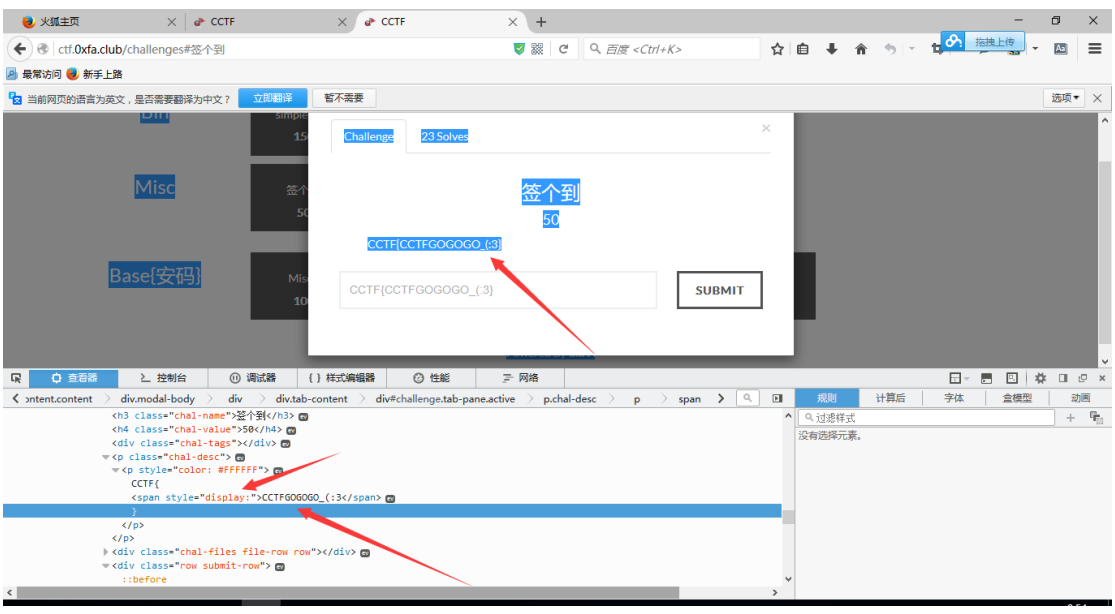
你丑你先睡

0x01 签到题

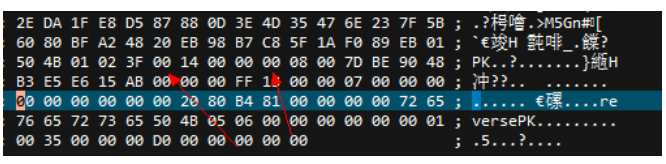
签到的位置。。。提交框上面有东西 CCTF{，用鼠标选中就能看见，f12 找到那串内容的元素，向下看，有个 display:none，把 none 去了，flag 在刚才 CCTF{那个地方就完整了，复制，粘贴。



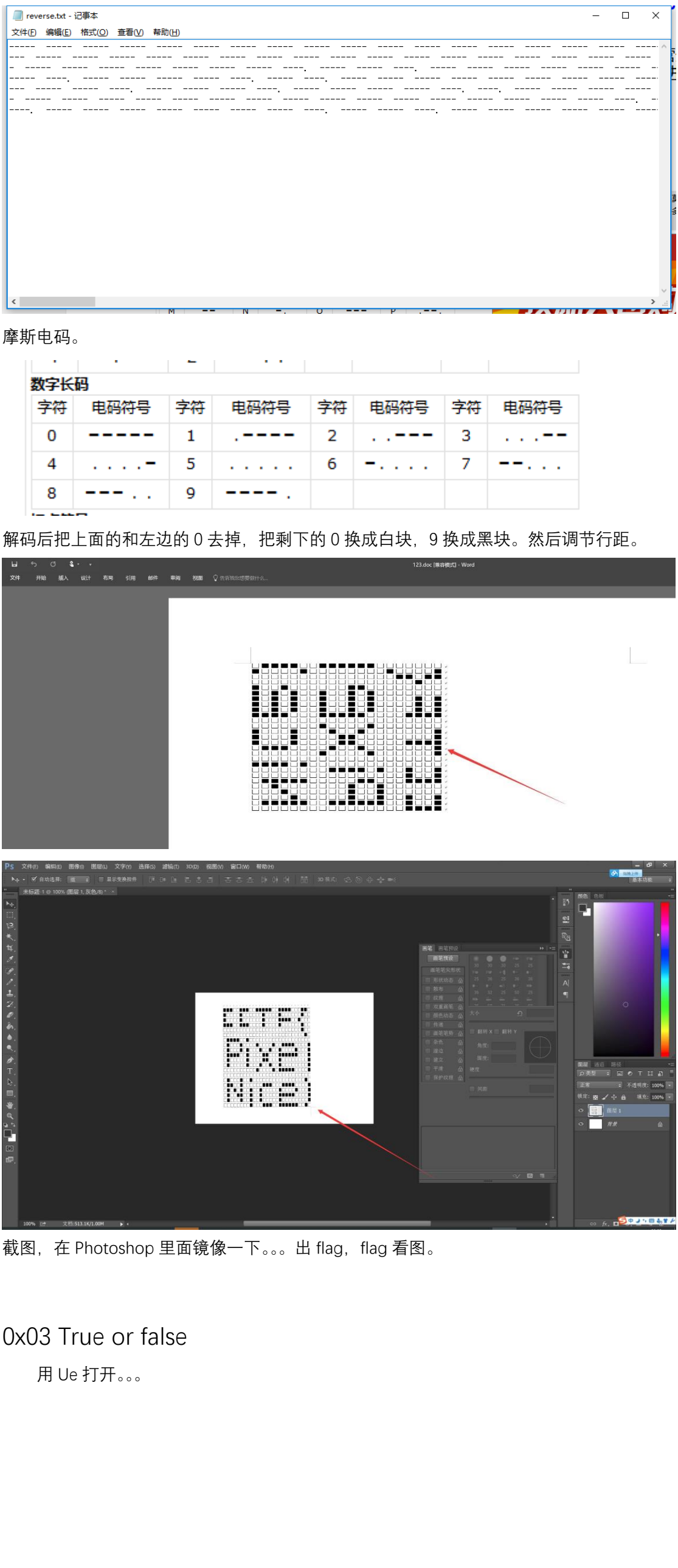
你问我要 flag？看图就好骚年。

0x02 Best_easy_misc

这题脑洞略大



试了很久，zip 伪加密。。。。加密位改成 0 就好，解压出 reserve.txt



摩斯电码。

数字长码

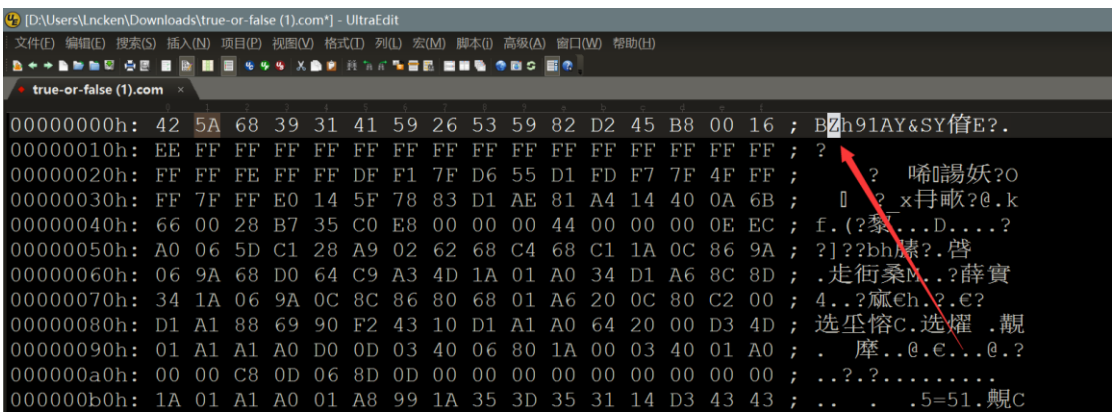
字符	电码符号	字符	电码符号	字符	电码符号	字符	电码符号
0	-----	1	.----	2	. .---	3	. . .--
4	. . . .-	5	. . . . .	6	-. . . .	7	--. . .
8	---. .	9	----.				

解码后把上面的和左边的 0 去掉，把剩下的 0 换成白块，9 换成黑块。然后调节行距。

截图，在 Photoshop 里面镜像一下。。。出 flag，flag 看图。

0x03 True or false

用 Ue 打开。。。

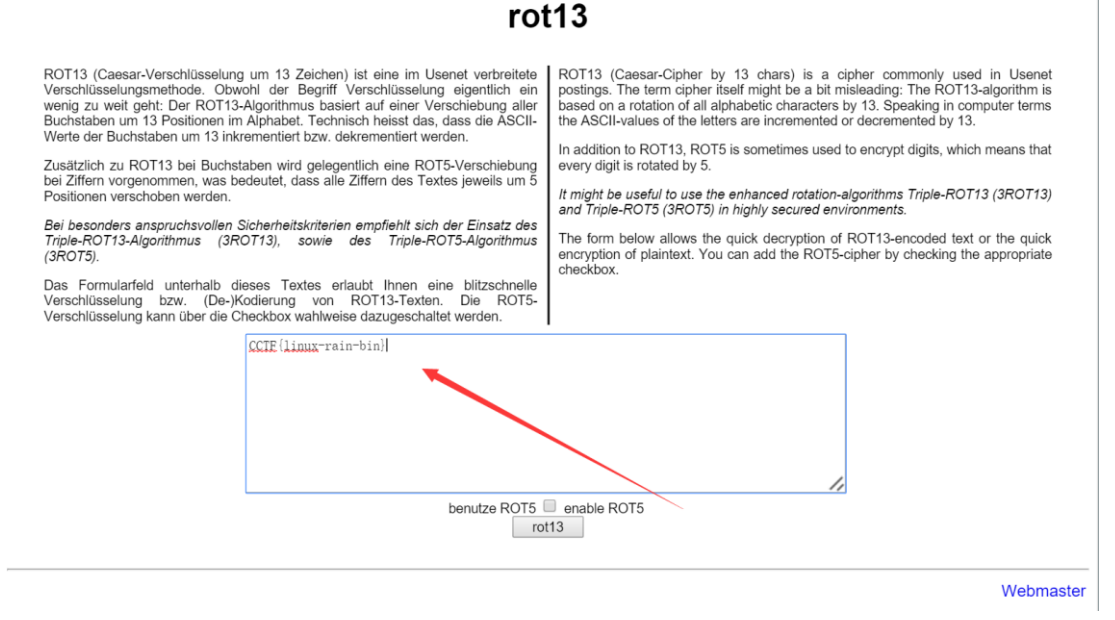


看文件头感觉有点像 BZ2。。。把头稍微修改成 BZ2 的。。。拉到 linux 下。。。。。

tar –jxvf 解压。。。。。。

然后得到两个可执行文件，true 和 false。。。。

拉回 Windows 进 IDA。。。。对比，发现，看图。。。



True 比 false 多了个 printf 函数。。进去 F5 反编译。。。

```
int print_f()
{
  char v1; // [sp+10h] [bp-20h]@1
  char v2; // [sp+11h] [bp-1Fh]@1
  char v3; // [sp+12h] [bp-1Eh]@1
  char v4; // [sp+13h] [bp-1Dh]@1
  char v5; // [sp+14h] [bp-1Ch]@1
  char v6; // [sp+15h] [bp-1Bh]@1
  char v7; // [sp+16h] [bp-1Ah]@1
  char v8; // [sp+17h] [bp-19h]@1
  char v9; // [sp+18h] [bp-18h]@1
  char v10; // [sp+19h] [bp-17h]@1
  char v11; // [sp+1Ah] [bp-16h]@1
  char v12; // [sp+1Bh] [bp-15h]@1
  int v13; // [sp+24h] [bp-Ch]@1
  __int64 v14; // [sp+28h] [bp-8h]@1

  v14 = (__int64)"yvahk-enva";
  v1 = 80;
  v2 = 80;
  v3 = 71;
  v4 = 83;
  v5 = 123;
  v6 = 0;
  v7 = 45;
  v8 = 111;
  v9 = 118;
  v10 = 97;
  v11 = 125;
  v13 = 12;
  v12 = 0;
  return printf("%s%s%s\n", &v1, "yvahk-enva", &v7);
}
```

format: char[]

感觉像是 flag。。。80 80 71 83 123 从 asc2 转字符是 PPGS{

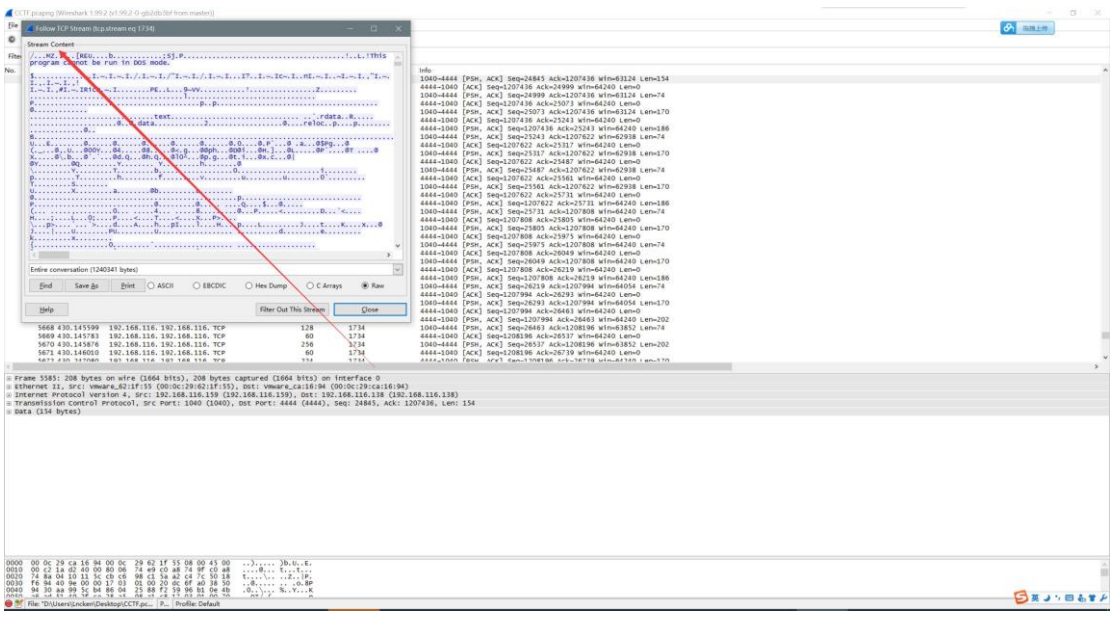
45 111 118 97 125 从 asc2 转字符是-ova}

所以 printf 出来的是 PPGS{yvahk-enva-ova}

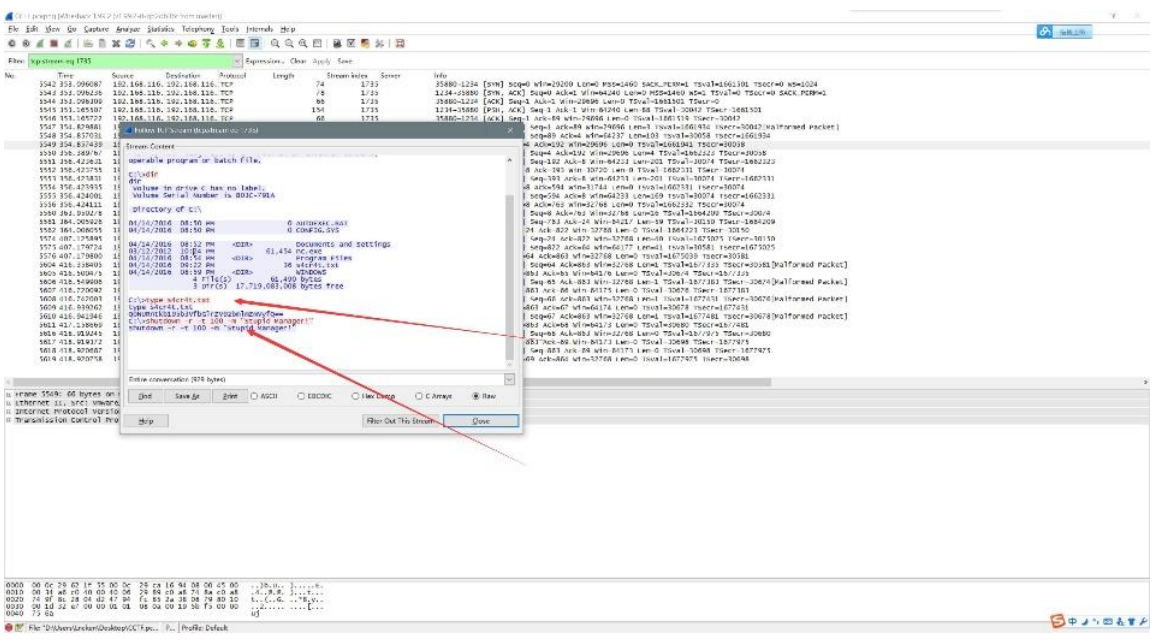
很像 flag 了，从 PPGS 应该对应 CCTF。。。那就是 rot-13 了

在线解一下。。。flag 看图。。。。

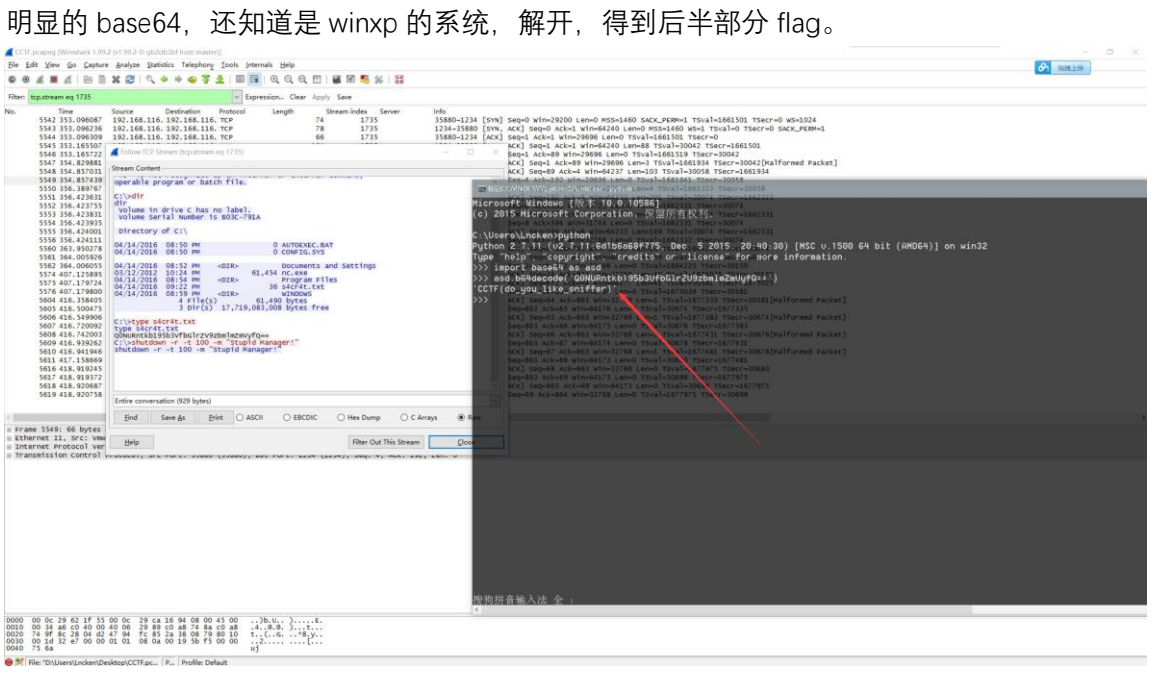
0x04 EZ_GAME



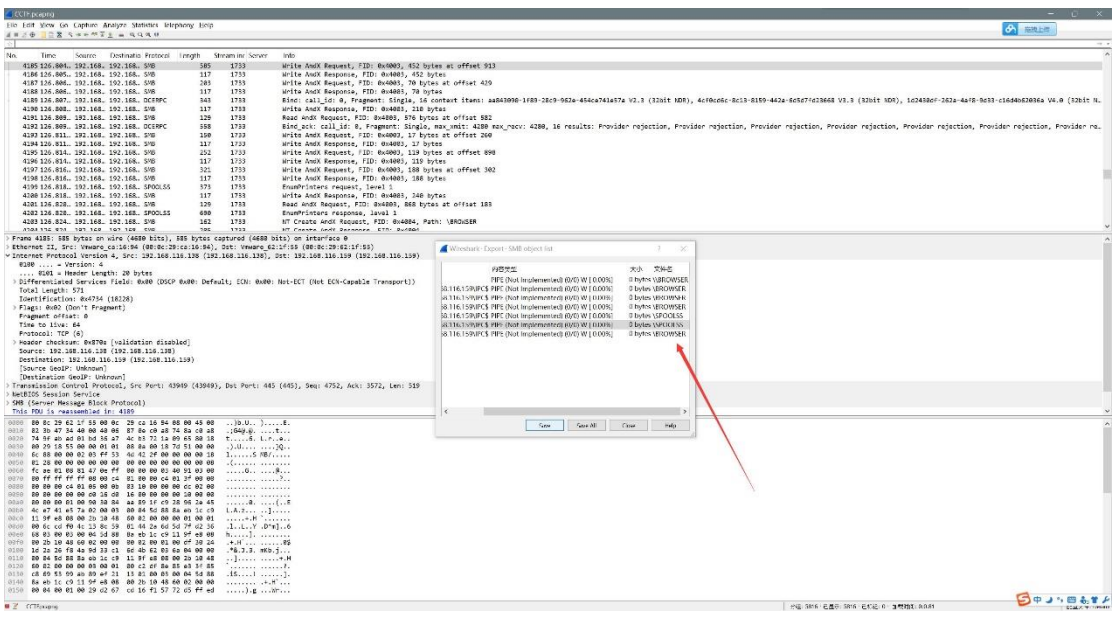
先观察，tcp 特别多。。。Follow tcp stream。看到 MZ，可执行文件，这个就是 exp，亲测 XP 下虚拟机可用。然后往后边的 tcp 链翻，看到这个。



明显的 base64，还知道是 winxp 的系统，解开，得到后半部分 flag。



一开始想猜洞，没成功，仔细一看，有 SMB 过程看文件名 SPOOL 和 BROWSER，还知道是 winxp 的系统，



百度之，得到 ms08-067,格式倒是试半天。。。。ms08067，不对，MS 大写。。

ms08-067漏洞 远程溢出入侵测试 - 十年磨一剑,霜刃未曾试! - 博客...

[*] Nmap: |_ MS07-029: CHECK DISABLED (remove...SMBPIPE BROWSER yes The pipe name to use (BROWSER...如果想漏洞支持什么操作系统,可以输入info命令,就能...

blog.csdn.net/sysprogr... - 百度快照 - 88%好评

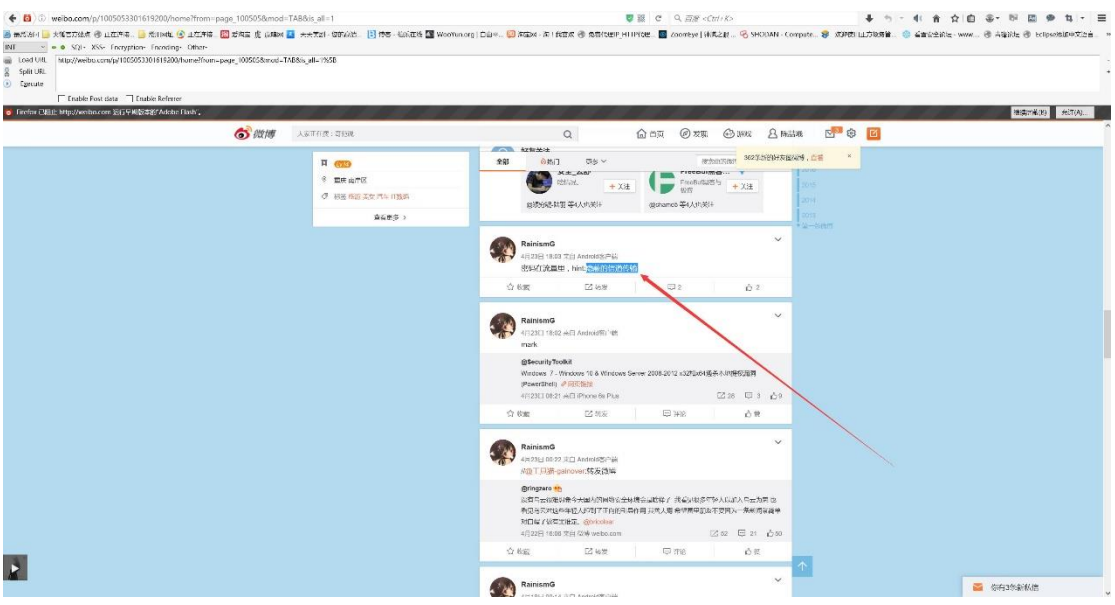
Flag 就不打了，图里有，没存。。。。太长，不想再打一遍。。

0x05　5ecr3t

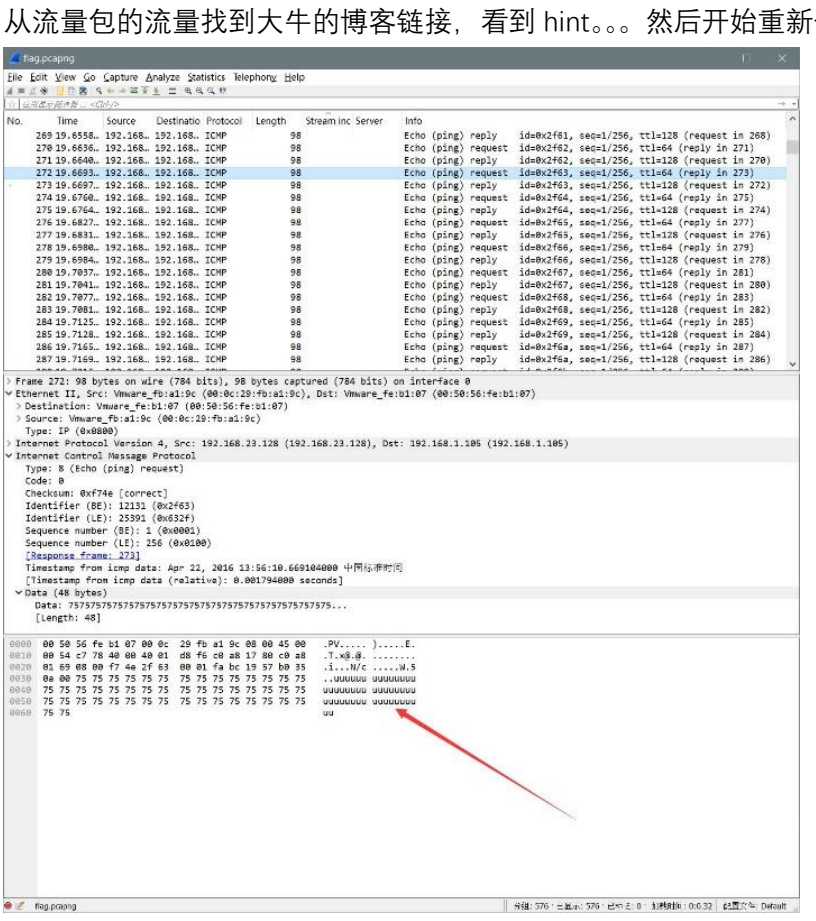
这题脑洞更大。。。

先是 winpcap 导出所有 http。。。。

发现有个压缩包，有密码



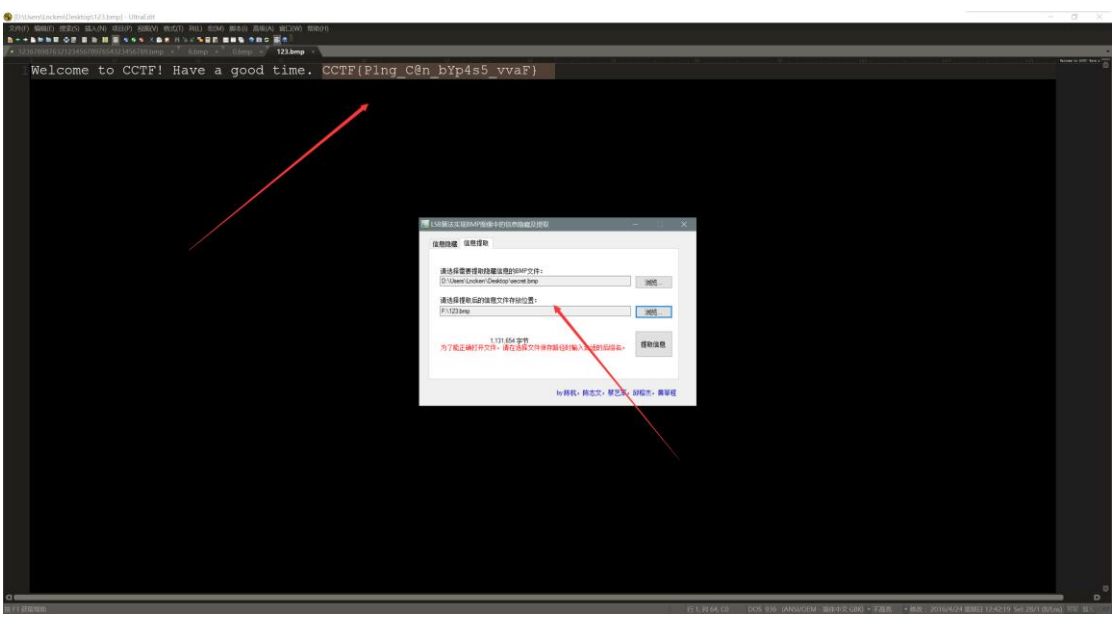
从流量包的流量找到大牛的博客链接，看到 hint。。。然后开始重新分析包。



发现 ping 包的数据块有点怪异，把每一帧的这个都提出来

就是上面那段，rot-13 解密得到密码，解压出文件

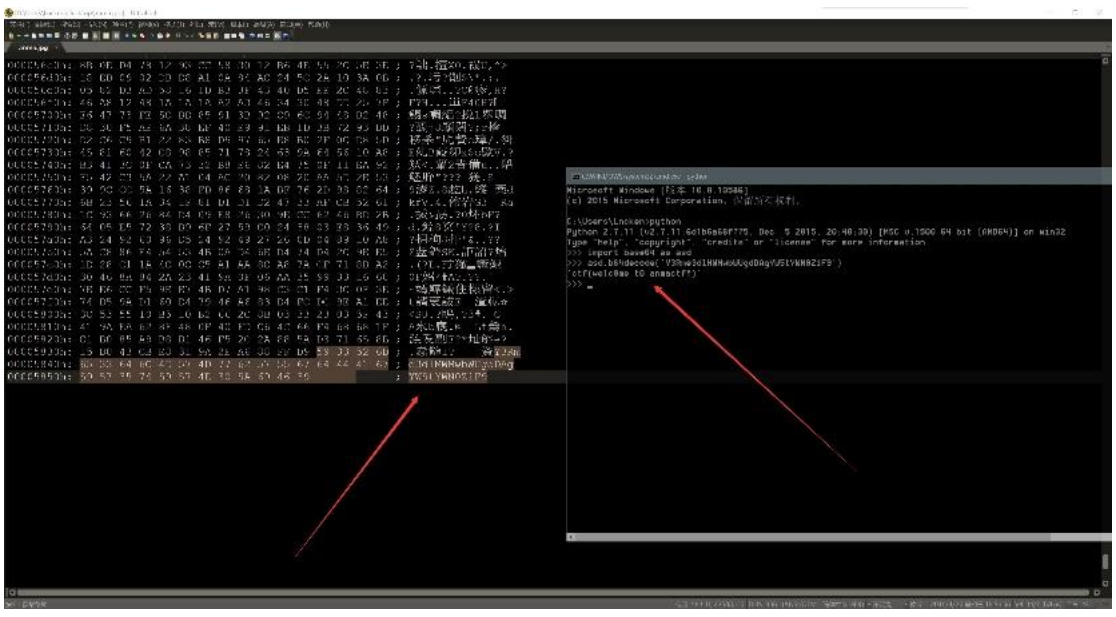
Bmp 文件，lsb 解密，上工具就好，得到 flag。



Flag 如图所示。

0x06 misc1

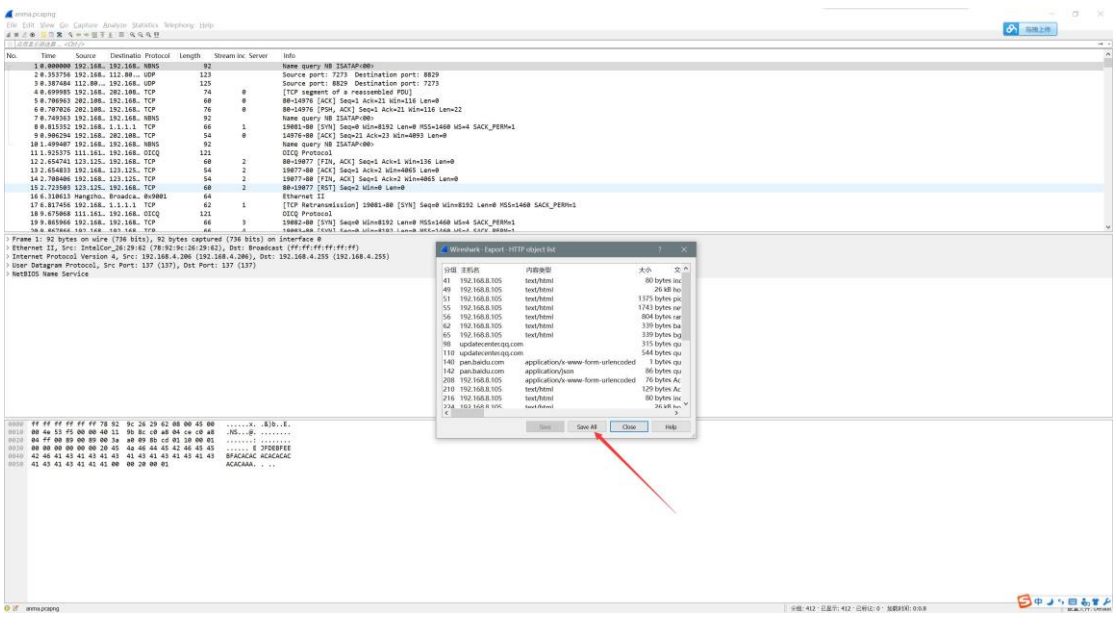
Ue 打开，文件尾，base64 解码。



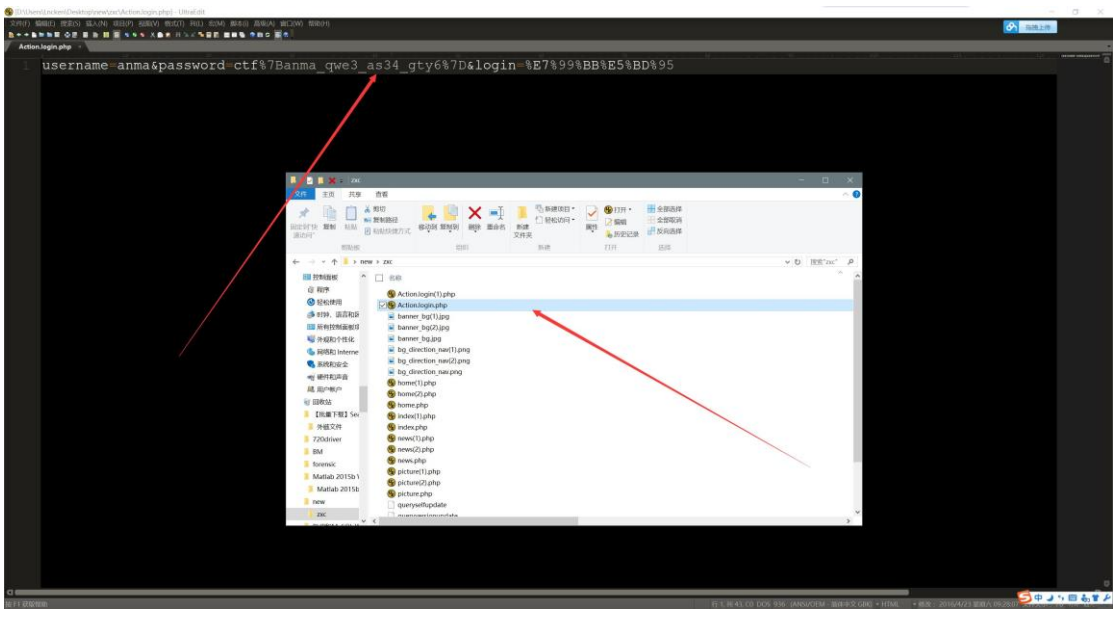
Flag？还是看图就好。

0x07 misc2

把所有的 http 导出。。



一个一个看。。发现有密码。。。还带了个 ctf，看上去就像 flag



url 编码。。。。。%7 是{，%7D 是} 所以，flag 还是看图。。。。

0x08 Re1

用 IDA 大概看下流程，命令行输入 2 各参数，一个随机数模一下参数个数，然后加 1 然后取这个数对应的第几个参数对应的字符串，md5_custom 这个函数什么都没干，check 函数里面和 0x8048c80 这个地址的字符串直接比较，相同就好了。但是我的 IDA 居然没有在 0x8048c80 这个地址没有识别出来交叉引用，不知道是什么黑科技，感觉挺牛逼的。

```
signed int __cdecl check(char *a1)
{
  signed int i; // [sp+Ch] [bp-4h]@1

  for ( i = 0; i <= 31; ++i )
  {
    if ( a1[i] != *(_BYTE *)(i + 0x8048C80) )
      return 0;
  }
  return 1;
}
```

```
.rodata:08048C63                                        ; DATA XREF: main:1
.rodata:08048C7F                        align 10h
.rodata:08048C80 aF2332291a6e1e6 db 'f2332291a6e1e6154f3cf4ad8b7504d8',0
.rodata:08048C80 _rodata                ends
.rodata:08048C80
```

Flag 就是：f2332291a6e1e6154f3cf4ad8b7504d8

0x09 Re2

.Net 程序，直接看源码，发现只需要在本地开一个端口不停地收，收完就好了。

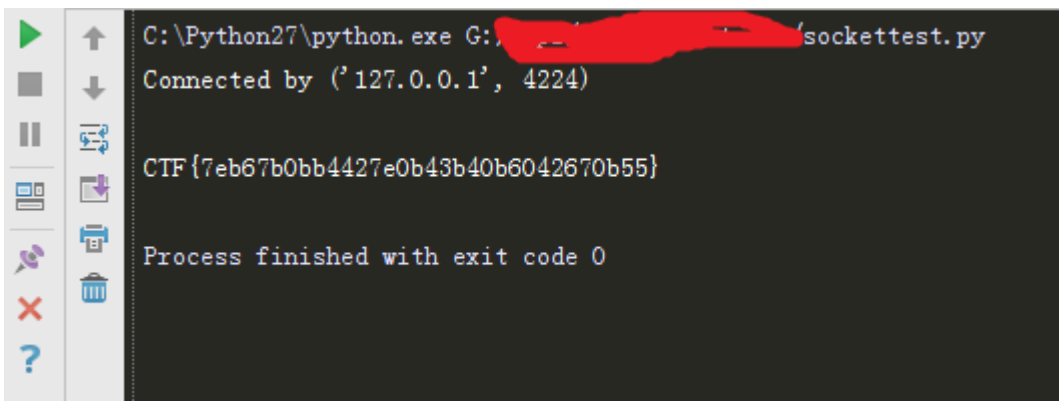
```
using System;
using System.Diagnostics;
using System.IO;
using System.Net.Sockets;
using System.Text;
```

```
namespace Rev_100
{
    internal class Program
    {
        private static void Main(string[] args)
        {
            string hostname = "127.0.0.1";
            int port = 31337;
            TcpClient tcpClient = new TcpClient();
            try
            {
                Console.WriteLine("Connecting...");
                tcpClient.Connect(hostname, port);
            }
            catch (Exception)
            {
                Console.WriteLine("Cannot connect!\nFail!");
                return;
            }
            Socket client = tcpClient.Client;
            string text = "Super Secret Key";
            string text2 = Program.read();
            client.Send(Encoding.ASCII.GetBytes("CTF{"));
            string text3 = text;
            for (int i = 0; i < text3.Length; i++)
            {
                char x = text3[i];
                client.Send(Encoding.ASCII.GetBytes(Program.search(x, text2)));
            }
            client.Send(Encoding.ASCII.GetBytes("}"));
            client.Close();
            tcpClient.Close();
            Console.WriteLine("Success!");
        }
        private static string read()
        {
            string fileName = Process.GetCurrentProcess().MainModule.FileName;
            string[] array = fileName.Split(new char[]
            {
                '\\'
            });
            string path = array[array.Length - 1];
            string result = "";
            using (StreamReader streamReader = new StreamReader(path))
            {
                result = streamReader.ReadToEnd();
            }
            return result;
        }
        private static string search(char x, string text)
        {
            int length = text.Length;
            for (int i = 0; i < length; i++)
            {
                if (x == text[i])
                {
                    int value = i * 1337 % 256;
                    return Convert.ToString(value, 16).PadLeft(2, '0');
                }
            }
            return "??";
        }
    }
}
```

```
import socket


HOST = ''                    # Symbolic name
meaning all available interfaces
PORT = 31337                 # Arbitrary non-
privileged port
s        =        socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(1)
conn, addr = s.accept()
print 'Connected by', addr
```

```python
s=''
while 1:
    data=conn.recv(1)
    if data:
        s=s+data
    else:
        break
print s
conn.close()
```



0x10 Re3

不知道要干嘛，反正我是没看懂，大概按照顺序输了几个数字就好了。。。

Ida f5 代码：

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int result; // eax@1
  char *Dest; // [sp+0h] [bp-148h]@0
  char Str1; // [sp+40h] [bp-108h]@3
  int v6; // [sp+B8h] [bp-90h]@3
  int v7; // [sp+BCh] [bp-8Ch]@3
  __int16 v8; // [sp+C0h] [bp-88h]@3
  char v9; // [sp+C2h] [bp-86h]@3
  int v10; // [sp+D8h] [bp-70h]@3
  int v11; // [sp+DCh] [bp-6Ch]@3
  int v12; // [sp+E0h] [bp-68h]@3
  int v13; // [sp+E4h] [bp-64h]@3
  int v14; // [sp+E8h] [bp-60h]@3
  int v15; // [sp+ECh] [bp-5Ch]@3
  char Str2[4]; // [sp+F0h] [bp-58h]@3
  char v17[4]; // [sp+100h] [bp-48h]@3
  char v18[4]; // [sp+118h] [bp-30h]@3
  char v19[4]; // [sp+120h] [bp-28h]@3
  int v20; // [sp+13Ch] [bp-Ch]@1

  _alloca((size_t)Dest);
  result = __main();
  v20 = 1;
  while ( v20 )
  {
    strcpy(v19, "123456");
    strcpy(v18, "7891011");
    strcpy(v17, "12131415");
    strcpy(Str2, "numbers:");
    v15 = 7365741;
    v14 = 7631717;
    v13 = 7237475;
    v12 = 7627107;
    v11 = 6647397;
    v10 = 6582895;
    v8 = 25970;
    v9 = 0;
    v7 = 3219507;
    v6 = 3285044;
    strcat(Str2, v18);
    strcat(Str2, v17);
    strcat(Str2, v19);
    printf("Trouvez moi si vous pouvez\n");
    scanf("%s", &Str1);
    if ( !strcmp(&Str1, Str2) )
    {
      v20 = 0;
      printf(
        "oh mon dieu t'as reussi bravo ! %s %s %s%s%s %s%s %s %s %s \n",
        &v15,
        &v14,
        &v13,
        &v12,
```

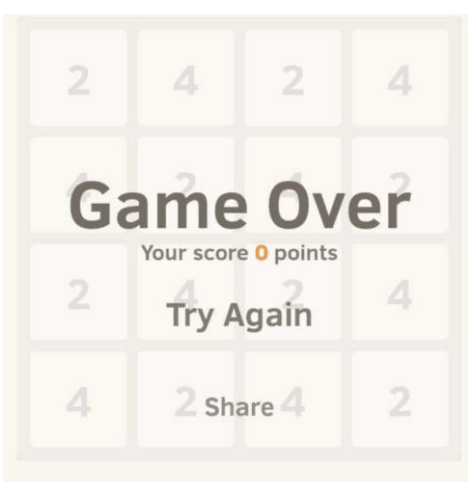
```
            &v11,
            &v10,
            &v8,
            &v14,
            &v6,
            &v7);
      }
      else
      {
        printf("ratcherche encore!\n");
      }
      result = getch();
    }
    return result;
}
```

```
G:[redacted]tf\Reverse_100T3>Reverse3.exe
Trouvez moi si vous pouvez
numbers:789101112131415123456
oh mon dieu t'as reussi bravo ! mdp est concatene ordre est 4 2 3 1
```

那个数字串试了下就是 flag 了。。。

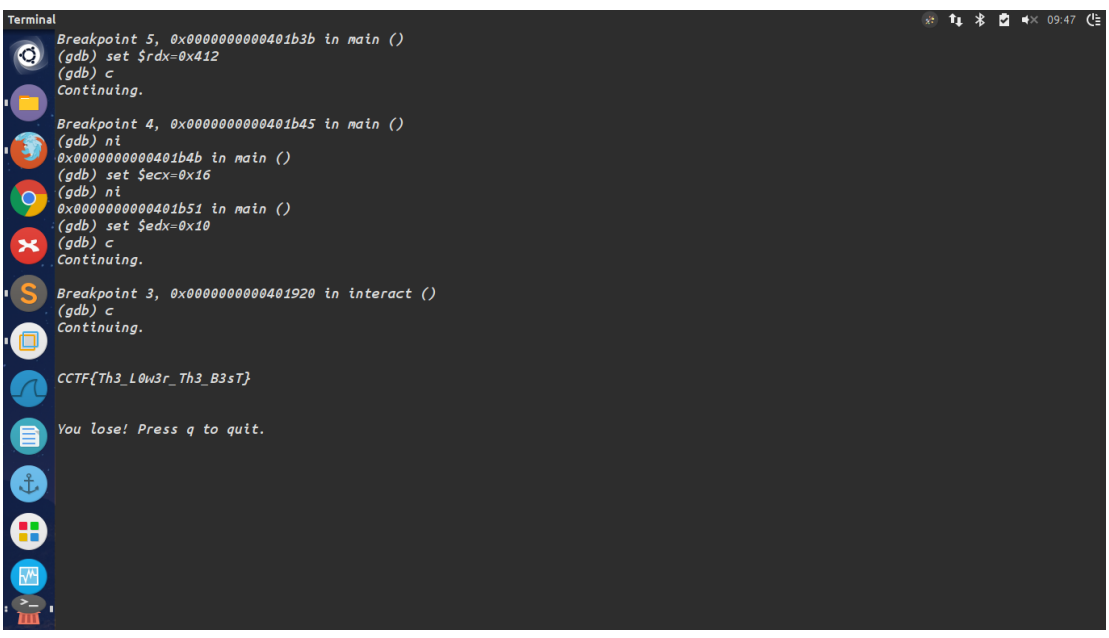
0x11 2048?4096

前前后后给了好多 Hint。。。总结就是分数越低越好。。。
百度一下，毕竟这么火的游戏。。。



在知乎看到 0 分这个。。。然后算了一下分数和步数。。
ida 发现是会往网络发数据的。。。抓包发现其实那个 not cheat 是网上来的。。
改寄存器分数和步数。。。not cheat。。。猜测服务器判定跟时间有关系，改游戏时间。。
贴上 gdb 过程指令。。。全部截图太多惹。。。。

```
b  main
b  *0x401abb
r
c
set  $eax  =  1
b  *0x401920
b  *0x401b45
b  *0x401b3b
c
set  $rdx  =  0x412
c
ni
set  $ecx  =  0x16
ni
set  $edx  =  0x10
c
c
```



你又问我 flag 是啥？看图 ！！！！

0x12 LOLI1 & LOLI2

题目给了个网址 http://www.loli.club/
撸上去啥都没有。。另外、hint 的颜色居然和背景色一样、提示找 blog
看源码，有这样一段

```
<!--
```

powered by PockyNya
诚招前端，请联系邮箱：pocky@loli.club
-->
各种误解，最后根据 RicterZ 的 githubhttps://github.com/RicterZ 找到了 PockyNya 的 Githubhttps://github.com/PockyNya
在 PockyNya 的 Github 里面有两个项目 minecraft-bot 和 pyprint。
进入 pyprint 就能看到 PockyNya 的 Blog 地址了。。我也是醉了
下载源码审计

```
class AddPostHandler(BaseHandler):
    @tornado.web.authenticated
    def get(self):
        self.background_render('add_post.html', post=None)

    def post(self):
        title = self.get_argument('title', None)
        content = self.get_argument('content', None)
        tags = self.get_argument('tags', '').strip().split(',')
        if not title or not content:
            return self.redirect('/kamisama/posts/add')

        post = self.orm.query(Post.title).filter(Post.title == title).all()
        if post:
            return self.write('<script>alert("Title has already existed");window.history.go(-1);</script>')
        self.orm.add(Post(title=title, content=content, created_time=date.today()))
        self.orm.commit()
        return self.redirect('/kamisama/posts')
```

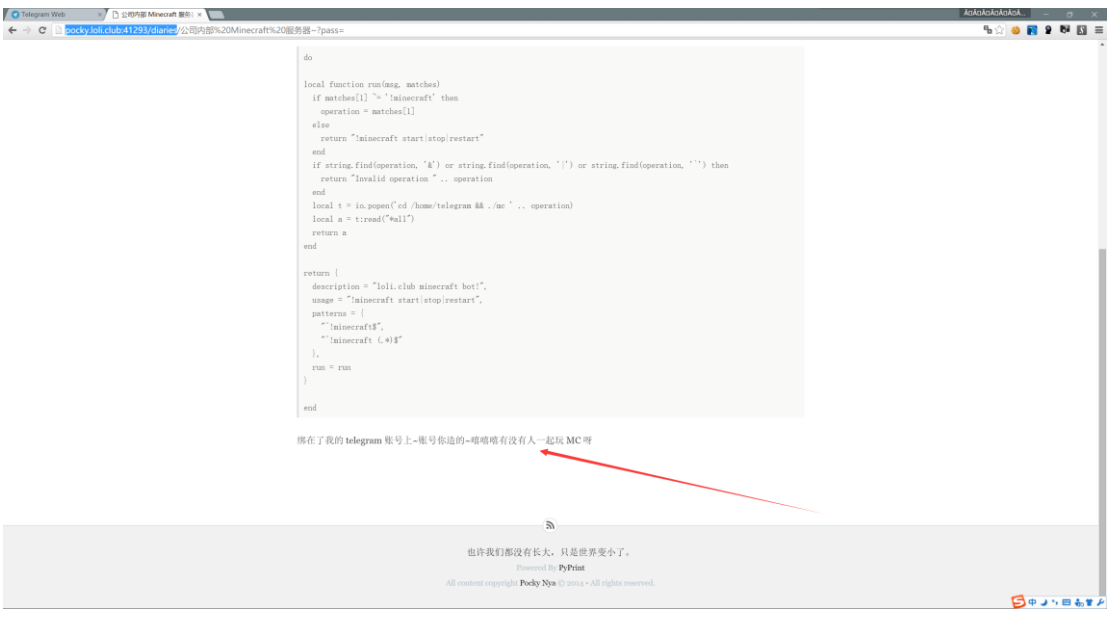
发表文章居然不检查权限，立马来了一波 XSS。。。分分钟拿到了在 COOKIE 的 Flag。unhex 就行了

```
Cookie:flag=434354467b434f44455f41554449545f425553544552537d;username="2|1:0|10:1461382264|8:username|12:cG9ja3lueWE=|d4e540d298981e80bd48150453751ef3db7a18611d2748f0f1d8cee4484d4958"
```

Flag：CCTF{CODE_AUDIT_BUSTERS}

0x13 loli-1

在 pyprintde 的项目下面继续审计。。发现有个 diaries。。日记？？rr 的内心世界？？还是 rr 日常发女装照片的地方？哦，想想就有点小激动耶！！！
http://pocky.loli.club:41293/diaries
发现只有一篇日记。。失望。没有女装（失望脸）。。。
日记里面是一个 telegram 的机器人。。代码都贴出来了。。。。。

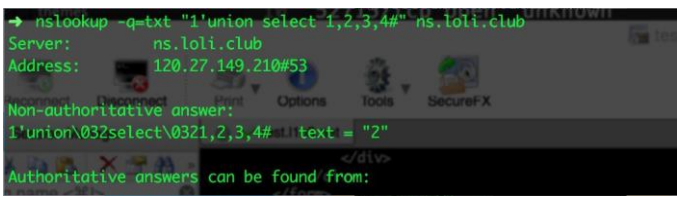


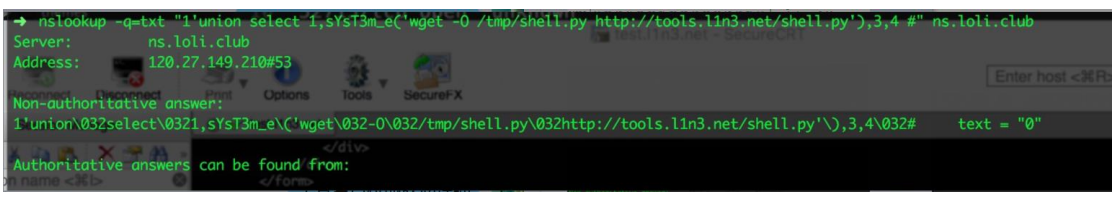
代码虽然不是很懂，但勉强还是能看出来是命令注入漏洞，过滤了&、|、`、%$，然而还是可以用那个;截断的。。。
再从日记最下面的那句话。。。找到账号，然后看图就好，flag 见图。。。。。。

我保证我没有调戏机器人。。。

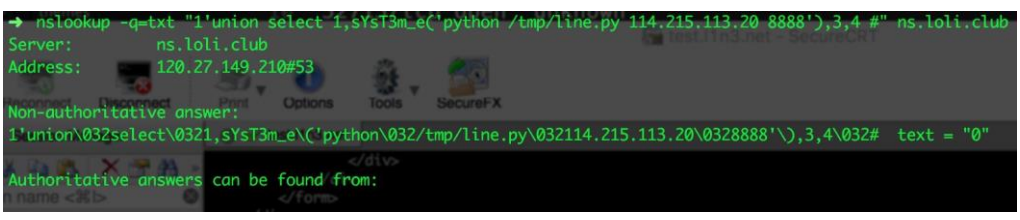
0x14 loli3

说好的 dns 说好的 txt 说好的 ns 记录，hint 还是很足的。。虽然我还是一脸蒙蔽了好久啊。。。
测试了很久，发现 dns 查询存在注入。。。

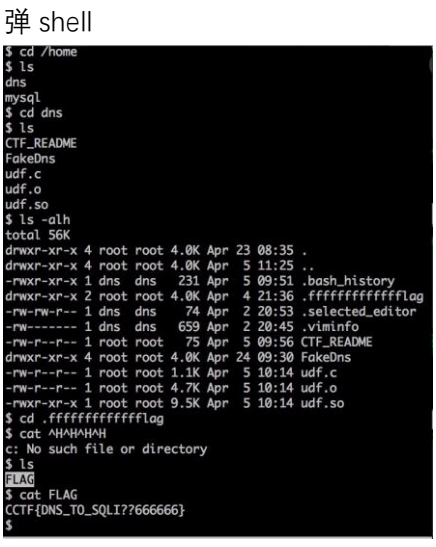




传脚本



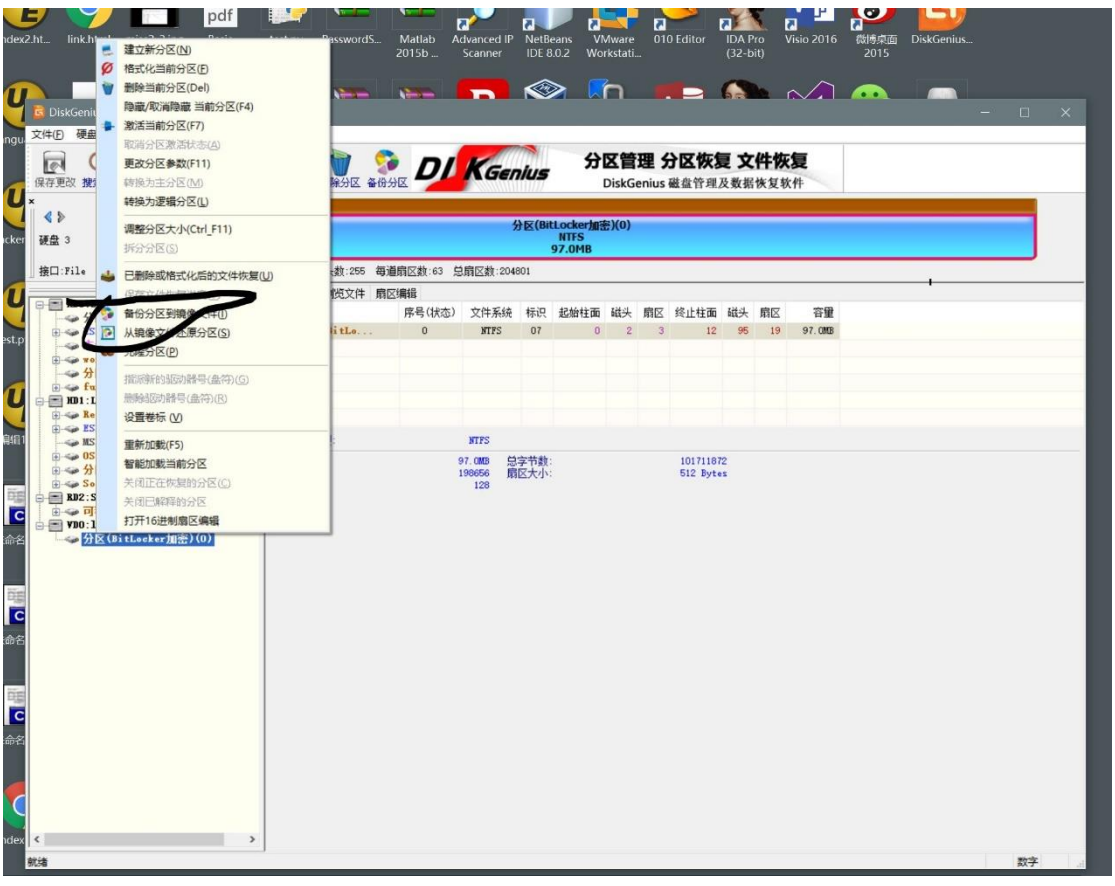
弹 shell



Flag 看图。。。

0x15　神秘文件 1

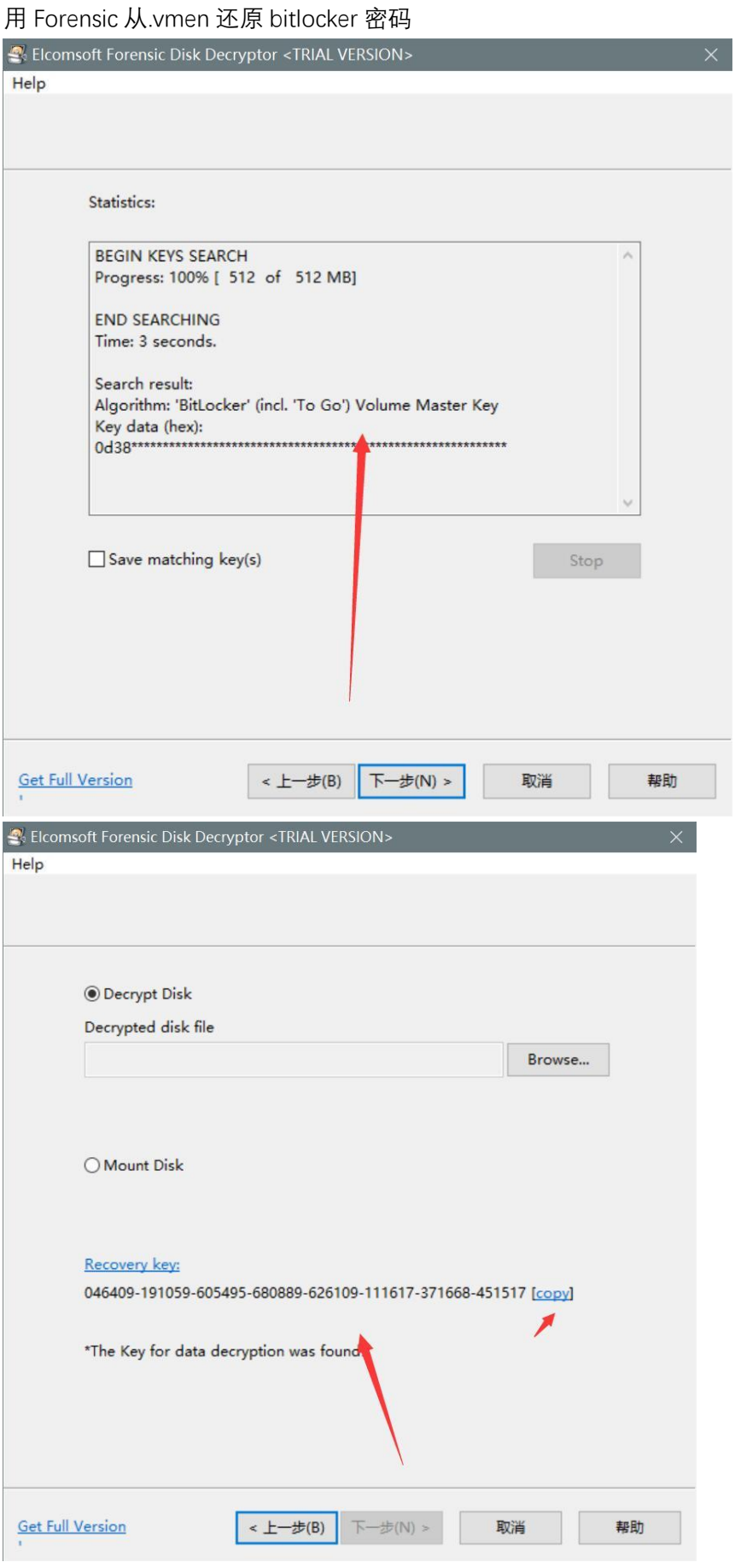
解压出 level1 和一个.vmen 文件。



先用 dg 打开，备份镜像。。。。

还原到物理磁盘，割物理磁盘分区时，扇区数要和镜像一致。。。(物理做题的我现在还没把硬盘还原回去。。。)

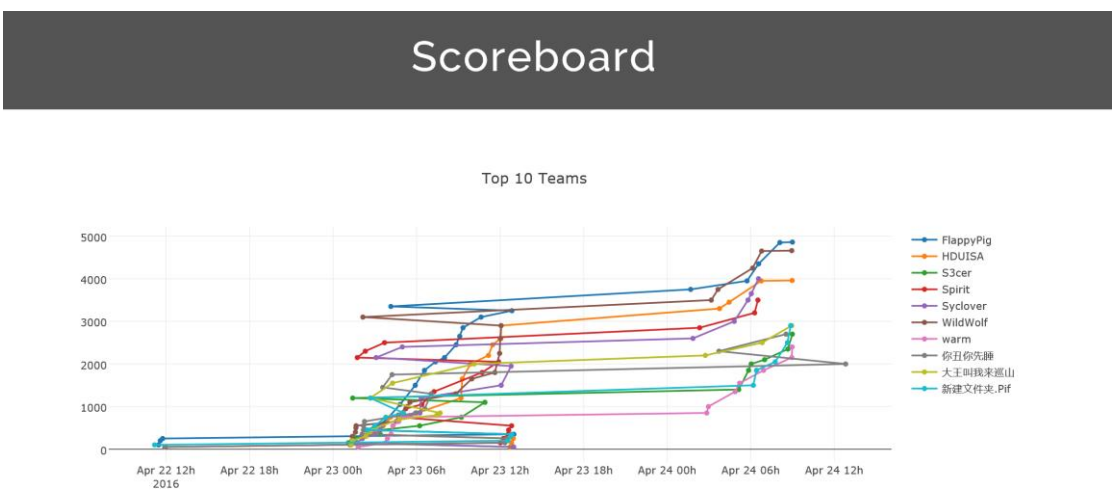
用 Forensic 从.vmen 还原 bitlocker 密码



直接用恢复密钥打开磁盘。。。

Flag 看图

0x16 写在最后



我一直觉得。。这个曲线。。很有艺术感。。