), (50, 409), (50, 485), (51, 388), (51, 487), (51, 485), (52, 387), (52, 395), (52, 482), (53, 241), (53, 393), (53, 483), (54, 387), (54, 394), (54, 486), (55, 237), (55, 389), (55, 430), (55, 448), (56, 137), (56, 387), (56, 427), (56, 437), (57, 70), (57, 30)

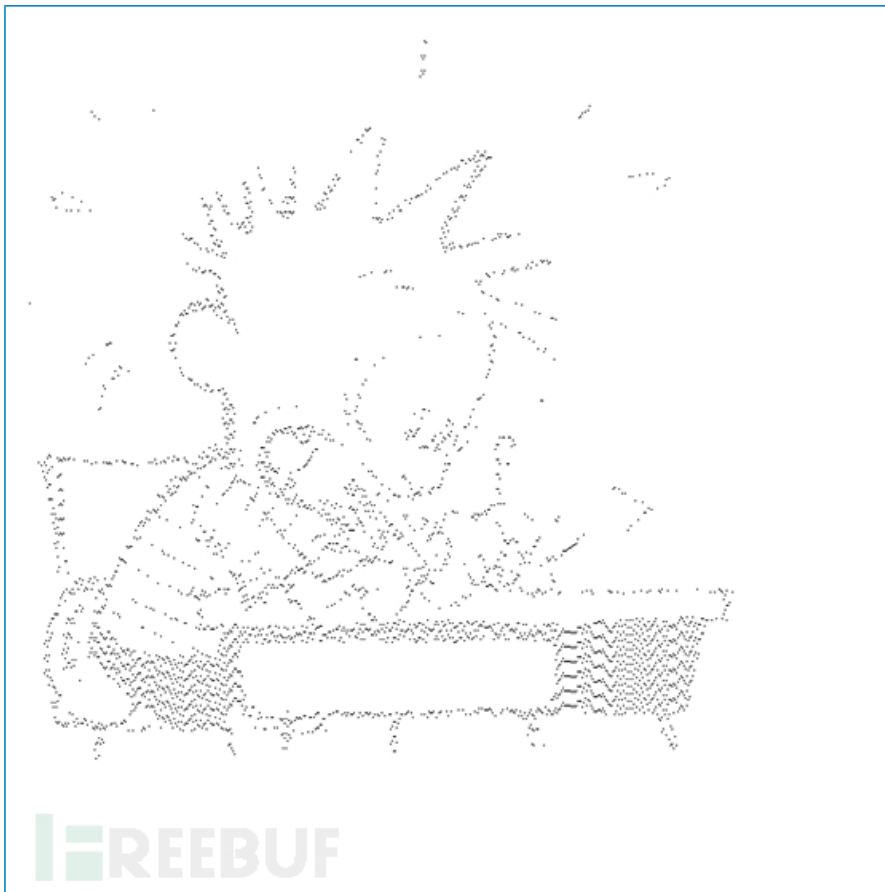
```

由第一句话得知 ()里面的数字是黑色像素的坐标, 然后通过脚本得到图片, 脚本以及图片如下

```

1 import pickle
2 from PIL import Image
3
4
5 f = open('pixels.jpg.pkl')
6 locates = pickle.load(f)
7
8 pic = Image.new('RGB', (600,600), "white")
9 pixels_out = pic.load()
10
11 for i in locates[1:]:
12     pixels_out[i[0],i[1]]=(0,0,0)
13
14 pic.save("misc300.png","png")

```



查看该图片发现图片为美国经典漫画《卡尔文与霍布斯虎》中的卡通人物，其中作者是比尔·沃特森（Bill Watterson）。根据ctf中的套路，将其改成小写的billwatterson提交flag。

热身运动（题目已更新）

过程：

图片背景是一个8X8的宫格，左上角第一个编号为0，横向编号，右下角最后一个编号为63，按照GIF中，老虎出现的顺序，转换为数字是：

25,38,49,33,25,55,44,49,29,5,60,49,13,21,61,38,29,22,57,46,30,23,52

8X8=64，联系到Base64,查看base64表，将数字转换为对应的字母或数字

Base64索引表:

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

转换之后是ZmxhZ3sxdF8xNV9mdW5ueX0，对该字符串使用base64解码，即得到flag

pyc分析

题目：pyc逆向分析，解密enc

过程：

将pyc用工具还原成py,发现是一行python代码

使用lamda和for var in value进行混淆，分析源码可知，该代码会丢失原始数据的最高两位，只有6位有效，但是通过转换的table的长度只有94，所以高8位是0，高7位可能为0或1，根据算法写出逆算法，分别求高7位为1和0的情况，代码如下图：


```

1 table='0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!@#$%^&*()-+=,./:;<=>?[\]^_`{|}~'
2 result1 = ''
3 result2 = ''
4 def decode(s):
5     global result1,result2
6     a = s>>16
7     b = (s>>8) & 0xFF
8     c = s & 0xFF
9     s1 = (8-len(bin(a[2:])))*'0'+bin(a[2:])
10    s2 = (8-len(bin(b[2:])))*'0'+bin(b[2:])
11    s3 = (8-len(bin(c[2:])))*'0'+bin(c[2:])
12    result1+=table[(int('1'+s1[2:],2)-1)%94]+table[(int('1'+s2[4:]+s1[2:],2)-1)%94]+table[(int('1'+s3[6:]+s2[4:],2)-1)%94]+table[(int(s1[2:],2)-1)%94]
13    result2+=table[(int(s1[2:],2)-1)%94]+table[(int(s2[4:]+s1[2:],2)-1)%94]+table[(int(s3[6:]+s2[4:],2)-1)%94]+table[(int(s3[6:],2)-1)%94]
14    # print int(s1[2:],2)-1,int('1'+s1[2:],2)-1
15    # print int(s2[4:]+s1[2:],2)-1,int('1'+s2[4:]+s1[2:],2)-1
16    # print int(s3[6:]+s2[4:],2)-1,int('1'+s3[6:]+s2[4:],2)-1
17    # print int(s3[6:],2)-1,int('1'+s3[6:],2)-1
18
19 a=open('key.enc','rb')
20 a=a.read()
21 for i in xrange(0, len(a), 3):
22     decode(int(a[i:i+3].encode('hex'), 16))
23
24 print result1
25 print result2

```

执行结果:

```

V1ANKing:PYC分析 SPY$ python decode.py
>>>>{{{g;4:,*&&/#*#.%$&+;(<)}}<<<<<<""""""""""""""""""""
hhhhhqqqqKeyd9733c070b2138e5fsssffffff~~~~~~

```

其中“hhhhhqqqqKeyd9733c070b2138e5fsssffffff”是高7位为0的情况。

“>>>>{{{g;4:,*&&/#*#.%\$&+;(<)}}<<<<<<xxxxxxxxxxxx”是高7位为1的情况。

根据题目，只取{}中的内容，为g;4:,*&&/#*#.%\$&+;(<或Keyd9733c070b2138e5f，根据Key分析，可能的原字符串为Key:9733c070b2138e5f,提交Key后面的内容，正确，所以9733c070b2138e5f即为flag

最后这次比赛还是学到了很多，欢迎各路师傅交流。

*本文作者：寻梦小生，转载请注明来自 FreeBuf.COM



寻梦小生 1 篇文章 等级： 1级

- 上一篇: [OpenATS续篇：搭建自己的卫星地球站](#)
- 下一篇: [认证云安全专家（CCSP）考试攻略](#)

这些评论亮了




[D14tr0y](#) (3级) 这家伙太懒了，还未填写个人描述! [回复](#)

freebuf文章图片这边看起来特别不舒服，图片是多大，打开就是多大图片都不能放大的。看都看不清)15 (亮了

[发表评论](#)

已有 3 条评论



• 常运  (5级) c4td0g, 信安从业者, 信安爱好者。(各位爷, 轻点喷) 2017-12-10 [回复 1楼](#)
咋writeup是抄来的, 图片都看不清额?

亮了(2)



• D14tr0y (3级) 这家伙太懒了, 还未填写个人描述! 2017-12-10 [回复 2楼](#)
freebuf文章图片这边看起来特别不舒服, 图片是多大, 打开就是多大图片都不能放大的。看都看不清

亮了(15)



• 142 2017-12-12 [回复 3楼](#)
基本上都是网上搬的原题, 原创的基本上没有