ctfshow月饼杯misc1_共婵娟

原创

是Mumuzi 于 2021-01-28 18:14:26 发布 1454 收藏 5

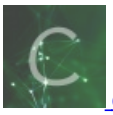
分类专栏： ctf ctfshow 文章标签： 网络安全

本文链接：https://blog.csdn.net/qq_42880719/article/details/113344037


ctf 同时被 2 个专栏收录

75 篇文章 28 订阅

订阅专栏


ctfshow

23 篇文章 8 订阅

订阅专栏

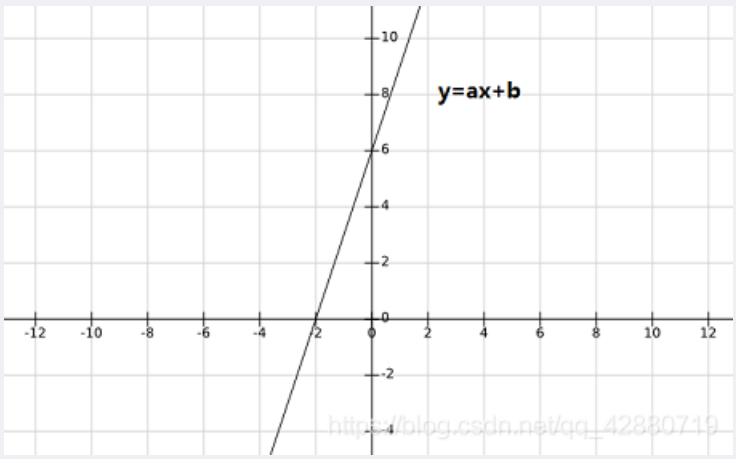
不得不说，la师傅出的misc题tql

对我这种才学一两个月的新人简直不讲武德

层层套，层层解密

感谢DonmeN一起前进和提供USB流量思路

misc1_共婵娟
但愿人长久，千里共婵娟
MISC之王争夺战！
两密一一对应，填入神秘代码
@□Lazzaro□师傅供题
hint1:神秘代码中有东西缺失，填补后开阔思维， / 仅是分隔线
hint2:https://pan.baidu.com/

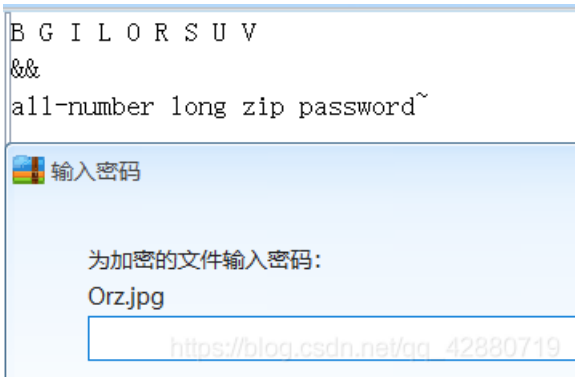


hint3:

有了hint2其实就用不了hint1了，等会再讲。

第一层:zip密码猜测

下载得到附件，发现密码加密



winhex查看发现没有伪加密，所以想办法得到zip密码

all-number和long。emmm

思考了一会儿，尝试将字母对应的数字填进去（A=1,B=2……）

279121518192122

解压得到Orz.jpg，有趣的地方才刚刚开始。

第二层：寻找流量包解压密码

拿到图片后，尝试foremost分解一下，分解出了zip包，打开一看需要密码，那么图片肯定藏的有密码

查看jpg的文件尾，就是刚刚那个zip，然后我看了看属性，发现高和宽

分辨率	225 x 204
宽度	225 像素
高度	204 像素

有问题，按道理来说，这种图应该是方方正正的吧，改一下高度试试

204的十六进制是CC，所以在winhex搜索00CC，随便改一个比如01CC

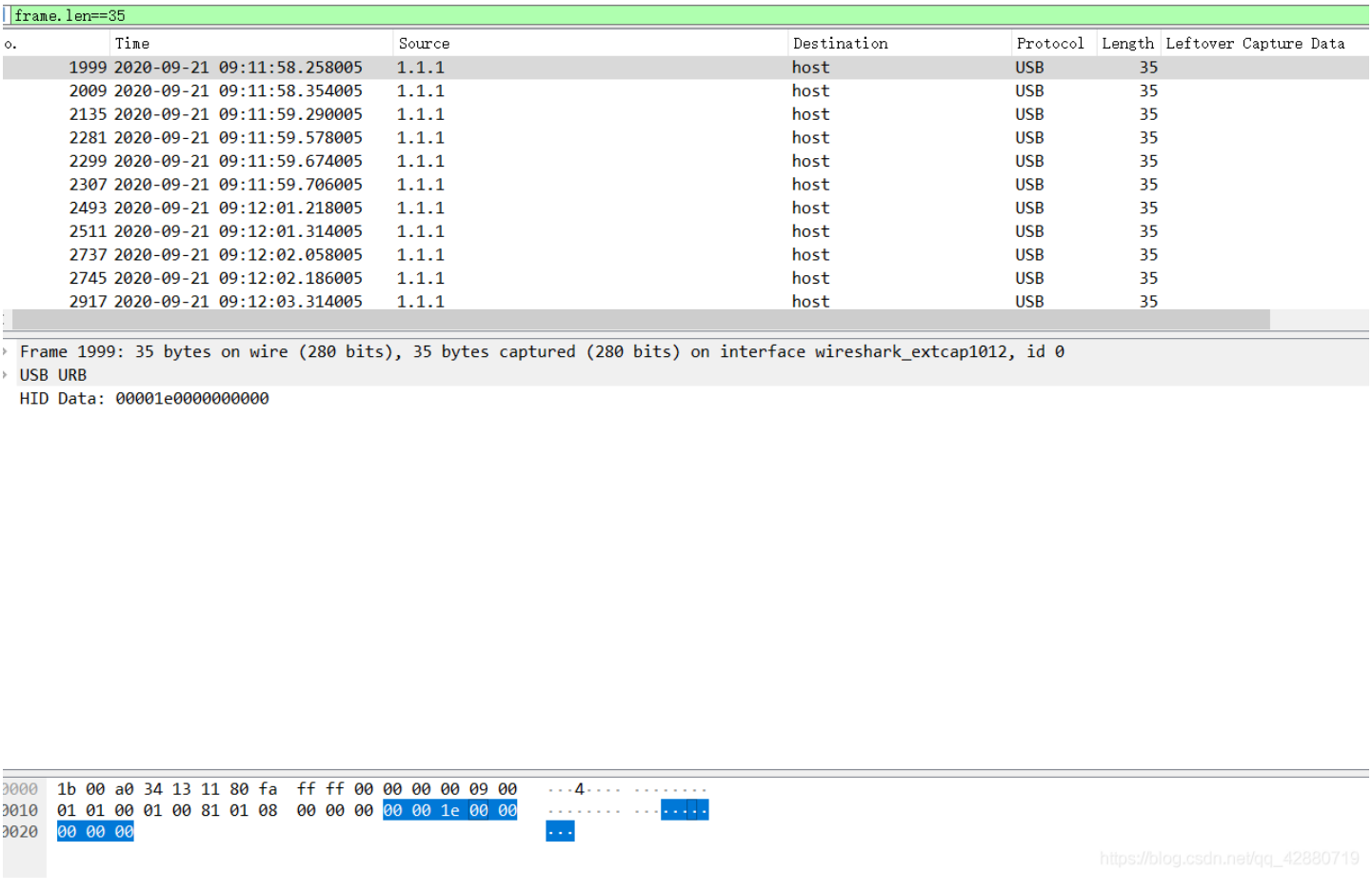
Offset	搜索结果	时间
A3	00CC	2021/01/28 1...
319F	00CC	2021/01/28 1...
32C5	00CC	2021/01/28 1...
13B8F	00CC	2021/01/28 1...
1C544	00CC	2021/01/28 1...
497F7	00CC	2021/01/28 1...
4F1A9	00CC	2021/01/28 1...
5D264	00CC	2021/01/28 1...
67931	00CC	2021/01/28 1...
6B89C	00CC	2021/01/28 1...
87A4E	00CC	2021/01/28 1...
8F62A	00CC	2021/01/28 1...
A2858	00CC	2021/01/28 1...
A8C94	00CC	2021/01/28 1...
AB549	00CC	2021/01/28 1...
AF3F8	00CC	2021/01/28 1...
B749E	00CC	2021/01/28 1...

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000000	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	01	00	60	ÿØÿà JFIF `
00000010	00	60	00	00	FF	DB	00	43	00	02	01	01	02	01	01	02	` ÿÛ C
00000020	02	02	02	02	02	02	02	03	05	03	03	03	03	03	06	04	
00000030	04	03	05	07	06	07	07	07	06	07	07	08	09	0B	09	08	
00000040	08	0A	08	07	07	0A	0D	0A	0A	0B	0C	0C	0C	0C	07	09	
00000050	0E	0F	0D	0C	0E	0B	0C	0C	0C	FF	DB	00	43	01	02	02	ÿÛ C
00000060	02	03	03	03	06	03	03	06	0C	08	07	08	0C	0C	0C	0C	
00000070	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	
00000080	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	
00000090	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	FF	C0	ÿÀ
000000A0	00	11	0	00	CC	0	E1	03	01	22	00	02	11	01	03	11	Ì á "
000000B0	01	FF	C4	00	1F	00	00	01	05	01	01	01	01	01	01	00	ÿÄ

改了之后再查看图片

得到解压密码JZHUHDGVC
得到流量包：mysterious.pcapng。

第三层：USB流量包分析
打开附件，得到流量包，发现全是USB协议
USB协议怎么那么多，那么杂呢？
闻了一下la师傅，la师傅说那些可能是在出题的时候混进去的
好家伙，卡了那么久的流量包，就应该按照正常思路去做。
这里参照的是http://www.ga1axy.top/index.php/archives/22/
而且发现中间有很多长度34，35，36之类的
分别查看一下



最后发现当长度为35的时候满足键盘流量（8字节），kali下用指令提取出来并分隔成2byte形式

> kali下提取len=35指令：tshark -r mysterious.pcapng -T fields -e usb.capdata -Y frame.len==35 >out.txt

得到

```
00:00:1e:00:00:00:00:00
00:00:00:00:00:00:00:00
20:00:00:00:00:00:00:00
20:00:0d:00:00:00:00:00
20:00:00:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:26:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:07:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:0b:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:09:00:00:00:00:00
00:00:00:00:00:00:00:00
20:00:00:00:00:00:00:00
20:00:1d:00:00:00:00:00
20:00:00:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:0f:00:00:00:00:00
00:00:00:00:00:00:00:00
20:00:00:00:00:00:00:00
20:00:0b:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:06:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:0a:00:00:00:00:00
00:00:00:00:00:00:00:00
20:00:00:00:00:00:00:00
20:00:18:00:00:00:00:00
20:00:00:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:1e:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:06:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:16:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:2a:00:00:00:00:00
00:00:00:00:00:00:00:00
20:00:00:00:00:00:00:00
20:00:0b:00:00:00:00:00
20:00:00:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:07:00:00:00:00:00
```

这里讲一下 前面的20和00
下方图片来源：记一次CTF的USB流量分析

```
BYTE1 --
        |--bit0:   Left Control是否按下，按下为1
        |--bit1:   Left Shift  是否按下，按下为1
        |--bit2:   Left Alt    是否按下，按下为1
        |--bit3:   Left GUI    是否按下，按下为1
        |--bit4:   Right Control是否按下，按下为1
        |--bit5:   Right Shift 是否按下，按下为1
```

```
        |--bit6:   Right Alt   是否按下，按下为1
        |--bit7:   Right GUI   是否按下，按下为1
BYTE2 -- 暂不清楚，有的地方说是保留位
BYTE3--BYTE8 -- 这六个为普通按键
```


20转换成二进制是00010100
对应bit3和bit5
而按下shift时，打出来的是大写
（02也是，转二进制是00000010，bit1）

```
normalKeys = {
    "04":"a", "05":"b", "06":"c", "07":"d", "08":"e",
    "09":"f", "0a":"g", "0b":"h", "0c":"i", "0d":"j",
     "0e":"k", "0f":"l", "10":"m", "11":"n", "12":"o",
      "13":"p", "14":"q", "15":"r", "16":"s", "17":"t",
       "18":"u", "19":"v", "1a":"w", "1b":"x", "1c":"y",
        "1d":"z","1e":"1", "1f":"2", "20":"3", "21":"4",
         "22":"5", "23":"6","24":"7","25":"8","26":"9",
         "27":"0","28":"<RET>","29":"<ESC>","2a":"<DEL>", "2b":"\t",
         "2c":"<SPACE>","2d":"-","2e":"=","2f":"[","30":"]","31":"\\",
         "32":"<NON>","33":";","34":"'","35":"<GA>","36":",","37":".",
         "38":"/","39":"<CAP>","3a":"<F1>","3b":"<F2>", "3c":"<F3>","3d
         "3e":"<F5>","3f":"<F6>","40":"<F7>","41":"<F8>","42":"<F9>","4
         "44":"<F11>","45":"<F12>"}
shiftKeys = {
    "04":"A", "05":"B", "06":"C", "07":"D", "08":"E",
     "09":"F", "0a":"G", "0b":"H", "0c":"I", "0d":"J",
      "0e":"K", "0f":"L", "10":"M", "11":"N", "12":"O",
       "13":"P", "14":"Q", "15":"R", "16":"S", "17":"T",
        "18":"U", "19":"V", "1a":"W", "1b":"X", "1c":"Y",
         "1d":"Z","1e":"!", "1f":"@", "20":"#", "21":"$",
          "22":"%", "23":"^","24":"&","25":"*","26":"(","27":")",
          "28":"<RET>","29":"<ESC>","2a":"<DEL>", "2b":"\t","2c":"<SPAC
          "2d":"_","2e":"+","2f":"{","30":"}","31":"|","32":"<NON>","33
          "34":":","35":"<GA>","36":"<","37":">","38":"?","39":"<CAP>",
          "3b":"<F2>", "3c":"<F3>","3d":"<F4>","3e":"<F5>","3f":"<F6>",
          "41":"<F8>","42":"<F9>","43":"<F10>","44":"<F11>","45":"<F12>
```


所以用脚本将键盘信息提取出来是

> 1J9dhfZlHcgU1cHdnDG4VCw/dmdm(已经删除了<del>)

根据hint2和百度网盘分享长度可知，前面正是分享的链接，而后面是提取的密码，访问即可得到下一步。

第四层：零宽字符隐写

下载百度盘的附件，可以得到中秋节的来历、由来、传说和习俗.txt

用winhex一看就发现了零宽字符。kali下查看

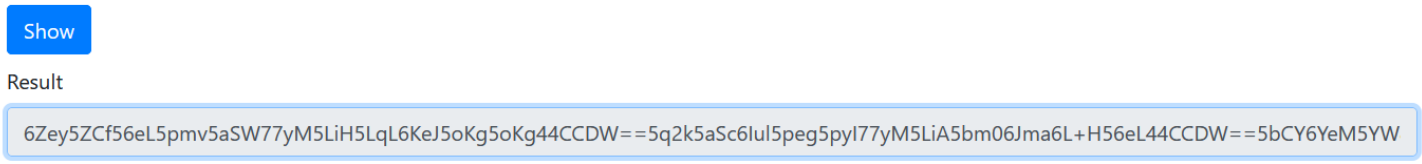
```
00d><200f><200c><200b><200b><200b><200b><200d><200f><200f><200b><200b><200b><200
b><200d><200b><200d><200b><200b><200b><200b><200d><200b><200d><200b><200b><200b>
<200b><200d><200e><200d><200b><200b><200b><200b><200d><200e><200d><200b><200b><2
00b><200b><200d><200e><200e><200b><200b><200b><200b><200e><200d><200e><200b><200
b><200b><200b><200d><200d><200c><200b><200b><200b><200b><200d><200d><200c><200b>
<200b><200b><200b><200d><200b><200e><200b><200b><200b><200b><200f><200d><200f><2
00b><200b><200b><200b><200f><200e><200d><200b><200b><200b><200b><200f><200b><200
f><200b><200b><200b><200b><200d><200b><200e><200b><200b><200b><200b><200f><200d>
<200d><200b><200b><200b><200b><200f><200f><200c><200b><200b><200b><200b><200d><2
00f><200e><200b><200b><200b><200b><200d><200b><200f><200b><200b><200b><200b><200
e><200f><200d><200b><200b><200b><200b><200e><200b><200f><200b><200b><200b><200b>
<200f><200b><200c><200b><200b><200b><200b><200d><200b><200e><200b><200b><200b><2
00b><200f><200d><200d><200b><200b><200b><200b><200f><200c><200b><200b><200b><200
b><200b><200e><200b><200f><200b><200b><200b><200b><200d><200b><200f><200b><200b>
<200b><200b><200e><200e><200b><200b><200b><200b><200b><200e><200d><200d><200b><2
00b><200b><200b><200e><200f><200f><200b><200b><200b><200b><200d><200c><200b><200
b><200b><200b><200b><200d><200c><200b><200b><200b><200b><200b><200f><200f><200c>
<200b><200b><200b><200b><200e><200b><200d><200b><200b><200b><200b><200d><200b><2
00e><200b><200b><200b><200b><200e><200f><200e><200b><200b><200b><200b><200d><200
b><200b><200b><200b><200b><200b><200e><200c><200e><200b><200b><200b><200b><200d>
<200b><200e><200b><200b><200b><200b><200e><200f><200e><200b><200b><200b><200b><2
00c><200e><200e><200b><200b><200b><200b><200d><200e><200e><200b><200b><200b><200
b><200d><200b><200e><200b><200b><200b><200b><200f><200d><200c><200b><200b><200b>
```

U+200b U+200c U+200d U+200f

见f识人

全部复制下来，解密https://offdev.net/demos/zwsp-steg-js

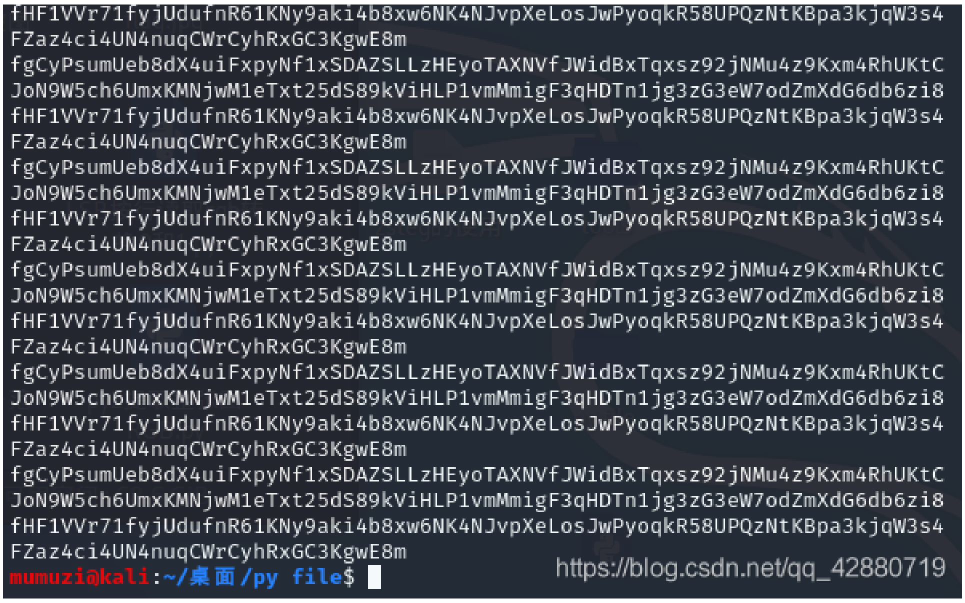
得到一长串含有等号的大小写跟数字



第五层：base64隐写

为了确定这确实是base64，拿去base64解密第一行

确实是，那么我将一串base64放进txt，使用替换功能正则匹配将==替换成==\n
这样一来，方便进行解密，使用脚本解密得到



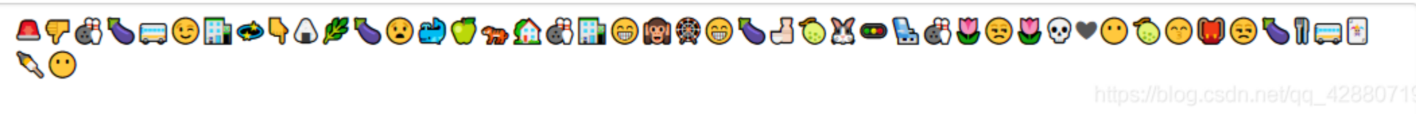
```
fHF1VVr71fyjUdufnR61KNy9aki4b8xw6NK4NJvpXeLosJwPyoqkR58UPQzNtKBpa3kjqW3s4
FZaz4ci4UN4nuqCWrCyhRxGC3KgwE8m
fgCyPsumUeb8dX4uiFxpyNf1xSDAZSLLzHEyoTAXNVfJWidBxTqxsz92jNMu4z9Kxm4RhUKtC
JoN9W5ch6UmxKMNjwM1eTxt25dS89kViHLP1vmMmigF3qHDTn1jg3zG3eW7odZmXdG6db6zi8
fHF1VVr71fyjUdufnR61KNy9aki4b8xw6NK4NJvpXeLosJwPyoqkR58UPQzNtKBpa3kjqW3s4
FZaz4ci4UN4nuqCWrCyhRxGC3KgwE8m
fgCyPsumUeb8dX4uiFxpyNf1xSDAZSLLzHEyoTAXNVfJWidBxTqxsz92jNMu4z9Kxm4RhUKtC
JoN9W5ch6UmxKMNjwM1eTxt25dS89kViHLP1vmMmigF3qHDTn1jg3zG3eW7odZmXdG6db6zi8
fHF1VVr71fyjUdufnR61KNy9aki4b8xw6NK4NJvpXeLosJwPyoqkR58UPQzNtKBpa3kjqW3s4
FZaz4ci4UN4nuqCWrCyhRxGC3KgwE8m
fgCyPsumUeb8dX4uiFxpyNf1xSDAZSLLzHEyoTAXNVfJWidBxTqxsz92jNMu4z9Kxm4RhUKtC
JoN9W5ch6UmxKMNjwM1eTxt25dS89kViHLP1vmMmigF3qHDTn1jg3zG3eW7odZmXdG6db6zi8
fHF1VVr71fyjUdufnR61KNy9aki4b8xw6NK4NJvpXeLosJwPyoqkR58UPQzNtKBpa3kjqW3s4
FZaz4ci4UN4nuqCWrCyhRxGC3KgwE8m
fgCyPsumUeb8dX4uiFxpyNf1xSDAZSLLzHEyoTAXNVfJWidBxTqxsz92jNMu4z9Kxm4RhUKtC
JoN9W5ch6UmxKMNjwM1eTxt25dS89kViHLP1vmMmigF3qHDTn1jg3zG3eW7odZmXdG6db6zi8
fHF1VVr71fyjUdufnR61KNy9aki4b8xw6NK4NJvpXeLosJwPyoqkR58UPQzNtKBpa3kjqW3s4
FZaz4ci4UN4nuqCWrCyhRxGC3KgwE8m
fgCyPsumUeb8dX4uiFxpyNf1xSDAZSLLzHEyoTAXNVfJWidBxTqxsz92jNMu4z9Kxm4RhUKtC
JoN9W5ch6UmxKMNjwM1eTxt25dS89kViHLP1vmMmigF3qHDTn1jg3zG3eW7odZmXdG6db6zi8
fHF1VVr71fyjUdufnR61KNy9aki4b8xw6NK4NJvpXeLosJwPyoqkR58UPQzNtKBpa3kjqW3s4
FZaz4ci4UN4nuqCWrCyhRxGC3KgwE8m
mumuzi@kali:~/桌面/py file$
```


将最后那一串拿去base64解码，发现不对，长度为251却没有等号，进行base58解码发现解码成了emoji。

fgCyPsumUeb8dX4uiFxpyNf1xSDAZSLLzHEyoTAXNVfJWidBxTqxsz92jNMu4z9Kxm4RhUKtCJoN9W5ch6UmxKMNjwM1eTxt25dS89kViHLP1vmMmigF3qHDTn1jg3zG3eW7odZmXdG6db6zi8fHF1VVr71fyjUdufnR61KNy9aki4b8xw6NK4NJvpXeLosJwPyoqkR58UPQzNtKBpa3kjqW3s4FZaz4ci4UN4nuqCWrCyhRxGC3KgwE8m

编码Base58> 解码Base58>

转换后:



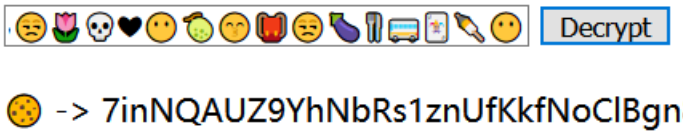
第六层：伤脑的emoji

得到emoji后，进行了大量的尝试和翻阅，最后发现
http://taqini.space/codemoji/#/encrypt
这个符合条件，但是遇到两个问题

> 第一：只能编码无法手动解码，除非分享链接，但是解码内容固定，我当时想把密文加密再然后再去解密，但是这样并不知道加密方式，无法解密
> 第二：密文是加密形式，不知道是那种加密方式，这里就无法进行正确的解密

最后，在重重又重重又重重又重重又重重又重重又重重又重重的努力下，找到了个爆破的神器，大小仅100k不到。
https://github.com/pavelvodrazka/ctf-writeups/tree/master/hackyeaster2018/challenges/egg17

下载下来，打开cracker.html，输入你的emoji，直接爆破，他会给你正确的加密方式和密文



得到加密方式和密文后，去官网http://taqini.space/codemoji/#/encrypt进行检验，发现正确，这一步终于大功告成，flag就在眼前？

第七层：奇怪的base62

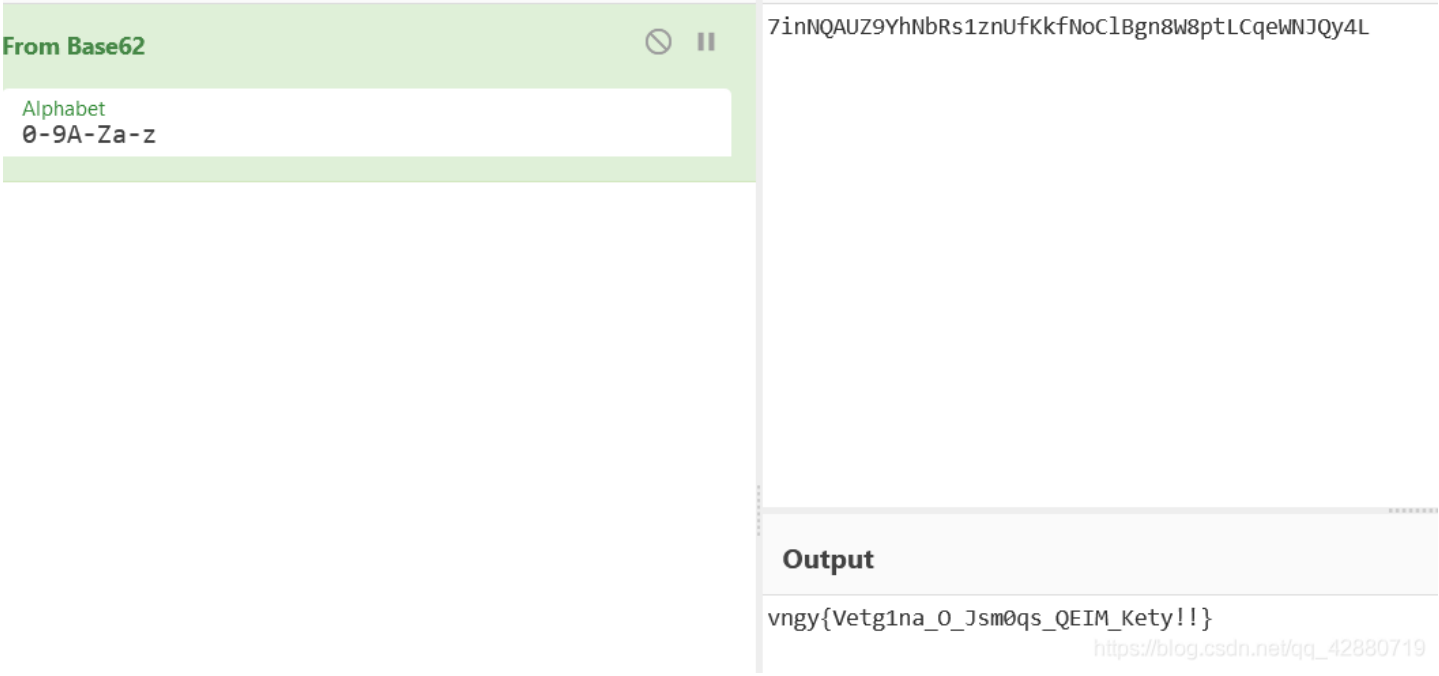
这里也是一大卡点，解出来了之后，放不同的平台都解不出来，但是能确定的是这不是64或者58

长度46，已经不满足base64了（除非自己添加==）

字符串中含有字母l，不满足58

有大小写数字的就只剩下base62

但是这里遇到瓶颈，找了几个base62平台，解出来分别是数字、数字、乱码、直接报错，只有CyberChef解出来是正确的（之后补充原因）



第八层：仿射密码

> hint3:y=ax+b,a是斜率3，b是常数6

这里得注意，他得到的值包含大小写和数字，有的平台或者软件是解不出来的，要找的正确的工具，这里还是用CyberChef。

```
vngy{Vetg1na_O_Jsm0qs_QEIM_Kety!!}
```

Output

```
flag{Fina1ly_U_Bec0me_MISC_King!!}
```


得到flag
套娃好玩，希望简单亿点点。