

BugKuCTF-杂项-writeup 大全?

原创

置顶 [疯狂棒棒糖7](#) 于 2018-06-20 11:14:36 发布 17765 收藏 49

分类专栏: [CTF](#) 文章标签: [writeup](#) [BugKuCTF](#) [杂项](#)

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/weixin_42373210/article/details/80743213

版权



[CTF 专栏收录该内容](#)

1 篇文章 1 订阅

订阅专栏

最近玩了一个 CTF 的练习平台-----BugkuCTF

下面会把一些题目的方法记录下来

先讲一下杂项

1.签到题

只要关注公众号就可以得到 flag---开胃菜

2.这是一张单纯的图片

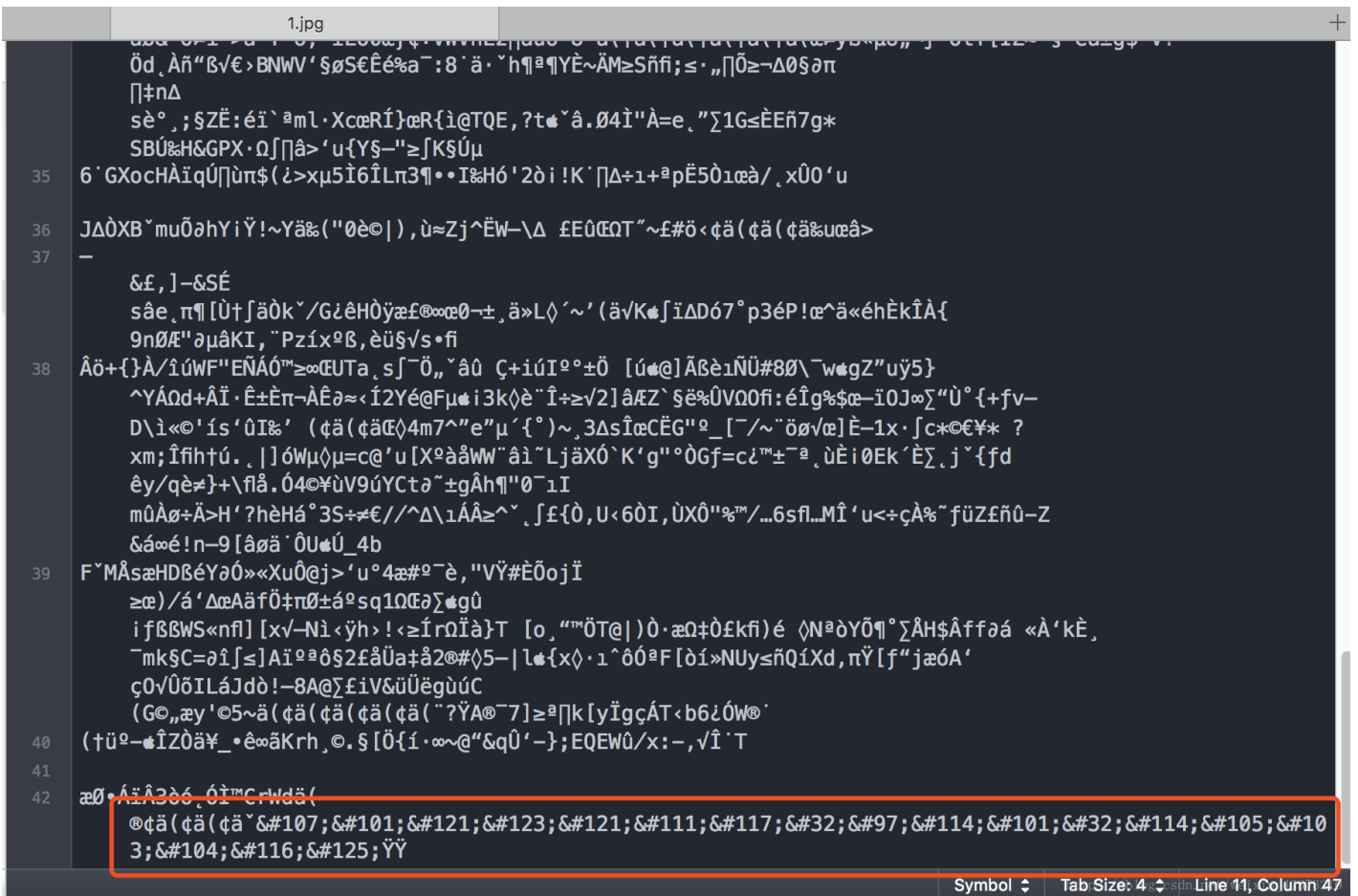
网站上打开是无法加载, 将图片保存到本地是个小兔子, 然后用编辑器打开图片, 发现了最下面一行的 unicode 编码, 解密就有 flag



1.jpg

136 × 152

blog.csdn.net/weixin_42373210



3. 隐写

下载2.rar，解压后得到2.png。

首先查看图片属性，没有找到 flag。

然后放到 binwalk 里面，发现有 zlib 的文件，但是没有发现别的信息。

最后放到 windows 里面用系统图片查看器可以正常打开图片，想到可能是照片的宽和高信息被改了，扔到 winhex 中尝试改宽高信息后发现是高被改了，将高修改为和宽一致即可得到 flag

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI	ASCII
00000000	89	50	4E	47	0D	0A	1A	07	00	00	00	0D	49	48	44	52	%PNG	IHDR
00000010	00	00	01	F4	00	00	01	F4	08	06	00	00	00	CB	D6	DF	ô	ô EÖß
00000020	8A	00	00	00	09	70	48	59	73	00	00	12	74	00	00	12	Š	pHYs t
00000030	74	01	DE	66	1F	78	00	00	0A	4D	69	43	43	50	50	68	t	Éf x MiCCPPh
00000040	6F	74	6F	73	68	6F	70	20	49	43	43	20	70	72	6F	66	o	toshop ICC prof
00000050	69	6C	65	00	00	78	DA	9D	53	77	58	93	F7	16	3E	DF	ile	xŪ SwX"÷ >ß
00000060	F7	65	0F	56	42	D8	F0	B1	97	6C	81	00	22	23	AC	08	÷e	VB0ð±-1 "#-
00000070	C8	10	59	A2	10	92	00	61	84	10	12	40	C5	85	88	0A	È	Yc ' a,, @Ä...^
00000080	56	14	15	11	9C	48	55	C4	82	D5	0A	48	9D	88	E2	A0	V	αHUA, Ö H ^ â
00000090	28	B8	67	41	8A	88	5A	8B	55	5C	38	EE	1F	DC	A7	B5	(αAS Zc U\81 ūSu

Bugku...

BUG { [REDACTED] }

https://blog.csdn.net/weixin_42373210

4.telnet

这个比较简单，因为之前玩过 wireshark所以下载下来的是networking.pacp，将这个数据包放到 wireshark中，任意一处右键追踪流，TCP 流，即可发现 flag

The screenshot shows the Wireshark interface with a packet list on the left and a context menu on the right. The packet list shows several packets, including TCP and TELNET. The context menu is open, and the '跟踪流' (Follow Stream) option is highlighted. A red box highlights the '跟踪流' option and the 'TCP 流' (Follow TCP Stream) option in the sub-menu.

No.	Time	Source	Destination	Protocol	Length	Info
164		66	1146 → 23	TCP		[SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
128		66	23 → 1146	TCP		[SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=32
164		54	1146 → 23	TCP		[ACK] Seq=1 Ack=1 Win=65536 Len=0
164		75		TELNET		Telnet Data ...
128		60	23 → 1146	TCP		[ACK] Seq=1 Ack=22 Win=14624 Len=0
128		66		TELNET		Telnet Data ...
164		57		TELNET		Telnet Data ...

Frame 3: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)

- Ethernet II, Src: Vmware_84:86:5f (00:0c:29:84:86:5f), Dst: Vmware_26:7e
- Internet Protocol Version 4, Src: 192.168.221.128, Dst: 192.168.221.164
- Transmission Control Protocol, Src Port: 1146, Dst Port: 23, Seq: 1, Ack

0000 00 0c 29 26 7e 0e 00 0c 29 84 86 5f 08 00
0010 00 28 07 99 40 00 80 06 00 00 c0 a8 dd 8
0020 dd a4 04 7a 00 17 46 01 d3 fc 68 f0 29 f

跟踪流

- TCP 流 Ctrl+Alt+Shift+T
- UDP 流 Ctrl+Alt+Shift+U
- SSL 流 Ctrl+Alt+Shift+S
- HTTP 流 Ctrl+Alt+Shift+H

https://blog.csdn.net/weixin_42373210

```
Wireshark · 追踪 TCP 流 (tcp.stream eq 0) · networking.pcap
.....'.....#..'..#.....P.....'.....
.38400,38400.....XTERM.....!.....!Ubuntu 12.04.2 LTS
hockeyinjune-virtual-machine login: ccssaaww

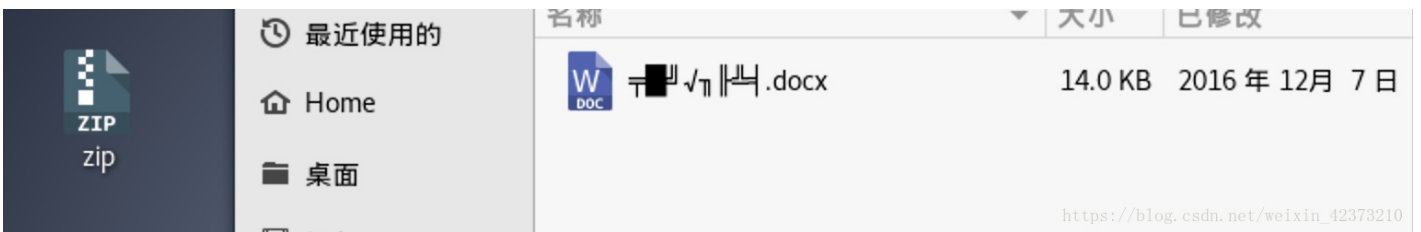
Password: flag: [REDACTED]

Login incorrect
hockeyinjune-virtual-machine login: .
...^C
```

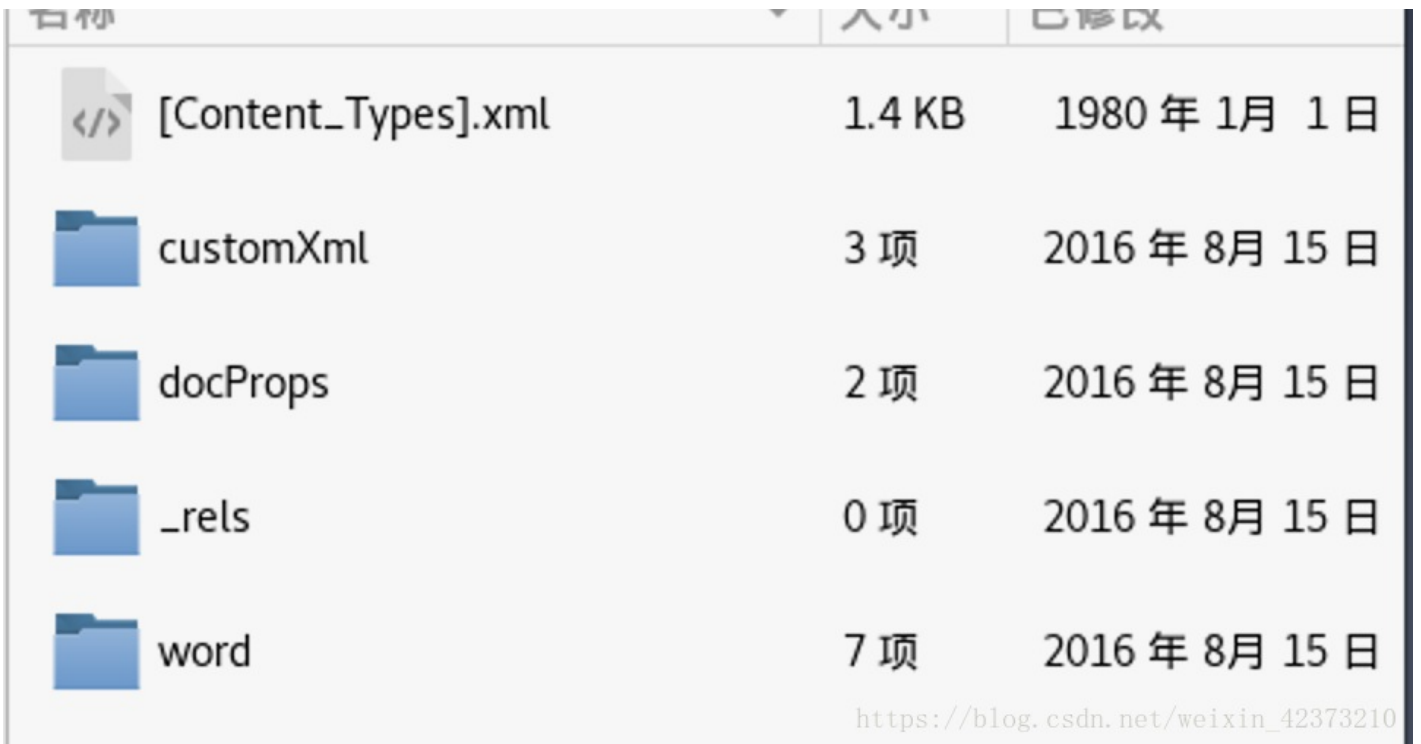
https://blog.csdn.net/weixin_42373210

5.眼见非实 (ISCCTF)

这套题下载 zip 压缩包后放到kali 里面解压



发现是一个破损的 docx，然后用 binwalk 查看后发现是个压缩文件，改后缀解压后得到一个文件夹



打开第一个文件，发现其中有很多partName 后写的是目录下的xml 文件，感觉有个能某一个文件里面有 flag

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Types xmlns="http://schemas.openxmlformats.org/package/2006/content-types"><Default
Extension="rels" ContentType="application/vnd.openxmlformats-package.relationships+xml"/><Default
Extension="xml" ContentType="application/xml"/><Override PartName="/word/document.xml"
ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.document.main+xml"/
><Override PartName="/customXml/itemProps1.xml" ContentType="application/vnd.openxmlformats-
officedocument.customXmlProperties+xml"/><Override PartName="/word/styles.xml"
ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.styles+xml"/><Override
PartName="/word/settings.xml" ContentType="application/vnd.openxmlformats-
officedocument.wordprocessingml.settings+xml"/><Override PartName="/word/webSettings.xml"
ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.webSettings+xml"/
><Override PartName="/word/fontTable.xml" ContentType="application/vnd.openxmlformats-
officedocument.wordprocessingml.fontTable+xml"/><Override PartName="/word/theme/theme1.xml"
ContentType="application/vnd.openxmlformats-officedocument.theme+xml"/><Override PartName="/
docProps/core.xml" ContentType="application/vnd.openxmlformats-package.core-properties+xml"/
><Override PartName="/docProps/app.xml" ContentType="application/vnd.openxmlformats-
officedocument.extended-properties+xml"/></Types>
```

https://blog.csdn.net/weixin_42373210

先打开 document.xml 果然发现了 flag

```
schemas.microsoft.com/office/word/2010/wordprocessingml" xmlns:wps="http://schemas.microsoft.com/
office/word/2006/wordml" xmlns:wmc="http://schemas.microsoft.com/office/word/2010/
wordprocessingShape" mc:Ignorable="w14 w15 wp14"><w:body><w:p w:rsidR="002B3D8D"
w:rsidRDefault="002B3D8D"><w:r><w:t>Flag</w:t></w:r><w:r><w:t>在这里哟！</w:t></w:r></w:p><w:p
w:rsidR="002B3D8D" w:rsidRPr="002B3D8D" w:rsidRDefault="002B3D8D"><w:pPr><w:rPr><w:rFonts
w:hint="eastAsia"/><w:vanish/></w:rPr></w:pPr><w:r w:rsidRPr="002B3D8D"><w:rPr><w:vanish/></
w:rPr><w:t>flag</w:t></w:r><w:bookmarkStart w:id="0" w:name="_GoBack"/><w:bookmarkEnd
w:id="0"/></w:p><w:sectPr w:rsidR="002B3D8D" w:rsidRPr="002B3D8D"><w:pgSz w:w="11906" w:h="16838" /
><w:pgMar w:top="1440" w:right="1800" w:bottom="1440" w:left="1800" w:header="851" w:footer="992" /
w:gutter="0"/><w:cols w:space="425"/><w:docGrid w:type="lines" w:linePitch="312"/></w:sectPr></
w:body></w:document>
```

https://blog.csdn.net/weixin_42373210

6. 又一张图片，还单纯吗

这次应对这个照片用到了 StegSolve 工具。用 Analyse-->Frame Browser 可以看到有 2 帧的图片，第二个就是 flag


```
RouterPassView - C:\Users\boom\Desktop\conf.bin
文件(F) 编辑(E) 查看(V) 选项(O) 帮助(H)
<SubnetMask val=0.0.0.0 />
<DefaultGateway val=0.0.0.0 />
<DNSServers val=0.0.0.0,0.0.0.0 />
<MACAddress val=D0:C7:C0:43:53:69 />
<X_TP_IfName val=eth1 />
</WANIPConnection>
<WANIPConnection nextInstance=3 />
<WANPPPConnection instance=1 >
  <Enable val=1 />
  <DefaultGateway val=10.177.144.1 />
  <Name val=pppoe_eth1_d />
  <Uptime val=671521 />
  <Username val=053700357621 />
  <Password val=210265 />
```

9.隐写2

如果完成了隐写1那么到了这里应该就能有基本的思路，将welcome_.jpg放到 binwalk 中可以看到图片里面含有 zip 压缩文件，并且发现了 flag.rar的文件。证明找对了！

```
root@Crazy-kali:~# binwalk '/root/桌面/Welcome_.jpg'
DECIMAL skip HEXADECIMAL DESCRIPTION
-----
0      root@C 0x00 图片 JPEG image data, JFIF standard 1.01
30     found 0x1E TIFF image data, big-endian, offset of first image
directory: 8aaa:
4444   root@C 0x115C 文档 Unix path: /www.w3.org/1999/02/22-rdf-syntax-ns#>
<rdf:Description rdf:about="uuid:faf5bdd5-ba3d-11da-ad31-d33d75182f1b" xmlns:dc=
"http://p
4900   you 0x1324 Unix path: /www.w3.org/1999/02/22-rdf-syntax-ns#>
<rdf:li xml:lang="x-default">hint:</rdf:li></rdf:Alt>
52516  aaaa 0xCD24 Zip archive data, at least v1.0 to extract, compressed size: 6732, uncompressed size: 6732, name: flag.rar
59264  found 0xE780 End of Zip archive
147852 aaaa 0x2418C Home End of Zip archive
```

这里使用foremost工具分离，分离后在 home 目录下的output文件夹里，经过又一层解压缩后找到了 flag.rar和提示.jpg。

首先打开提示.jpg，开始划重点，没错就是这一个信息，其余都是无用信息。

告诉你们一个秘密，**密码是3个数**哦。

查理曼：

查理曼，法兰克王国国王，征服了西欧与中欧大部分土地，具有了至高无上的权威，下令全国人民信仰基督教，查理重振了西罗马帝国。

雅典娜：

女神帕拉斯·雅典娜，是希腊神话中的女战神也是智慧女神，雅典是以她命名的。

兰斯洛特，

英格兰传说中的人物，是亚瑟王圆桌骑士团中的一员。看上去就是一个清秀年轻的帅小伙儿，由于传说中他是一名出色的箭手，因此梅花J手持箭支。兰斯洛特与王后的恋爱导致了他与亚瑟王之间的战争。

Hint:

其实斗地主挺好玩的。

https://blog.csdn.net/weixin_42373210

接下来生成一个3位数的字典，考虑到接下来可能还会用到此字典，就写了个 python 脚本来生成字典，很简洁很高效，可修改位数

```
import string
s = string.digits
f = open('evil.txt', 'w')
for i in s:
    for j in s:
        for k in s:
            f.write(i+j+k+'\n')
f.close()
```

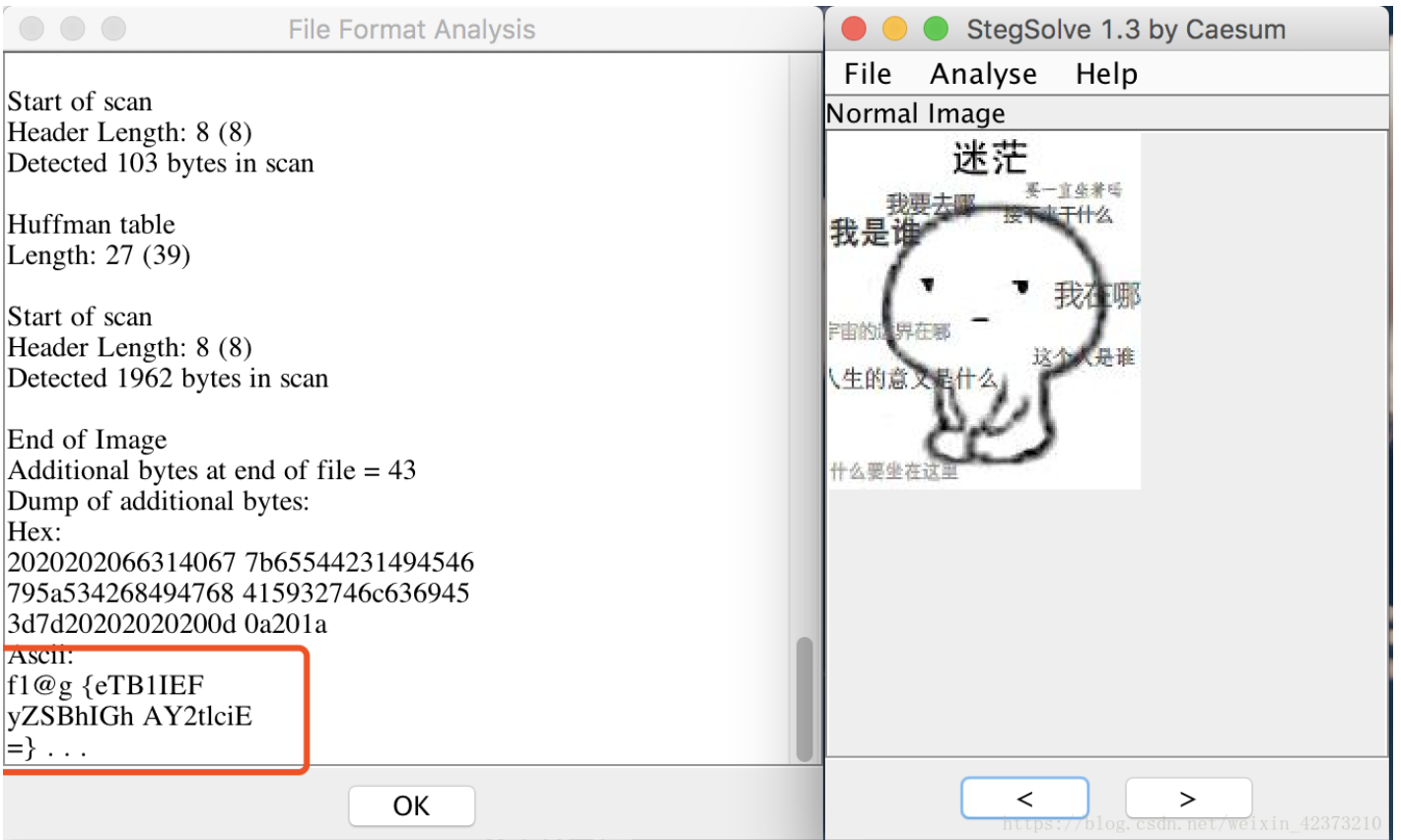
之后用kali 自带的fcrackzip工具解密，得到解密密码871。

这里另外解释一下 fcrackzip 命令的含义：

-D 就是用的字典模式 -p指定起始破解密码 -u这个参数是为了显示密码用 -v是展示更多信息

```
root@Crazy-kali:~/文档/zip口令破解测试# fcrackzip -D -p evil.txt -u flag.zip -v
found file '3.jpg', (size cp/uc 6588/ 6769, flags 801, chk 102c)
PASSWORD FOUND!!!!: pw == 871
```

输入密码解压后得到一个3.jpg，用StegSolve 工具的file format进行分析，发现Ascii码处出现flag



别以为到这里就完了，这个 flag 是一个 base64加密，经过解密后获得真正的 flag



10. 多种方法解决

本题我用的方法如下：

首先下载了一个 zip 文件，解压缩后得到KEY.exe, 将 exe 扩展名改为 txt 后可以看到是base64加密的密文

```
data:image/jpeg;base64,iVBORw0KGgoAAAANSU... [truncated] ...sbyF93Z9h6Xu095mtP3ec8Klrfw3q7vcPXy+c1Pc/o+75eE8be2/Udzv9X+sv/OF/881/3qtvcdbh+AAAABJRU5ErkJggg==
```

用在线解密工具将base64密文还原为图片，发现了和本题提示相同的二维码，用手机扫描二维码后得到 flag

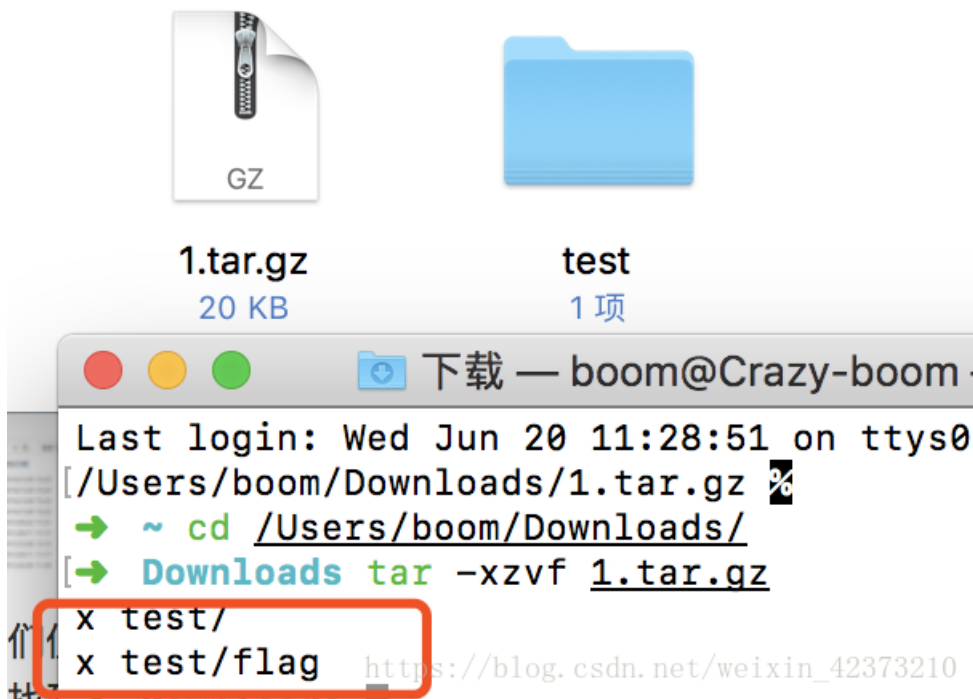
以下是您的 Base64 代码所解码出来的图片，右键另存为保存图片。



https://blog.csdn.net/weixin_42373210

11. linux

下载下来本题的1.tar.gz 压缩包，提示为 linux 基础问题，所以直接解压得到了文件夹里面是 flag



既然是 linux 基础问题，接着用 cd 到 test 目录下用 vi flag 即可打开 flag，文件比较大最好用 vi 命令，然后一点点浏览即可找到 key。ps：这里需要把命令终端窗口最大化才能看到 key，不然就被遮挡住了。



12. 中国菜刀

国产神器啊哈哈，下载好压缩文件后，解压得到 caidao.pcapng，不用说又是数据包，放到 wireshark 中追踪数据流可以看到

```
POST /3.php HTTP/1.1
X-Forwarded-For: 241.38.53.25
Referer: http://192.168.1.145/
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)
Host: 192.168.1.145
Content-Length: 774
Cache-Control: no-cache
```

```
123=array_map("ass"."ert",array("ev"."Al("\\\\\\"$xx%3D\\"Ba"."SE6"."4_dEc"."0dE\\\\";"@ev"."al(\\\\"$xx('QGluaV9zZXQoImRpc3BsYX1fZXJyb3JzIiwuMCIp00BzZXRfdGltZV9saW1pdCgwKTtpZihQSFbFVkvSU01PTjwnNS4zLjAnKXtAc2V0X21hZ21jX3F1b3Rlc19ydW50aW11KDAp0307ZWNoBygiWEBZiik7JEQ9J0M6FX3d3dyb290XFwnOyRGPUBvcGVuZGlyKCREKTtpZigkrj09T1VMTC17ZWNoBygiRVJST1I6Ly8gUGF0aCB0b3QGRm91bmQgT3Igtm8gUGVybWlzc2lubiEiKTt9ZWxzZXskTT10VUxMOyRMPU5VTWV7d2hpbGUoJE49QHJlYWRkaXI0JEYpKXskUD0kRc4nLycuJE47JFQ9QGRhdGUoIlktdS1kIEg6aTtzIixAZmlsZW10aW11KCRQKS7QCkRFPXN1YnN0cihiYXNlX2NvbzZlcnQoQGZpbGVwZXJtcyZkUcksMTAsO0ksLTQpOyRSPSJCdCIuJFQuIlx0Ii5AZmlsZXNpemUoJFAPLiJcdCIuJEUuIlxuIjtpZihAaXNfZGlyKCRQKSskTS49JE4uIi8iLiRSO2Vsc2UgJEwuPSROLiRSO31lY2hvICRNLiRMO0BjbG9zZWRpcigkrIk7FTt1Y2hvKCJYQFkiKTtkaWUoKTs%3D'));"");HTTP/1.1
200 OK
```

https://blog.csdn.net/weixin_42373210

第一时间还去解密了 base64，发现是一些配置文件，回去继续往下浏览发现了重点-flag.tar.gz。

```
1.php      2016-01-28 08:54:46 1740      0666
3.php      2016-06-01 03:36:25 27         0666
flag.tar.gz      2016-06-27 08:45:38 203        0666
log.txt    2015-06-03 12:18:46 1502      0666
```

然后想到会不会是隐藏了压缩包文件，于是把数据包文件放到 binwalk 看一下，发现了在7747块偏移后是一个zip 的压缩包

```
root@Crazy-kali:~# binwalk -e '/root/桌面/caidao.pcapng'
```

DECIMAL	HEXADECIMAL	DESCRIPTION
7747	0x1E43	gzip compressed data, from Unix, last modified: 2016-06-27 08:44:39

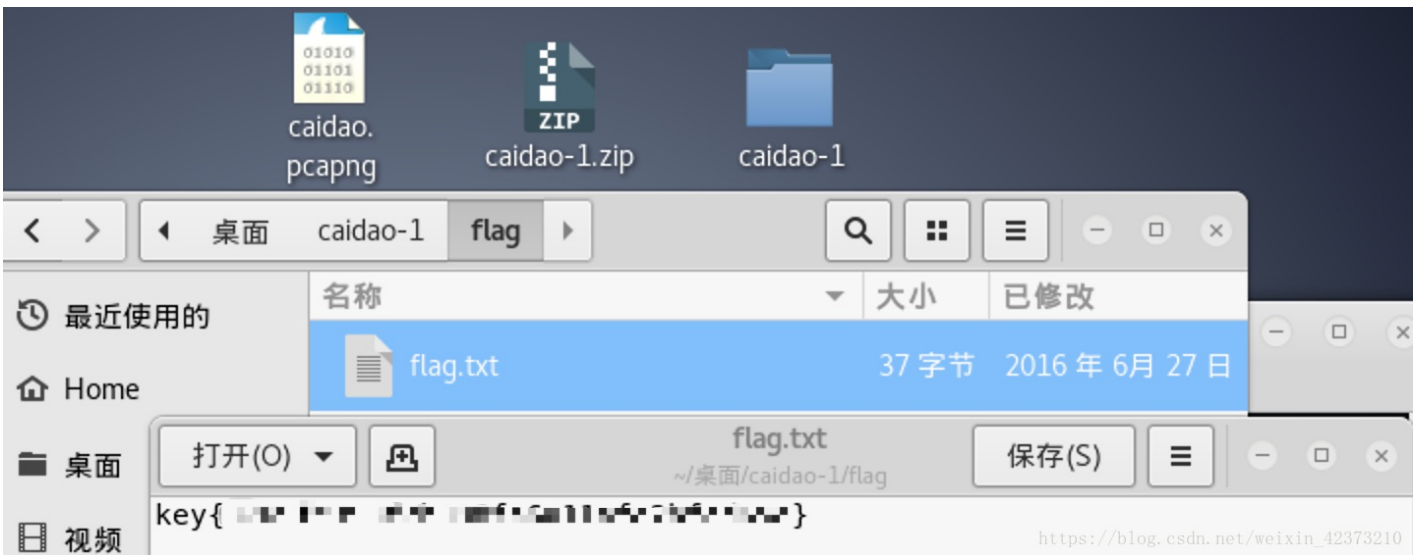
https://blog.csdn.net/weixin_42373210

用命令开始分离压缩包，用dd命令分离 dd if=caidao.pcapng of=caidao-1.zip skip=7747 bs=1

```
root@Crazy-kali:~/桌面# dd if=caidao.pcapng of=caidao-1.zip skip=7747 bs=1
记录了301+0 的读入
记录了301+0 的写出
301 bytes copied, 0.00119799 s, 251 kB/s
```

https://blog.csdn.net/weixin_42373210

得到压缩包后解压，获得 flag



13. 这么多数据包

下载了数据包后放到 Wireshark 中去解析，发现真的有很多数据包...

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.116.159	192.168.116.2	NBNS	110	Refresh NB <01><02>_MSBROWSE_<02><01>
2	0.640205	Vmware_c0:00:08	Broadcast	ARP	42	Who has 192.168.116.2? Tell 192.168.116.1
3	2.039844	Vmware_c0:00:08	Broadcast	ARP	42	Who has 192.168.116.2? Tell 192.168.116.1
4	2.640022	Vmware_c0:00:08	Broadcast	ARP	42	Who has 192.168.116.2? Tell 192.168.116.1
5	3.062223	192.168.116.138	91.189.94.4	NTP	90	NTP Version 4, client
6	3.440397	91.189.94.4	192.168.116.138	NTP	90	NTP Version 4, server
7	3.640410	Vmware_c0:00:08	Broadcast	ARP	42	Who has 192.168.116.2? Tell 192.168.116.1
8	8.050266	Vmware_ca:16:94	Vmware_f1:eb:e0	ARP	60	Who has 192.168.116.2? Tell 192.168.116.138
9	8.050294	Vmware_f1:eb:e0	Vmware_ca:16:94	ARP	42	192.168.116.2 is at 00:50:56:f1:eb:e0
10	18.581648	192.168.116.138	192.168.116.159	ICMP	98	Echo (ping) request id=0xa6f7, seq=1/256, ttl=64 (reply in 11)
11	18.581742	192.168.116.159	192.168.116.138	ICMP	98	Echo (ping) reply id=0xa6f7, seq=1/256, ttl=128 (request in 10)
12	19.582551	192.168.116.138	192.168.116.159	ICMP	98	Echo (ping) request id=0xa6f7, seq=2/512, ttl=64 (reply in 13)
13	19.582629	192.168.116.159	192.168.116.138	ICMP	98	Echo (ping) reply id=0xa6f7, seq=2/512, ttl=128 (request in 12)
14	20.583601	192.168.116.138	192.168.116.159	ICMP	98	Echo (ping) request id=0xa6f7, seq=3/768, ttl=64 (reply in 15)
15	20.583748	192.168.116.159	192.168.116.138	ICMP	98	Echo (ping) reply id=0xa6f7, seq=3/768, ttl=128 (request in 14)
16	21.584729	192.168.116.138	192.168.116.159	ICMP	98	Echo (ping) request id=0xa6f7, seq=4/1024, ttl=64 (reply in 17)
17	21.584827	192.168.116.159	192.168.116.138	ICMP	98	Echo (ping) reply id=0xa6f7, seq=4/1024, ttl=128 (request in 16)
18	31.787200	192.168.116.138	224.0.0.251	MDNS	88	Standard query 0x0001 PTR _services._dns-sd._udp.local, "QM" ques

按照本题的提示先找到 getshell 的流，开始寻找ing。发现从104开始就是 TCP 一连串的端口扫描，继续看下去发现了比较有意思的。

5574	407.125895	192.168.116.138	192.168.116.159	TCP	106	35880 → 1234 [PSH, ACK] Seq=24 Ack=822 Win=32768 Len=40 TSval=167...
5575	407.179724	192.168.116.159	192.168.116.138	TCP	107	1234 → 35880 [PSH, ACK] Seq=822 Ack=64 Win=64177 Len=41 TSval=305...
5576	407.179800	192.168.116.138	192.168.116.159	TCP	66	35880 → 1234 [ACK] Seq=64 Ack=863 Win=32768 Len=0 TSval=1675039 T...
5577	407.206204	192.168.116.159	192.168.116.2	NBNS	92	Name query NB STUPID MANAGER!<20>
5578	407.240667	192.168.116.159	192.168.116.2	DNS	87	Standard query 0x0188 A Stupid Manager!.localdomain
5579	408.239702	192.168.116.159	192.168.116.2	DNS	87	Standard query 0x0188 A Stupid Manager!.localdomain
5580	408.699108	192.168.116.159	192.168.116.2	NBNS	92	Name query NB STUPID MANAGER!<20>
5581	409.100830	Vmware_c0:00:08	Broadcast	ARP	42	Who has 192.168.116.2? Tell 192.168.116.1
5582	409.417150	192.168.116.138	192.168.116.159	TCP	208	Application Data Application Data

追踪一下 TCP 流，可以判断该攻击开始于5542，在下面发现base64加密的文件，不出意外应该是 flag。

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
```

```
C:\>ls
```

```
ls
```

```
'ls' is not recognized as an internal or external command,
operable program or batch file.
```

```
C:\>dir
```

```
dir
```

```
Volume in drive C has no label.
Volume Serial Number is B03C-791A
```

```
Directory of C:\
```

```
04/14/2016  08:50 PM          0 AUTOEXEC.BAT
04/14/2016  08:50 PM          0 CONFIG.SYS
```

```
04/14/2016  08:52 PM    <DIR>          Documents and Settings
03/12/2012  10:24 PM          61,454 nc.exe
04/14/2016  08:54 PM    <DIR>          Program Files
04/14/2016  09:22 PM          36 s4cr4t.txt
04/14/2016  08:59 PM    <DIR>          WINDOWS
               4 File(s)          61,490 bytes
               3 Dir(s)  17,719,083,008 bytes free
```

```
C:\>type s4cr4t.txt
```

```
type s4cr4t.txt
```

```
Q0NURntkb195b3Vib3Vib3Vib3Vib3VyfQ==
```

```
C:\>shutdown -r -t 100 -m "Stupid Manager!"
```

```
shutdown -r -t 100 -m "Stupid Manager!"
```

果然解密后出现 flag。

```
CCTF
```

```
Q0NURntkb195b3Vib3Vib3Vib3VyfQ==
```

14. 隐写3

下载图片后在 mac 中发现不能打开 png 图片，而在 windows 里面可以直接打开，从而根据之前的题型判断图片被改了高。



然后用 winhex 打开图片，不断改高值，发现图片慢慢变高，最后修改为C7的时候发现了 flag。

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49
00000016	00	00	02	A7	00	00	01	C7	08	06	00	00	00
00000032	35	00	00	00	01	73	52	47	42	00	AE	CE	1C
00000048	00	04	67	41	4D	41	00	00	B1	8F	0B	FC	61

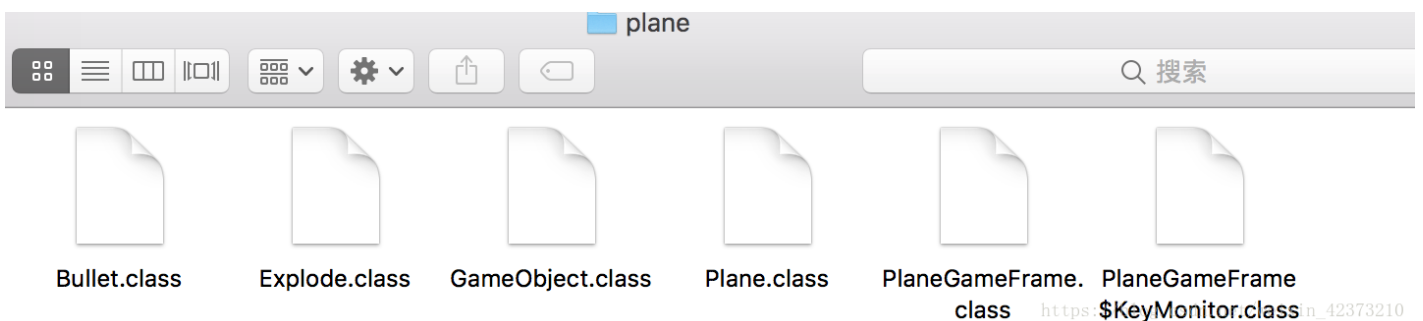


15做个游戏 (08067CTF)

下载一个.jar 的文件，提示坚持60秒

打开是一个游戏，然后发现坚持到60秒太困难了，索性放到 windows 里用 winhex 查看一下文件，发现头文件是PK，是压缩文件，将后缀修改后可以解压。

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI	ASCII
00000000	50	4B	03	04	14	00	08	08	08	00	AC	86	52	4B	00	00	PK	-+RK
00000016	00	00	00	00	00	00	00	00	00	00	14	00	04	00	4D	45		ME
00000032	54	41	2D	49	4E	46	2F	4D	41	4E	49	46	45	53	54	2E		TA-INF/MANIFEST.
00000048	4D	46	FE	CA	00	00	F3	4D	CC	CB	4C	4B	2D	2E	D1	0D		MFPÊ óMÏELK-.Ñ
00000064	4B	2D	2A	CE	CC	CF	B3	52	30	D4	33	E0	E5	72	CE	49		K-*ÏÏ°ROÛ3àârÏI
00000080	2C	2E	D6	0D	48	2C	C9	B0	52	D0	E3	E5	F2	4D	CC	CC		,.Ö H,É°RDããòMÏi
00000096	D3	05	8B	59	29	24	E7	E9	25	65	15	57	94	E8	15	E4		Ó<Y)šçése.W"è ä



解压后发现有一个文件夹里是 java 程序的 class 文件，需要java反编译来查看一下，百度工具由于是 macOS 系统就选择了这款

Java反编译工具-JD-GUI - EasonJim - 博客园

2017年11月5日 - Java反编译工具-JD-GUI Java是跨平台的,JD-GUI提供了多个系统的支持,但是不建议直接安装,最快的方式推荐直接下载JAR包,然后用java -jar进行运行。就...

<https://www.cnblogs.com/EasonJ...> - 百度快照

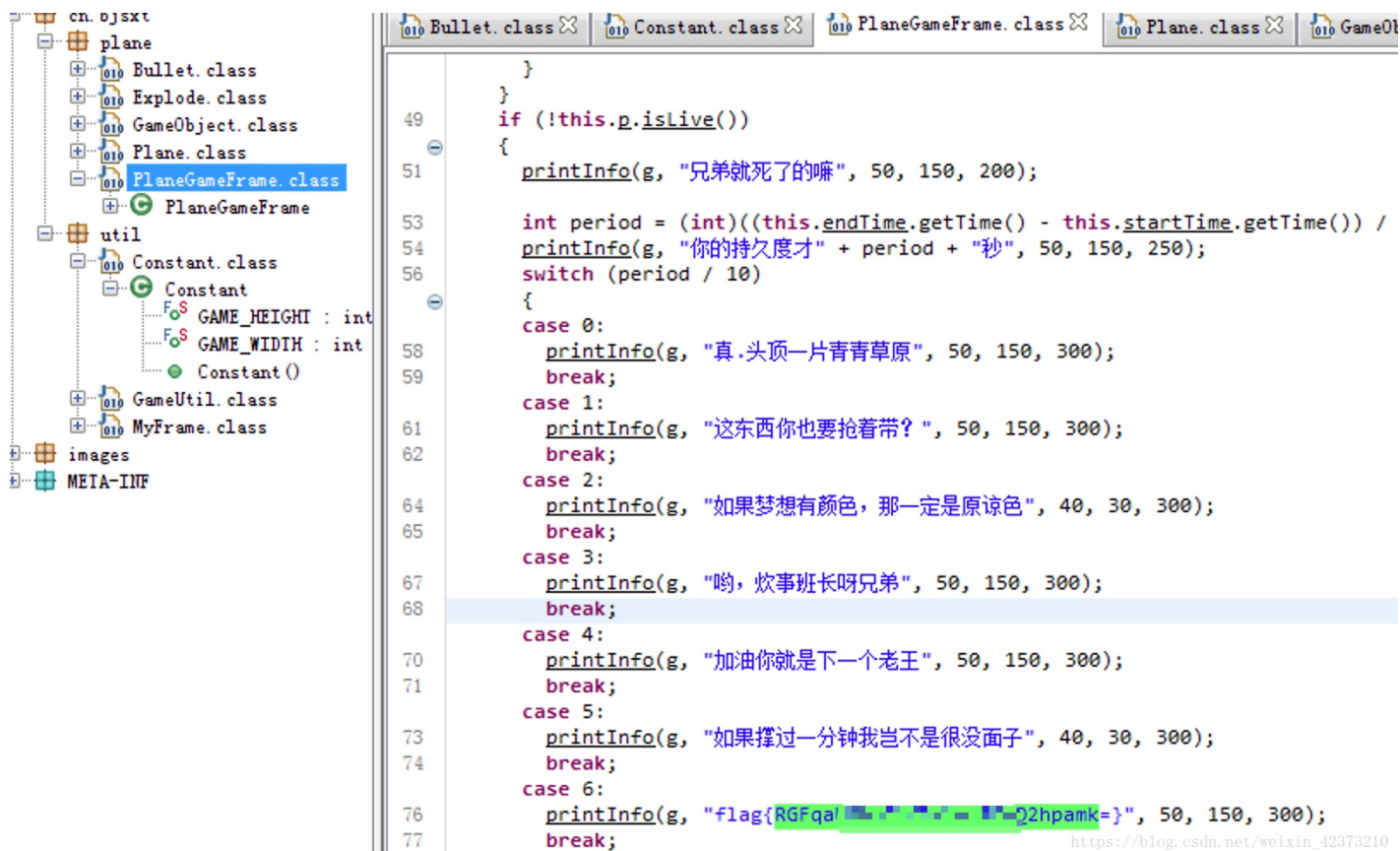
7款开源Java反编译工具 - CSDN博客

2017年2月25日 - 今天我们要来分享一些关于Java的反编译工具,反编译听起来是一个非常高大上的技术词汇,通俗的说,反编译是一个对目标可执行程序进行逆向分析,从而得到原...

<https://blog.csdn.net/r3t7o7/a...> - 百度快照

https://blog.csdn.net/weixin_42373210

下载好之后打开 class 文件,挨个找后在PlaneGameFrame.class中的 printInfo 中找到了 flag。(flag经过 Base64加密)



```
    }
  }
  if (!this.p.isLive())
  {
    printInfo(g, "兄弟就死了的嘛", 50, 150, 200);

    int period = (int)((this.endTime.getTime() - this.startTime.getTime()) /
    printInfo(g, "你的持久度才" + period + "秒", 50, 150, 250);
    switch (period / 10)
    {
      case 0:
        printInfo(g, "真.头顶一片青青草原", 50, 150, 300);
        break;
      case 1:
        printInfo(g, "这东西你也要抢着带?", 50, 150, 300);
        break;
      case 2:
        printInfo(g, "如果梦想有颜色,那一定是原谅色", 40, 30, 300);
        break;
      case 3:
        printInfo(g, "哟,炊事班长呀兄弟", 50, 150, 300);
        break;
      case 4:
        printInfo(g, "加油你就是下一个老王", 50, 150, 300);
        break;
      case 5:
        printInfo(g, "如果撑过一分钟我岂不是没面子", 40, 30, 300);
        break;
      case 6:
        printInfo(g, "flag{RGFqa! - 2hpamk=}", 50, 150, 300);
        break;
    }
  }
}
```

16. 想蹭网先解开密码

首先下载wifi.cap 文件,密码为手机号,给了前7位1391040,电话号码一共11位,用脚本生成一个四位数的密码然后加入前缀就是我们的密码字典文件了。

```
evil.txt x
1 13910400000
2 13910400001
3 13910400002
4 13910400003
5 13910400004
6 13910400005
7 13910400006
8 13910400007
9 13910400008
10 13910400009
11 13910400010
12 13910400011/blog.csdn.net/weixin_42373210
```

这里可以利用wifi.cap专用破解工具aircrack-ng -w "字典文件" ".cap 文件"进行破解 wifi.cap 文件密码。得到 KEY。

```

root@Crazy-kali:~/桌面# aircrack-ng -w evil.txt wifi.cap
Opening wifi.cap
Read 4257 packets.

# BSSID          ESSID          Encryption
1  3C:E5:A6:20:91:60  CATR          No data - WEP or WPA
2  3C:F5:A6:20:91:61  CATR-GUEST    None (10 2 28 31)
3  BC:F6:85:9E:4E:A3  D-Link_DIR-600A  WPA (1 handshake)

Index number of target network ? 3

Opening wifi.cap
Reading packets, please wait...
Aircrack-ng 1.2

[00:00:00] 7680/9999 keys tested (8265.30 k/s)

Time left: 0 seconds                                76.81%

KEY FOUND! [ 139104 ]

Master Key      : C4 60 FE 8B 14 7D 58 00 91 D7 0A 9C 3C DE 44 69
                  0B E1 CD 81 07 F8 28 DB EA 76 1E ED 81 A3 FF FD

Transient Key   : 0D 88 B3 F4 BC A3 C9 D2 06 12 28 43 FF 5E 21 3E
                  F5 23 8E 0B 7A 9F 25 59 E9 7C 86 1E 7A 78 E4 D4
                  D3 62 CD DD 4D 87 80 EE B9 E1 16 91 4A 6E 3E 09
                  1E CE 5E 62 38 3C 05 35 34 A6 EB 16 31 D8 CE 96

EAPOL HMAC     : 1C E7 D0 96 DE 87 93 56 88 1D 08 C8 B9 AA B3 B0

```

17. linux2

和之前linux一样的思路，这次用 notepad++打开 搜索一下 key 即可，丝毫不和 linux 无关。

