




# MISC 2020-12-26任务笔记

原创

[Mariamoon](#)  于 2021-01-02 11:27:48 发布  208  收藏

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/Mariamoon/article/details/112093676>

版权

攻防世界misc答题：

题目一：

**坚持60s** 👍 12 最佳Writeup由 [不要让我起名提供](#)

难度系数: ★★★★ 4.0

题目来源: 08067CTF

题目描述: 菜狗发现最近菜猫不爱理他, 反而迷上了菜鸡

题目场景: 暂无

题目附件: 附件1

<https://blog.csdn.net/Mariamoon>

下载附件发现是一个jar文件, 查看源代码, 使用 jd-gui可得flag, 但发现格式不对, 需要采用64base进行编译, 才能得出正确的flag

```

53     int period = (int)((this.endTime.getTime() - this.startTime.getTime()) / 1000L);
54     printInfo(g, "你的持久度才" + period + "秒", 50, 150, 250);
56     switch (period / 10)
57     {
58     case 0:
59         printInfo(g, "真.头顶一片青青草原", 50, 150, 300);
60         break;
61     case 1:
62         printInfo(g, "这东西你也要抢着带?", 50, 150, 300);
63         break;
64     case 2:
65         printInfo(g, "如果梦想有颜色, 那一定是原谅色", 40, 30, 300);
66         break;
67     case 3:
68         printInfo(g, "哟, 炊事班长呀兄弟", 50, 150, 300);
69         break;
70     case 4:
71         printInfo(g, "加油你就是下一个老王", 50, 150, 300);
72         break;
73     case 5:
74         printInfo(g, "如果撑过一分钟我岂不是很没面子", 40, 30, 300);
75         break;
76     case 6:
77         printInfo(g, "Flag{RGFqaURhbG1fSm1ud2FuQ2hpamk=}", 50, 150, 300);
78         break;
79     }
80 }
81 }

public void printInfo(Graphics g, String str, int size, int x, int y)
{

```

## 题目二

**SimpleRAR** 👍 61 最佳Writeup由 [它山提供](#)

难度系数: ★★★★★ 5.0

题目来源: 08067CTF

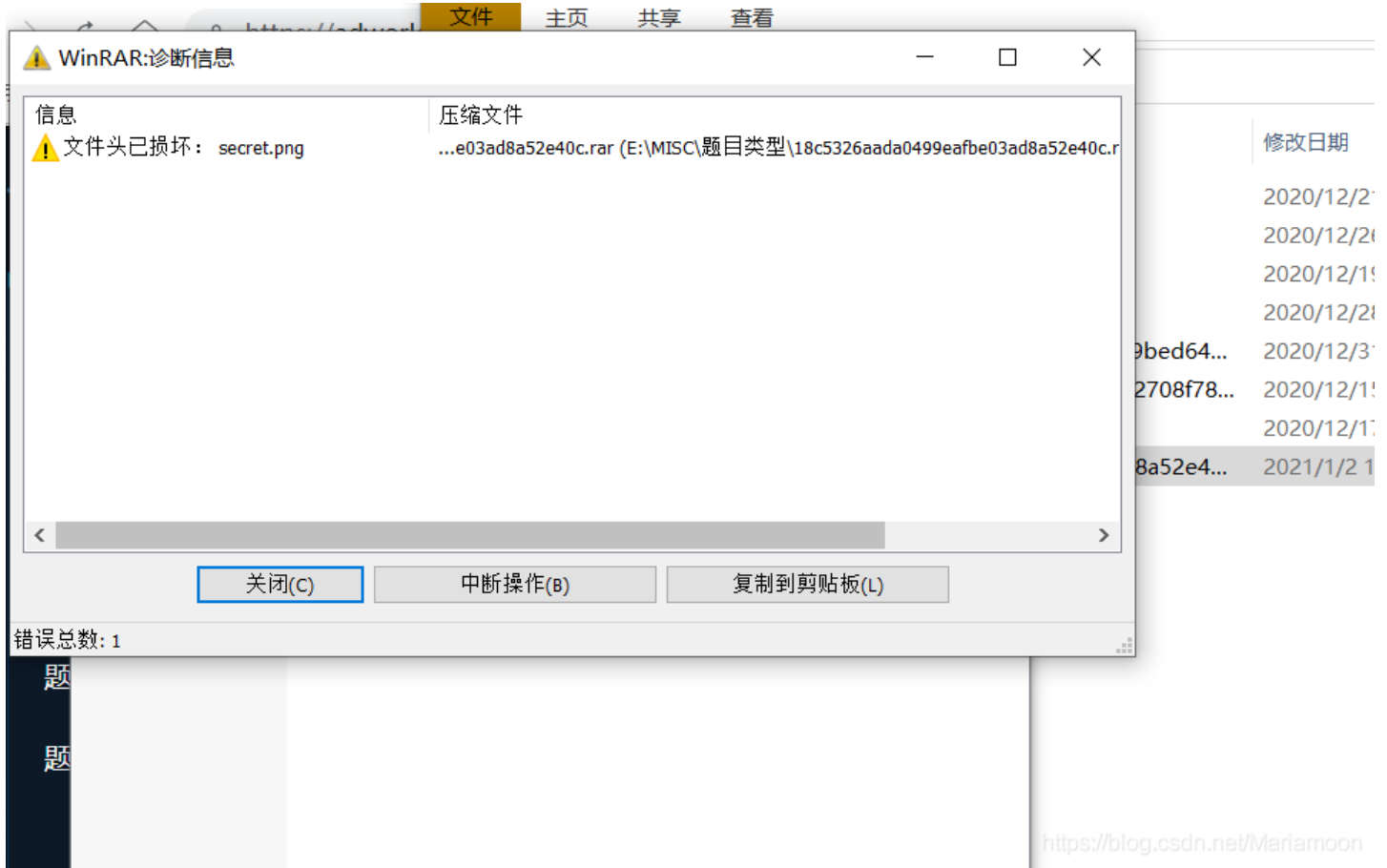
题目描述: 菜狗最近学会了拼图, 这是他刚拼好的, 可是却搞错了一块(ps:双图层)

题目场景: 暂无

题目附件: 附件1

<https://blog.csdn.net/Mariamoon>

打开下载文件提示RAR文件文件头损坏



打开winhex查看, 发现65是flag.txt的文件结尾, A8 3C代表了校验码, 所以7A代表了块类型, 这里我们的块类型是文件头, 所以

```

00000032 | 00 00 00 02 C7 88 67 36 6
00000048 | 20 00 00 00 66 6C 61 67 2
00000064 | 43 66 6C 61 67 20 69 73 2
00000080 | 65 A8 3C 7A 20 90 2F 00 3
00000096 | 02 BC E9 8C 2F 6E 84 4F 4
00000112 | 00 73 65 63 72 65 74 2E 7

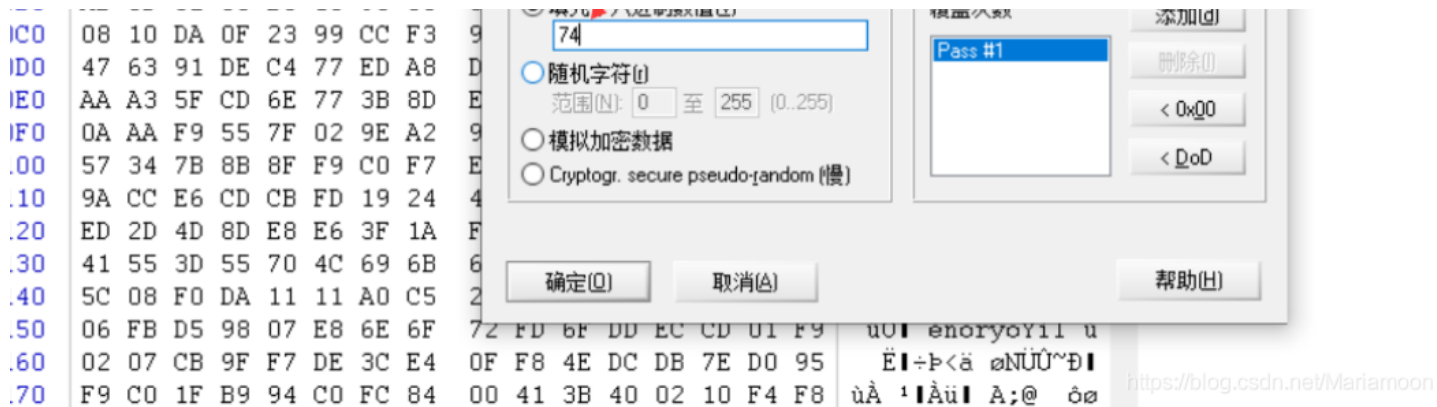
```

改成74就可以正常解压文件  
进行填充:

130	20 00 00 00 66 6C 61 67	2E 74 78 74 00 B0 57 00	flag.txt %W
140	43 66 6C 61 67 20 69 73	20 6E 6F 74 20 68 65 72	Cflag is not her
150	65 A8 3C 7B 20 90 2F 00	3A 15 00 00 42 16 00 00	e" <{ / : B
160	02 BC E9 8C 2F 6E 84 4F	4B 1D 33 0A 00 20 00 00	461/n1OK 3
170	00 73 65 63 72 65 74 2E	70 6E 67 00 F0 40 AB 18	secret.png δ@<
180	11 C1 11 55 08 D1 55 80	0D 99 C4 90 87 93 22 19	Á U ÑU IÄ   "
190	4C 58 DA 18 B1 A4 58 16		
1A0	04 05 85 96 21 AB 1A 43		
1B0	AB 3D 02 80 B0 10 90 C5		

填充选块

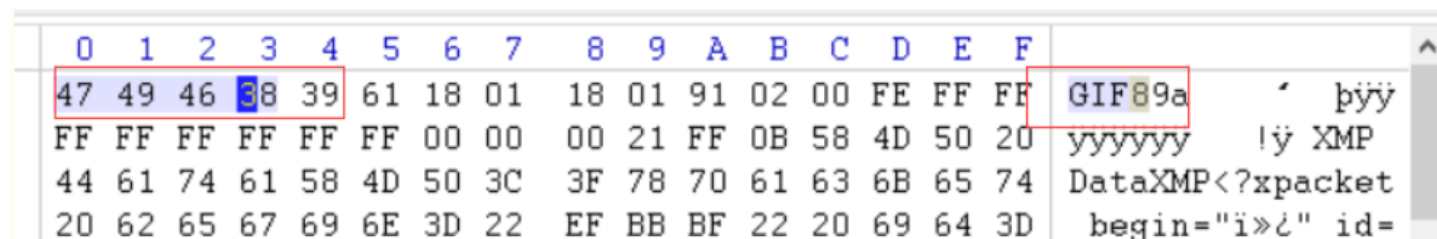
值在十六进制数值(v)  覆盖数据



保存后解压，再把sercet.png丢到winhex里发现文件头为gif图，将图片后缀名改为.gif。

jpg图像开始标志：FF D8 结束标志：FF D9

gif图像开始标志：47 49 46 38 39 61 结束标志：01 01 00 3B



修改后缀名为.gif，利用Stegsolve中的frame browser分解成两个图片然后用底下的左右箭头选择图层分离出二维码得到两张不全的二维码



由题可知为双层图，用ps打开分离图层后保存，最后ps全二维码，扫描得到flag: flag{yanji4n\_bu\_we1shi}

### 题目三：zip的伪加密

1.压缩源文件数据区:

50 4B 03 04: 这是头文件标记 (0x04034b50)  
14 03: 解压文件所需 pkware 版本  
00 00: 全局方式位标记 (判断有无加密的重要标志)  
08 00: 压缩方式  
68 BF: 最后修改文件时间  
9B 48: 最后修改文件日期  
FE 32 7D 4B: CRC-32校验  
E9 0D 00 00: 压缩后尺寸  
B5 1B 00 00: 未压缩尺寸  
09 00: 文件名长度  
00 00: 扩展记录长度

2.压缩源文件目录区:

50 4B 01 02: 目录中文件文件头标记(0x02014b50)  
3F 03: 压缩使用的 pkware 版本  
14 03: 解压文件所需 pkware 版本  
00 00: 全局方式位标记 (有无加密的重要标志, 这个更改这里进行伪加密, 改为09 00打开就会提示有密码了)  
08 00: 压缩方式  
68 BF: 最后修改文件时间  
9B 48: 最后修改文件日期  
FE 32 7D 4B: CRC-32校验 (1480B516)  
E9 0D 00 00: 压缩后尺寸 (25)  
B5 1B 00 00: 未压缩尺寸 (23)  
09 00: 文件名长度  
24 00: 扩展字段长度  
00 00: 文件注释长度  
00 00: 磁盘开始号  
00 00: 内部文件属性  
20 80 ED 81: 外部文件属性  
00 00 00 00: 局部头部偏移量

压缩源文件目录结束标志:

50 4B 05 06: 目录结束标记  
00 00: 当前磁盘编号  
00 00: 目录区开始磁盘编号  
01 00: 本磁盘上纪录总数  
01 00: 目录区中纪录总数  
5B 00 00 00: 目录区尺寸大小  
10 0E 00 00: 目录区对第一张磁盘的偏移量  
00 00: ZIP 文件注释长度

然后就是识别真假加密

### 1.无加密

压缩源文件数据区的全局加密应当为00 00

且压缩源文件目录区的全局方式位标记应当为00 00

### 2.假加密

压缩源文件数据区的全局加密应当为00 00

且压缩源文件目录区的全局方式位标记应当为09 00

### 3.真加密

压缩源文件数据区的全局加密应当为09 00

且压缩源文件目录区的全局方式位标记应当为09 00

writeup:

1.这题全局为00 00 但是在结尾发现是09 00，所以为假加密，把09 00 改成00 00就能解压打开文件了。

### 2.通过Zipcenop.jar工具对其进行解密

```
java -jar ZipCenOp.jar r xxx.zip
```

接

3.着我们打开txt文件，可以看到很多base64密码

```
U3RIZ2Fub2dyYXBoeSBpcyB0aGUgYXJ0IGFuZCBzY2llbmNlIG9m
IHdyaXRpbmcaGikZGVuIGl3NhZ2VzIGlulHN1Y2ggYSB3YXkgdGhhdCBubyBvbmV=
LCBhcGFydCBmcm9tHRoZSBzZW5kZXIyY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
Y3RzIHRoZSBleGlzdGVuY2Ugb2YgdGhllIGl3M=
YWdlLCBhIGZvcu0gb2Ygc2VjdXJpdHkgdGhyb3VnaCBvYnNjdXJpdHkulFS=
aGUgd29yZCBzdGVnYW5vZ3JhcGh5IGl3IG9mIEdyZWVrIG9yaWdpbiBhbmQgbWVhbnMgImNvbmluY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
bGVkIHdyaXRpbmciIGZyb20gdGhllIEdyZWVrIHdvcuRzIHN0ZWdhbm9zIGl1Y2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
dmVzZWQgb3IgcHJvdGVjdGVkIiwgYW5kIGdyYXBoZWludGVuZGVkIHJlY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
cmI0ZSllFRoZSBmaXJzdCBYb2VzIGludGVuZGVkIHJlY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
YW5uZXMgVHJpdGhblWl1cyBpbmQgc3VzcGU=
dGlzZSBvbiBjcnlwdG9ncmFwaHkgYW5kIHJlY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
dWlzZWQgYXMGYSBib29rIG9uIGl1hZ2lLiBHZW5lcmFsbHksIGl3P=
YWdlcyB3aWxsIGFwcGVhcnR0byBiZSBz21ldGhpbmcaGikZGVuIGl3NhZ2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
Y2xlcwgc2hvcHBpbmcaGikZGVuIGl3NhZ2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
aGVyIGNvdmludGVuZGVkIHJlY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
c2libGUgaW5rIGludGVuZGVkIHJlY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
IGFkdmludGVuZGVkIHJlY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
eXB0b2dyYXBoeSBhbG9uZSwaXMGdGhhdCBtZXNzYWdlcyBkbyBub3QgYXR0cmFjdCBhdHRlbnRpb25=
IHRvIHRoZW1zZWx2ZXMuIFBsYWlubHkgdmlzaWJsZSBibmNyeXB0ZWQgbWVzc2FnZXOXbm8gbWF0dGVyIF==
aG93IHVudYnJlY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
dXNwaWNpb24sIGFuc2VzIGludGVuZGVkIHJlY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
aW4gY291bnRyaWVzIHdoZXJlIGludGVuZGVkIHJlY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
IHdoZXJlY2VzIGludGVuZGVkIHJlY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
IGEGbWVzc2FnZSwaXMGdGhhdCBtZXNzYWdlcyBkbyBub3QgYXR0cmFjdCBhdHRlbnRpb25=
b3RoIGl3NhZ2VzIGludGVuZGVkIHJlY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
ZGVzIHRoZSBjb25jZWVudCwgc3Vzc2VzIGludGVuZGVkIHJlY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
cHV0ZXIyY2VzIGludGVuZGVkIHJlY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
cyBtYXkgYW5jbHVkZSBzdGVuY2VzIGludGVuZGVkIHJlY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
ZGUgb2YgYSB0cmFuc3Vvcu0gb2Ygc2VjdXJpdHkgdGhyb3VnaCBvYnNjdXJpdHkulFS=
ZSwgcHJlY2VzIGludGVuZGVkIHJlY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
ZmlsZXMGYXJ0IGFuZCBzY2llbmNlIG9mIHdyaXRpbmcaGikZGVuIGl3NhZ2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
biBiZWNhdXNlIG9mIHdyaXRpbmcaGikZGVuIGl3NhZ2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
YSBzaW1wbGUgZXhhbXB0eSBzZW5kZXIyY2VzIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
```

biBpbm5vY3VvdXMgaW1hZ2UgZmlsZSBhbmQgYWRqdXN0IHRoZSBjb2xvciBvZiBldmVyeSAxMDB0aCBwaXhlcD=  
dG8gY29ycmVzcG9uZCB0byBhIGxldHRlciBpbiB0aGUgYWxwaGFiZXQsIGF=  
IGNoYW5nZSBzbyBzdWJ0bGUgdGhhdCBzb21lb25lIG5vdCBzcGVjaWZpY2FsbHkgbG9va2luZyBm  
b3lgaXQgaXMgdW5saWtlbHkgdG8gbm90aWNlIGl0Lg0KDQpUaGU=  
IGZpcnN0IHJiY29yZGVkIHVzZXMgb2Ygc3RIZ2Fub2dyYXBoeSBjYW4gYmUgdHJ=  
YWNIZCBiYWNrIHRvDQ0MCBCQyB3aGVuEhlc9kb3R1cyBtZW50aW9ucyB0d28gZXhhdXBsZXMGb+==  
ZiBzdGVnYW5vZ3JhcGh5IGlulFRoZSBlaXN0b3JpZXMGb2Yg  
SGVyb2RvdHVzLiBEZW1hcmF0dXMgc2VudCBhIHdhcm5pbmcmgYWJvdXQgYSB=  
Zm9ydGhjb21pbmcmgYXR0YWNrIHRvIEdyZWWjZSBieSB3  
cmI0aW5nIGl0IGRpcmVjdGx5IG9uIHRoZSB3b29kZW4gYmFja2luZyBvZiBhIHdhcCB0YWJsZXQgYmVm  
b3JlIGFwcGx5aW5nIGl0cyBiZWVzd2F4IHN1cmZhY2UuIFdheCB0YWJsZXRzIHdlcmUgaW4gY29tbW9uIHVzZV==  
IHRoZW4gYXMgcmlvV1c2FibGUgd3JpdGluZyBzdXJmYWNlcywgc29tZXRpbWX=  
cyB1c2VklGZvciBzaG9ydGhbmQuIEFub3R0ZXlgYW5jaWVudCBleGFtcGxlIGlziHRoYXQgb9==  
ZiBlaXN0aWFlidXMsIHdobyBzaGF2ZWQgdGhllGhYlWQgb2YgaGlzIG1vc3QgdHJ1c3RIZCBz  
bGF2ZSBhbmQgdGF0dG9vZWQgYSBtZXNzYWdlIG9uIGl0LiBBZnRlciBoaXMgaGFpciBoYWQgZ5==  
cm93biB0aGUgbWVzc2FnZSB3YXMgaGlkZGVuLiBUaGUgcHVycG9zZSB3YXMgdG+=  
IGluc3RpZ2F0ZSBhIHJldm9sdCBhZ2FpbN0IHRoZSBQZXJzaWFucy4NCg0KU3RIZ2Fub2dyYXBoeSB0YXMGYm==  
ZWWuIHdpZGVseSB1c2VklCBpbmNsdWRpbmcmgaW4gcmVjZW50IGhpc3RvcmljYWwgdGltZXMGYW5kIHt=  
aGUgcHJlc2VudCBkYXkxulFbc3NpYmXlHBlcm11dGF0aW9ucyBhcmUgZW5kbGVzcyBhbmT=  
IGtub3dulGV4YW1wbGVzIGluY2x1ZGU6DQoqIEhpZGRlbiBtZXNzYWdlcyB3aXRoaW4gd2F4IHRh  
YmXldHM6IGlulGFuY2llbnQgR3JlZWNILCBwZW9wbGUgd3JvdGUgbWV=  
c3NhZ2VzIG9uIHRoZSB3b29kLCB0aGVuIGNvdmVyZWQgaXQgd2l0aCB3YXggdXBvbiB3aGljaCBhbiBpbm5vY2Vu  
dCBjb3ZlcmIuZyBtZXNzYWdlIHdhcyB3cmI0dGVu  
Lg0KKiBlaWRkZW4gbWVzc2FnZXMGb24gbWVzc2VuZ2VvYj3MgYm9keTogYWxzbyB1c2VklGluGFuY2llbt==  
dCBHcmVlY2UuEhlc9kb3R1cyB0ZWxscyB0aGUgc3Rvcnkgb1==  
ZiBhIG1lc3NhZ2UgdGF0dG9vZWQgb24gYSBzbGF2ZSdzlHNoYXZlZCB0ZWFlkCB0aWRkZW4gYnkgdGhl  
IGdyb3d0aCBvZiBoaXMgaGFpciwgYW5kIGV4cG9zZWQgYnkgc2hhdmluZyBoaXMgaGVhZM==  
IGFnYWluLiBUaGUgbWVzc2FnZSBhbGxlZ2VkbHkgY2FycmlZCBhIHdhcm5pbmcmgdG8gR3JlZWNlIGFib5==  
dXQgUGVyc2lhb1BpbmZhc2lubiBwbGFucy4gVGh=  
aXMgbWV0aG9kIGhcyBvYnZpb3VzIGRyYXdiYWNrcyz=  
IHN1Y2ggYXMgZGVsYXllZCB0cmFuc21pc3Npb24gd2hpbGUgd2FpdGluZyBmb3lkdGhllHP=  
bGF2ZSdzlGhhaXlkdG8gZ3JvdywgYW5kIHRoZSBYXN0cmIjdlvbnMgb3==  
biB0aGUgbnVtYmVylGFuZCBzaXplIG9mIG1lc3M=  
YWdlcyB0aGF0IGNhbiBiZSBibmNvZGVklG9uIG9uZSBwZXJzb24=  
J3Mgc2NhbHAuDQoqIElulFdXSUksIHRoZSBGcmVvY2ggUmVzaXN0YW5jZSBzZW50IHNVbWUgbWVzc2FnZXMGd2==  
cmI0dGVuIG9uIHRoZSBiYWNrcyBvZiBjb3VyaWVycyD=  
dXNpbmcmgaW52aXNpYmXlIGuay4NCiogSGlkZGVuIG1lc3NhZ2VzIG9uIHhlcGVyIHdy  
aXR0ZW4gaW4gc2VjcmV0IGlua3MsiHVuZGVyIG90aGVyIG1lc3NhZ2Vz  
IG9yIG9uIHRoZSBibGFuayBwYXJ0cyBvZiBvdGhlcT==  
IG1lc3NhZ2VzLg0KKiBNZXNzYWdlcyB3cmI0dGVuIGlulE1vcnNlIGNvZGUgd24ga25pdHRpbmcmgWfYbiBhbmQg  
dGhliBibmI0dGVklGluZG8gYSBwaWVjZSBvZiBjbG90aGUyB3b3K=  
biBieSBhIGNvdXJpZXluDQoqIE1lc3NhZ2VzIHdyaXR0ZW4gb24gdGhllGhY2sgb5==  
ZiBwb3N0YWdlIHNoY29yZy4NCiogRHVyaW5nIGFuZCBhZnRlcm==  
IFdvcmxkIFdheciBJSswgZXNwaW9uYWdlIGFnZW50cyB1c2VklHBob3RvZ3JhcGhpY2FsbHkgcO==  
cm9kdWNIZCBtaWNyb2RvdHMgdG8gc2VuZCBpbmZvcmlhdGVmBiYWNrIGFuZH==  
IGZvcnRoLiBNaWNyb2RvdHMgd2VyZSB0eXBpY2FsbHkg  
bWludXRILCBhcyB3pbpWF0ZWx5IGxl3MgdGhbiB0aGUgc2l6ZSBvZiB0aGUgcGVyaW9kIHByb2R=  
dWNIZCBieSBhIHR5cGV3cmI0ZXluIFdXSUkgbWljb3RzIG5lZWRIZCB0byBiZSBibWJlZGRIZB==  
IGlulHRoZSBwYXBciBhbmQgY292ZXJIZCB3aXR0IGFuGFkaGVzaXZlIChzdWN0IGFzIGNvbGxvZGVibiklFR=  
aGlzIHdhcyByZWZsZWNoaXZlIGFuZCB0aHVzIGRldGVjdGFibGUg

ynkgamiiaZiuzyBnZZFpbnnuIGdsYVv5Jaw5niGxpZZnULIBBdHKicm5ndGiZZSBUZWNoBmixdVvZIGUIYZX1ZGVK  
IGluc2VydGluZyBtaWNYb2RvdHMgaW50byBzbGl0cyBjdXQgaW50byB0aGUgZWRnZSBvZv==  
IHBvc3QgY2FyZHMuDQoqIER1cmIuZyBxb3JsZCBXYXl9SUksIGEgc3B5IGZvciB=  
SmFwYW4gaW4gTmV3IFlvcmsgQ2l0eSwgVmVsdmFsZWw=  
IERpY2tpbnNvbWwgc2VudCBpbmZvcmlhdGlvbiB0byBhY2NvbW1vZGF0aW9=  
biBhZGRyZXNzZXMGaW4gbmV1dHJhbCBTb3V0aCBBbWVyaW0=  
YS4gU2hlIHdhcyBhIGRIYWxciBpbiBkb2xscywgYW5kIG==  
aGVyIGxldHRlcuMgZGlzY3Vzc2VklGhvdYBtYW55IG9mIHRoaXMgb3lmdGhhdCBkb2xs  
IHRvIHNoaXAuIFRoZSBzdGVnb3RleHQgd2FzIHRoZSBkb2xsIG9yZGVyYcywgd2hpbGUgdGhl  
IGNvbmNlYWxkaW50ZXh0IiB3YXMGaXRzZWxmIGVuy2+=  
ZGVuIGFuZCBnYXZlIGluZm9ybWFOaW9uIGFib3V0IHNoaXAgaW92ZW1lbnRzLF==  
IGV0Yy4gSGVylGNhc2UgYmVjYW1lIHVvWV3aGF0IGZh  
bW91cyBhbmQgc2hlIGJY2FtZSBrbm93biBhcyB0aGX=  
IERvbGwvV29tYW4uDQoqIENvbGQgV2FyIGVudW50  
ZXltcHJvcGFuYw5kYS4gSW4gMTk2OCwgY3JldyBtZW1iZW==  
cnMgb2YgdGhlIFVtUyBQdWVibG8gKEFHRVltMikgaW50ZWxsaWdlbmlNIHNoaXAgaGVsZCBhcyBwcm==  
aXNvbMvYcyBieSBOb3J0aCBLb3JlYSwgY29tbXVuaWNhdGVklIGluIHnpZ25=  
IGxhbmd1YWdlIGR1cmIuZyBzdGFuZWQgcGhvdG8gb3Bwb3J0  
dW5pdGllcywgaW5mb3JtaW5nIHRoZSBVbml0ZWQgU3RhdGVzIHRoZXkg  
d2VyZSBub3QgZGVmZWNo3JzIGJ1dCBYyXRoZlZlZGVyZSBiZWluZyBoZWxkIGNh  
cHRpdmlUgYnkGdGhlIE5vcnR0IEtvcmlhbnMlElulG90aGVyIHBob3Rv  
cyBwcmVzZW50ZWQgdG8gdGhlIFVtLCBjcmV3IG1lbWJlcnMgZ2F2ZSAidGhllGZpbmdlciGdG8g  
dGhlIHVuc3VzcGVjdGluZyBoB3J0aCBLb3JlYw5zLCBpb3BiBhbiBhdHRlbiXB0IHRvIE==  
ZGlzY3JIZG0IHBob3RvcyB0aGF0IHNoB3dlZCB0aGVtIHNTaQ==  
bGluZyBhbmQgY29tZm9ydGFibGUuDQoNCi0tDQpodHRwOi8vZW4ud2lraXBIZGlhLm9yZw==  
L3dpa2kvU3RIZ2Fub2dyYXBoeQ0K

4.发现是bas64隐写，通过脚本解密：

```
#!/user/bin/env python
def get_base64_diff_value(s1, s2):
base64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
res = 0
for i in xrange(len(s1)):
if s1[i] != s2[i]:
return abs(base64chars.index(s1[i]) - base64chars.index(s2[i]))
return res

def solve_stego():
```



```
with open('1.txt', 'rb') as f:
    file_lines = f.readlines()

bin_str = ''
for line in file_lines:
    steg_line = line.replace('\n', '')
    norm_line = line.replace('\n', '').decode('base64').encode('base64').replace('\n', '')
    diff = get_base64_diff_value(steg_line, norm_line)
    pads_num = steg_line.count('=')
    if diff:
        bin_str += bin(diff)[2:].zfill(pads_num * 2)
    else:
        bin_str += '0' * pads_num * 2

res_str = ''

for i in xrange(0, len(bin_str), 8):
    res_str += chr(int(bin_str[i:i+8], 2))
print res_str
```

solve\_stego()

最终flag为:

flag{Base\_sixty\_four\_point\_five}