【CTF】记录一次CTF比赛的Writeup（附题目下载地址）

原创

TeamsSix 于 2019-09-25 17:16:15 发布 2122 收藏 12

分类专栏： Writeup 文章标签： CTF Writeup

本文链接：https://blog.csdn.net/qq_37683287/article/details/101374506

0x00 前言

最近因为省赛快来了，因此为实验室的小伙伴准备了这次比赛，总共10道题目，考虑到大多数小伙伴都刚从大一升到大二，因此整体难度不高，当然有几道难度还是有的。

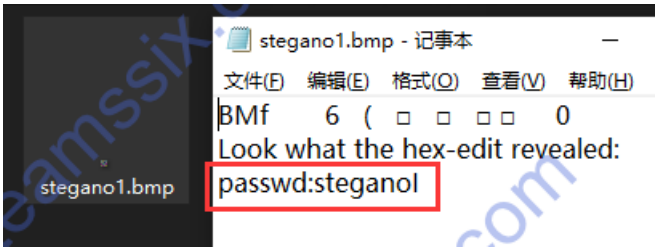
题目大多数都是从网上东找西找的，毕竟我也是个菜鸟呀，还要给他们出题，我太难了。

废话不多说，直接上Writeup吧，以下题目的文件下载地址可以在我的公众号（TeamsSix）回复CTF获取。

0x01 隐写 1

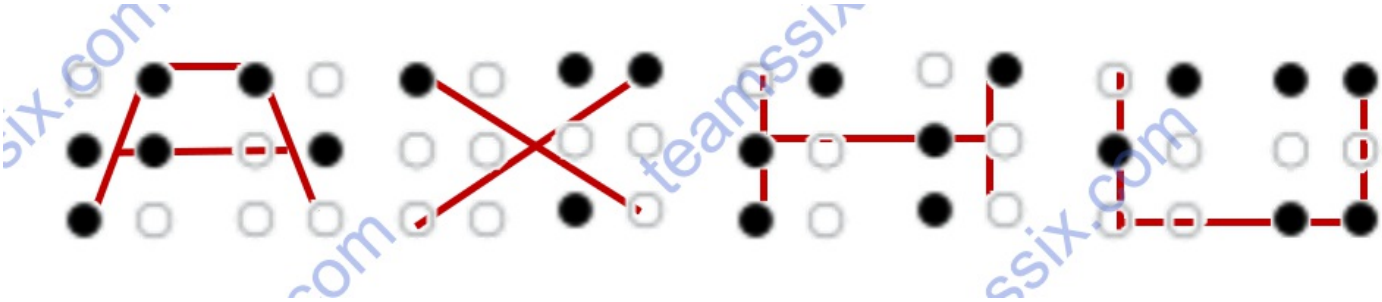
```
flag：steganoI

flag格式：passwd:

题目来源：http://www.wechall.net/challenge/training/stegano1/index.php
```

签到题，下载题目图片，利用记事本打开即可看到flag

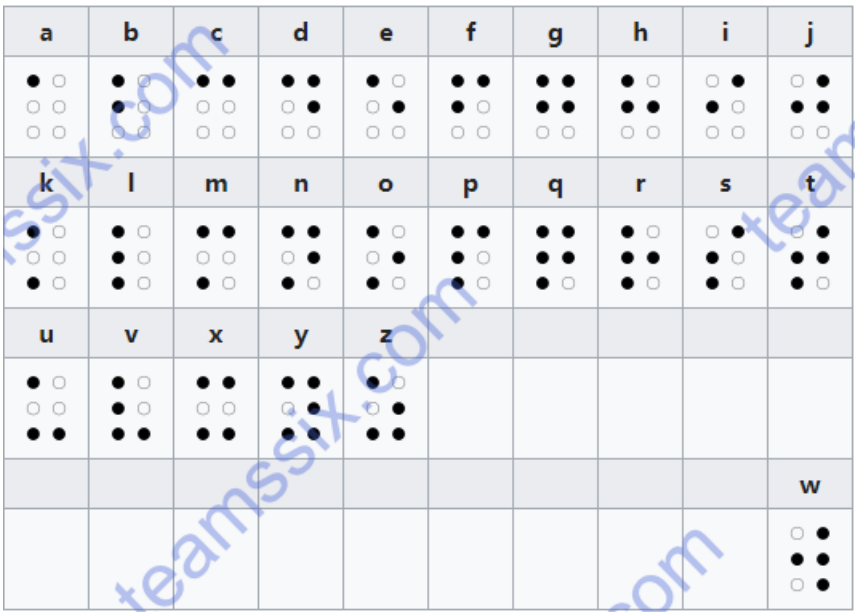


0x02 隐写 2

```
flag：teamssix

Hint:一般在公共场合才能看的见

题型参考：http://www.wechall.net/challenge/connect_the_dots/index.php
```



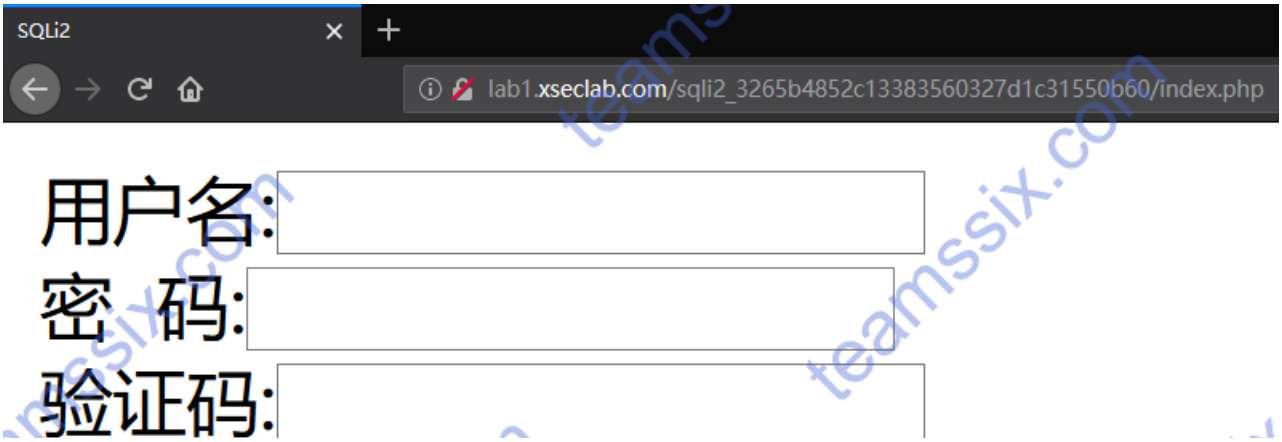
打开图片，参考题目提示说一般在公共场合才能看见，因此通过盲文对照表可以得出flag是teamssix，图片中的AXHU只是用来干扰的，这道题也是我参考wechall里面的一道题型。

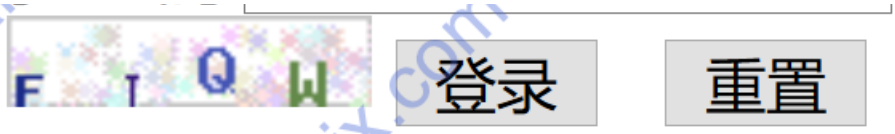


0x03 Web 1

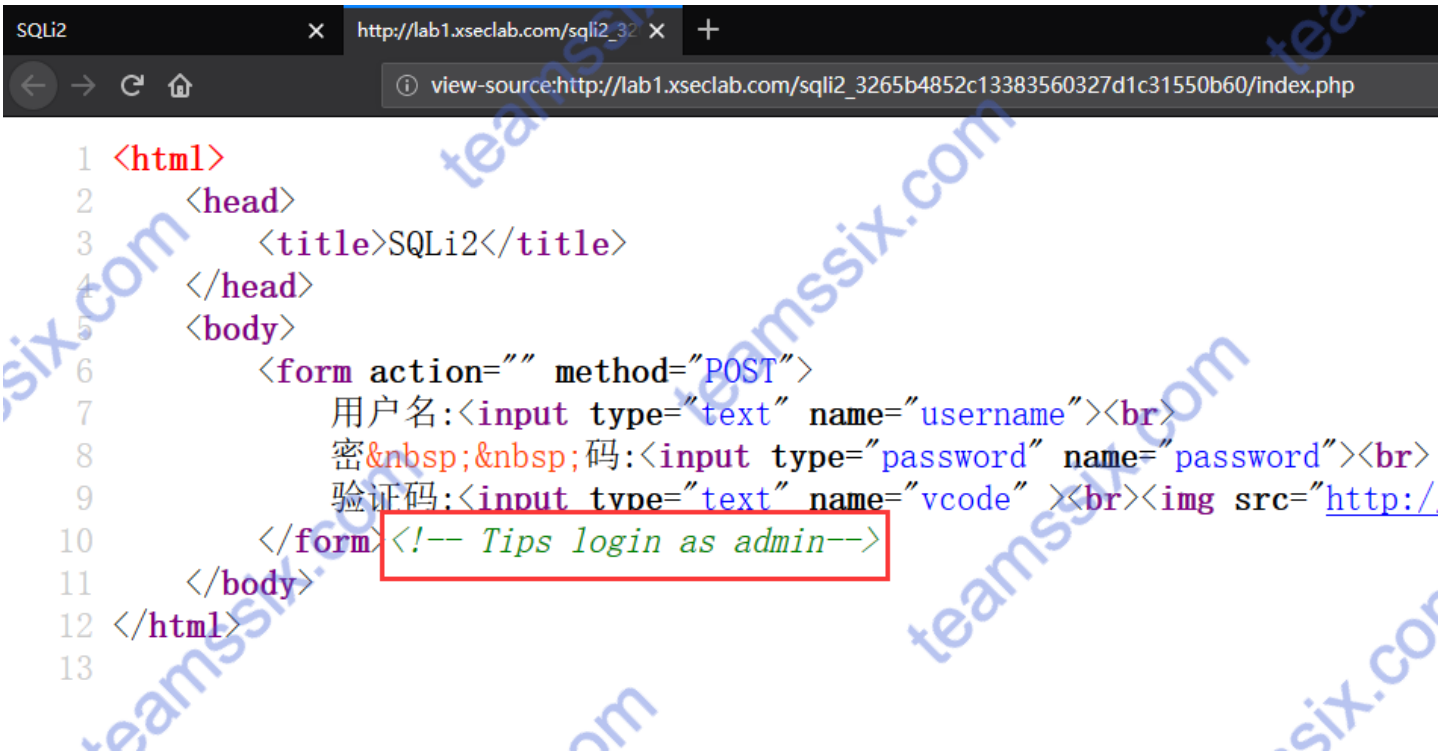
```
flag:iamflagsafsfskdf11223

Hint:站内有提示

题目地址：
http://lab1.xseclab.com/sqli2_3265b4852c13383560327d1c31550b60/index.php

参考来源：http://hackinglab.cn/ShowQues.php?type=sqlinject
```

1、打开题目地址



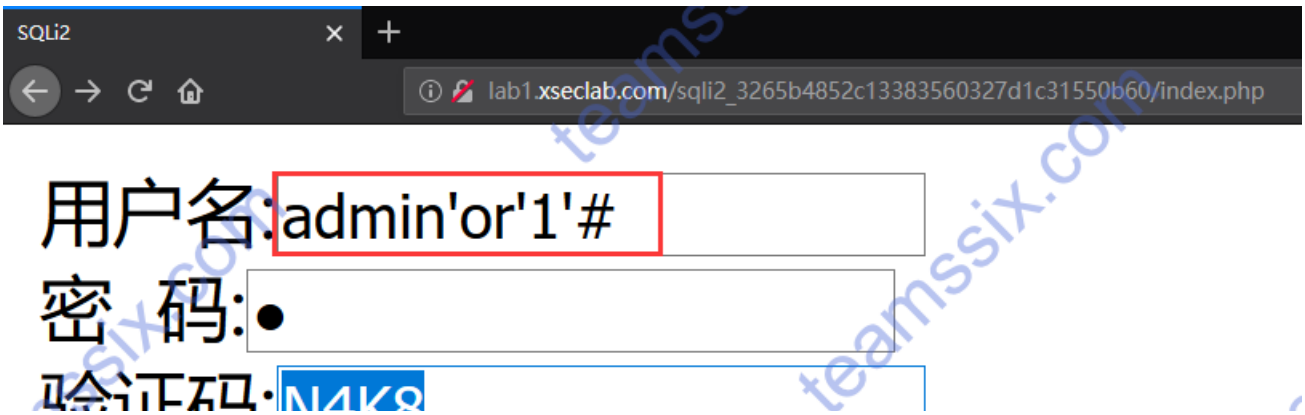


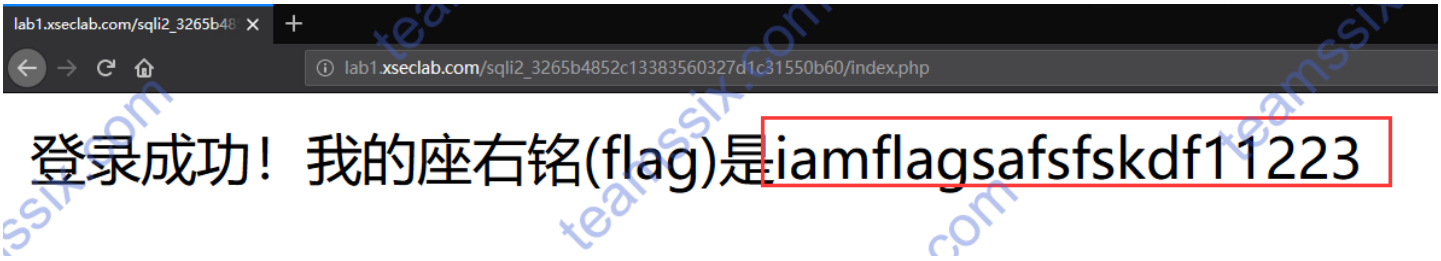
2、查看源码找到提示



3、根据提示使用admin登陆，并使用弱密码

4、尝试多次都提示失败，利用万能密码再做尝试，找到flag



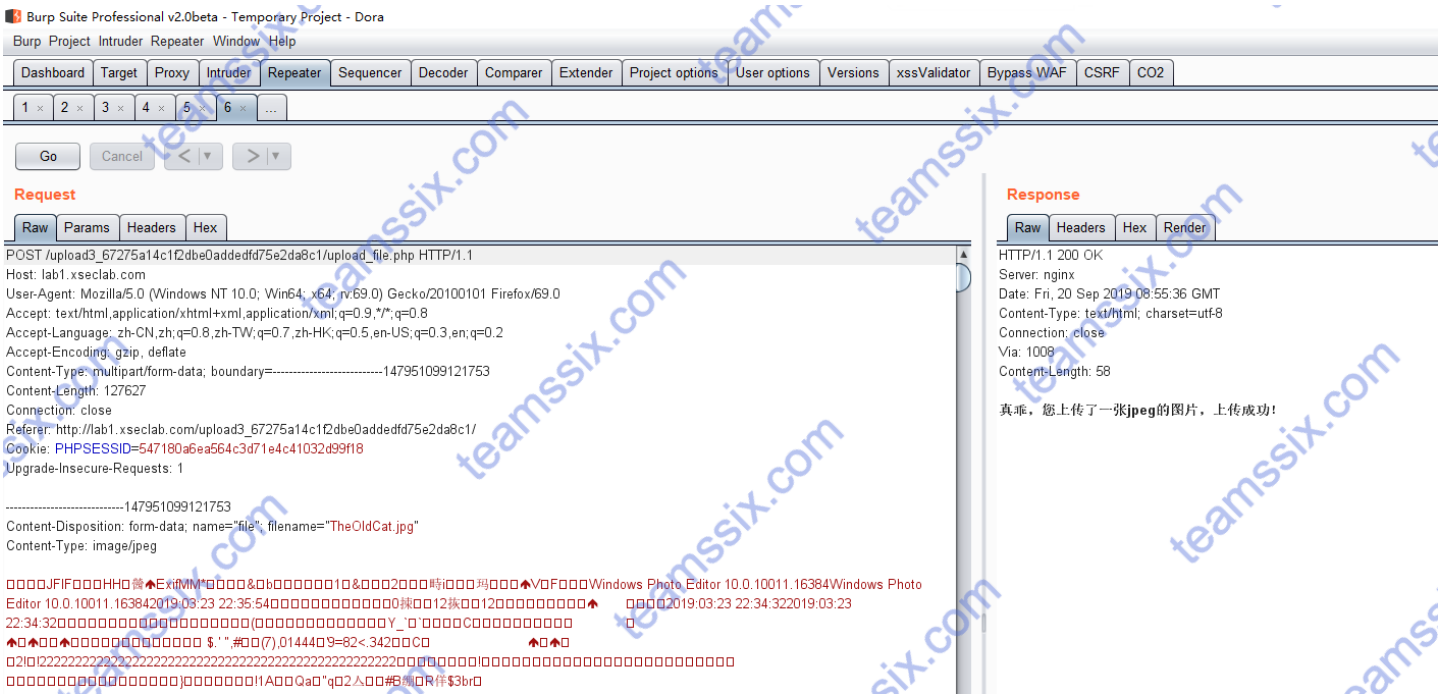


0x04 Web 2

```
flag:76tyuh12OKKytig#$%^&

题目地址：http://lab1.xseclab.com/upload3_67275a14c1f2dbe0addedfd75e2da8c1/

flag格式：key is :

题目来源：http://hackinglab.cn/ShowQues.php?type=upload
```

1、打开题目地址，发现是一个文件上传界面

2、先把Burp挂上，随便上传一个JPG图片试试，并来到Burp重发这个包



3、在Burp中对文件名进行修改，比如在jpg后加上.png或者其他东西，成功看到flag

0x05 soeasy

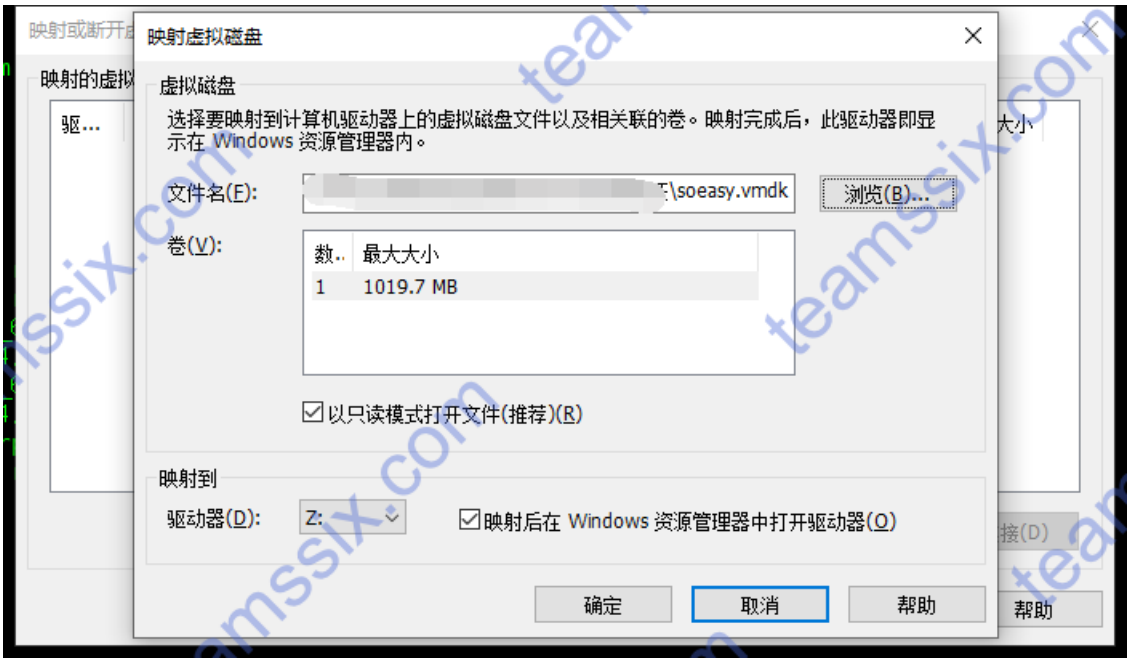
```
flag:HackingLabHdd1b7c2fb3ff3288bff

Hint:在这个文件中找到key就可以通关

flag格式:key:

题目来源：http://hackinglab.cn/ShowQues.php?type=pentest
```

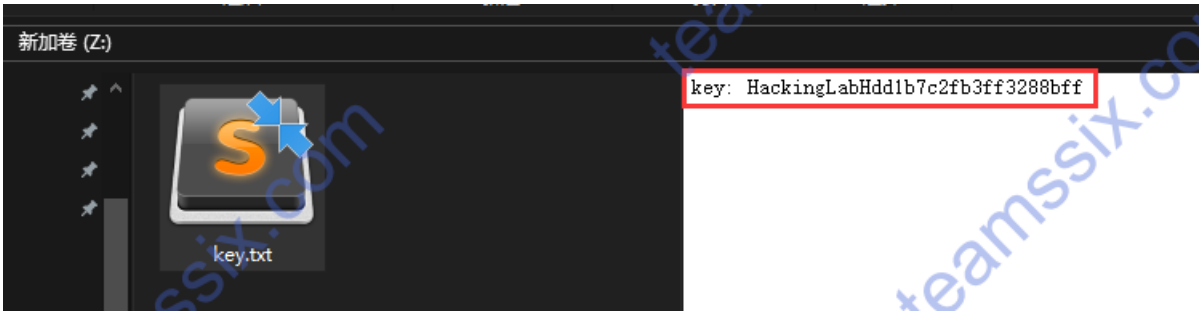
解法一：

1、下载文件后，发现是vmdk文件，利用DeskGenius打开后，发现Key，此为正确答案

解法二：

1、利用Vmware映射虚拟硬盘同样可以打开



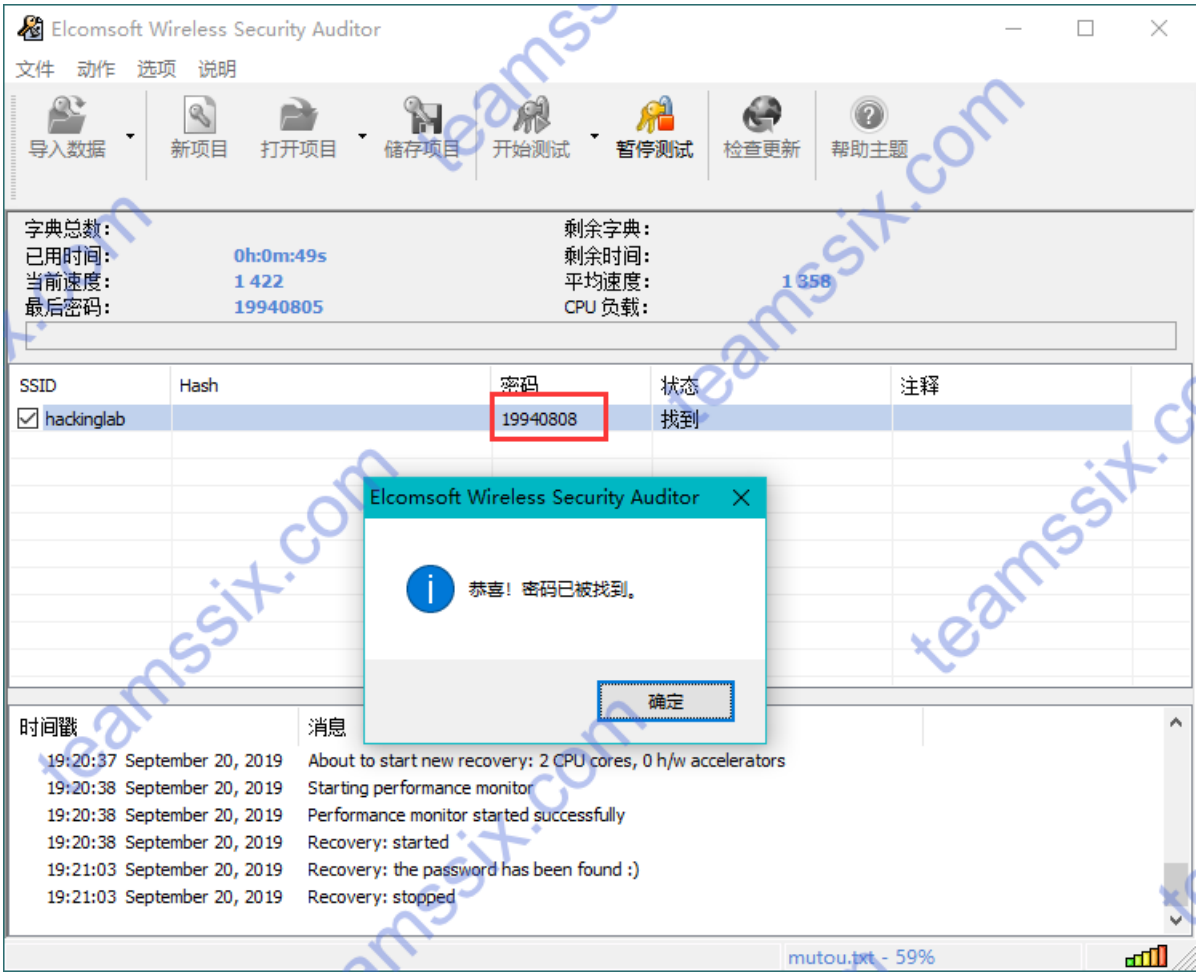


0x06 Crack

```
flag:19940808

Hint:flag就是密码

题目：邻居悄悄把密码改了，你只知道邻居1994年出生的，能找到她的密码吗？

题目来源：http://hackinglab.cn/ShowQues.php?type=decrypt
```

1、下载题目文件，根据题意，需要对WiFi密码破解，而且密码很有可能是邻居的生日，因此我们利用工具生成字典。

2、接下来利用ewsa进行破解，可以看到破解后的密码



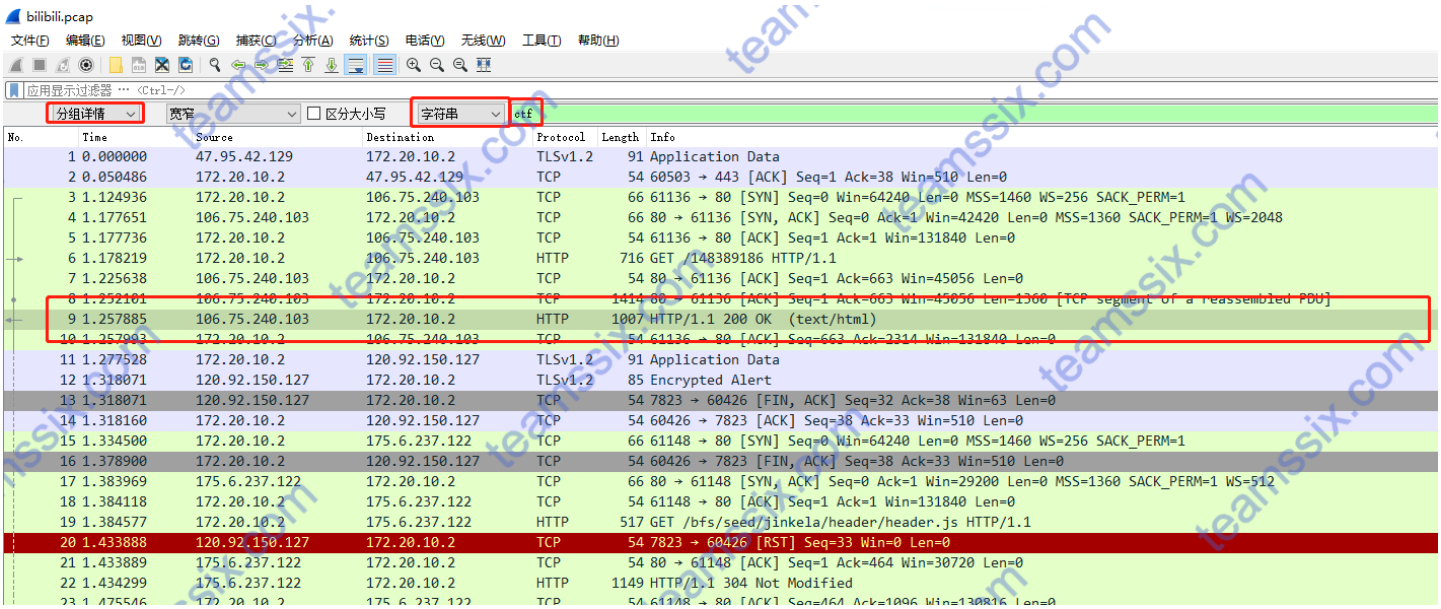
这道题目当时实验室有人用kali做的，kali下的工具感觉破解速度更快。

0x07 BiliBili

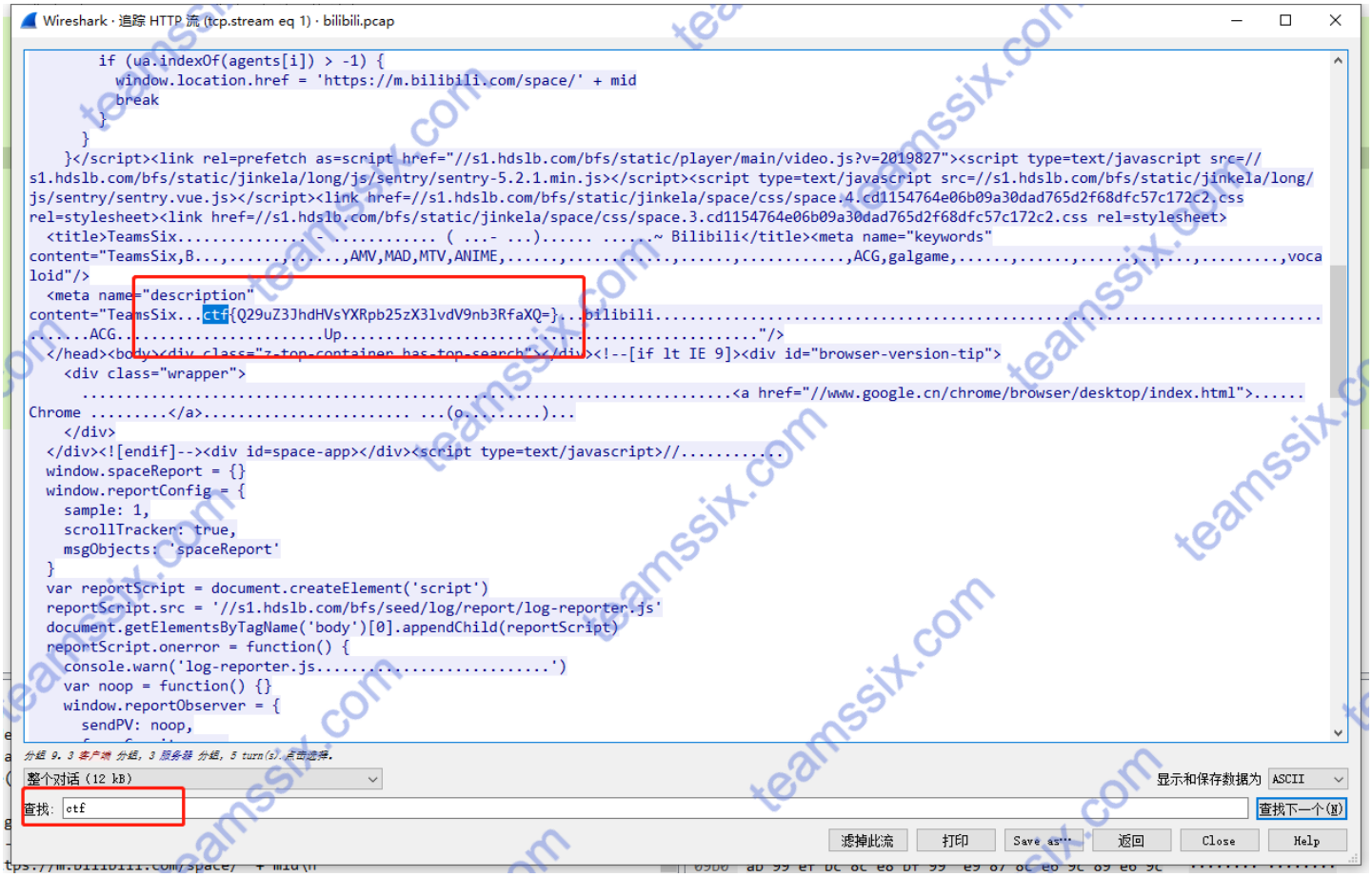
```
flag:Congratulations_you_got_it

题目：bilibili

flag格式：ctf{}
```

解法一：

1、使用Wireshark打开数据包，直接搜索ctf



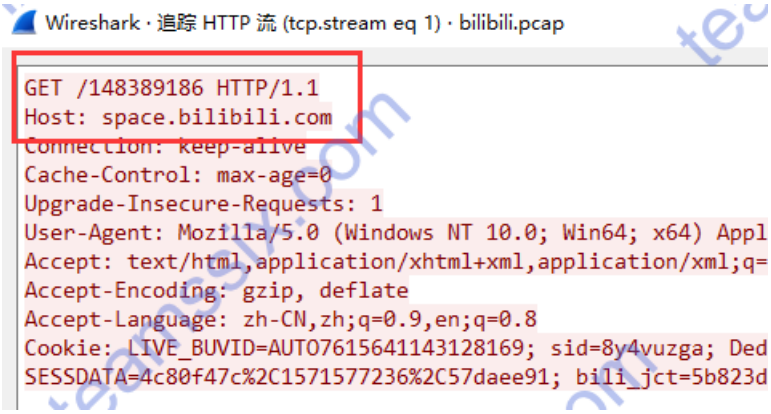
2、找到标识的那一行右击进行追踪对应的协议，比如这条是http协议就追踪http协议，之后再次查找ctf



3、发现ctf括号后的内容为base64加密，解密即可得到flag

解法二：

1、和解法一一样，对数据包进行追踪http流，不难看出这是访问space.bilibili.com/148389186的一个数据包



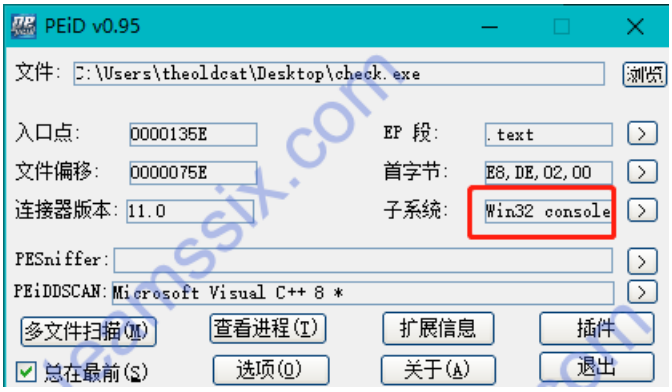
2、打开这个网址，同样可以看到被base64加密的flag

另外打个小广告，上面这个是我的bilibili号（TeamsSix），欢迎大家关注，嘿嘿

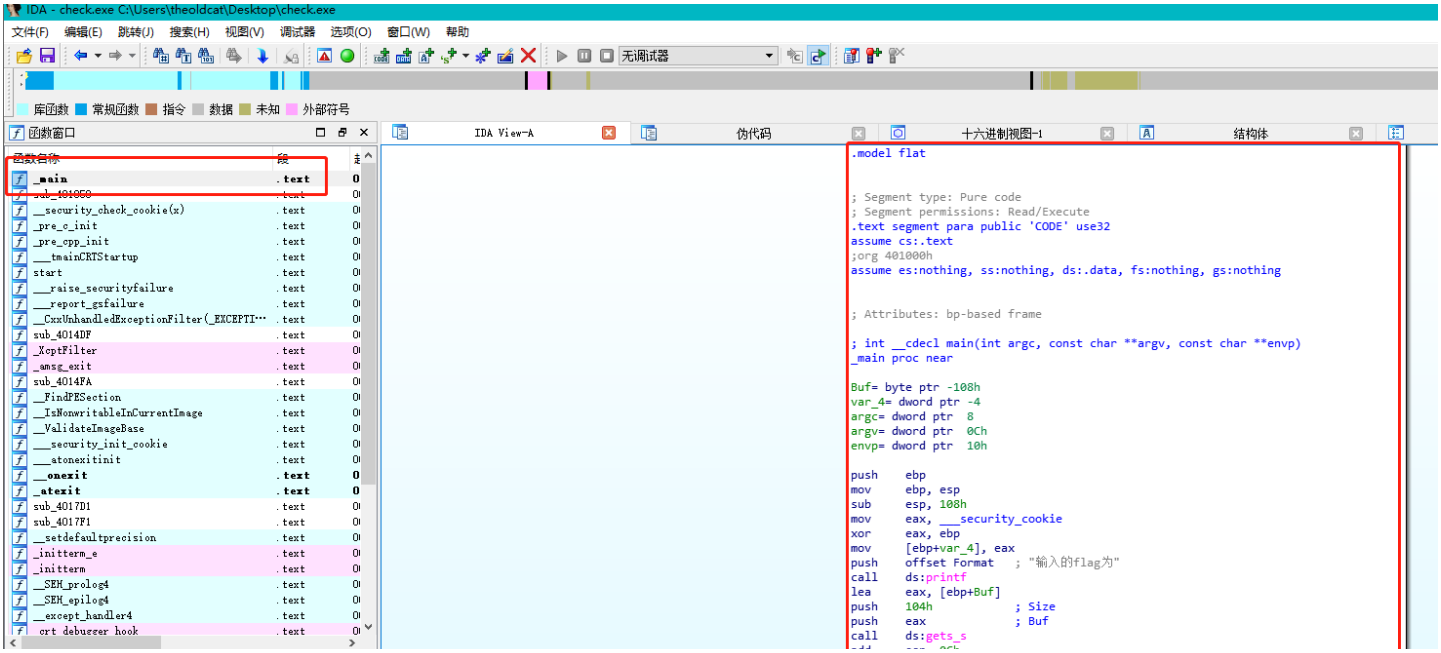
0x08 Check

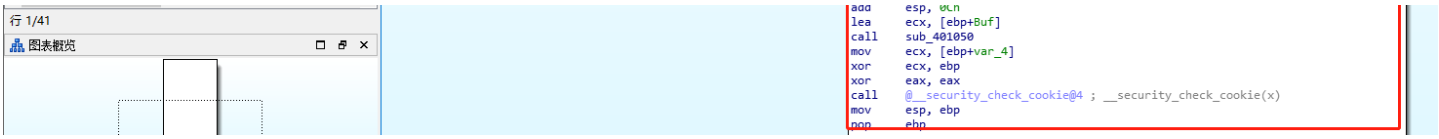
```
flag:sAdf_fDfkl_Fdf

题目：简单的逆向

flag格式：flag{}

题目来源：https://www.cnblogs.com/QKSword/p/9095242.html
```

1、下载文件，发现是exe文件，放到PEiD里看看有没有壳以及是什么语言编写的，如果有壳需要先脱壳。

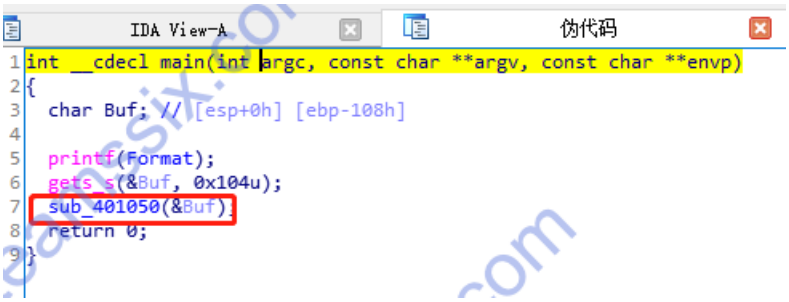


2、可以看到使用的C语言写的，同时是32位，因此使用IDA32位打开，之后找到main函数





3、按F5查看伪代码，并点击sub_401050子函数



4、不难看出下列是一个10进制到ASCII码的转换

```
int __thiscall sub_401050(_BYTE *this)
{
  int result; // eax

  if ( *this != 102 )
    goto LABEL_22;
  if ( this[1] == 108
    && this[2] == 97
    && this[3] == 103
    && this[4] == 123
    && this[5] == 115
    && this[6] == 65
    && this[7] == 100
    && this[8] == 102
    && this[9] == 95
    && this[10] == 102
    && this[11] == 68
    && this[12] == 102
    && this[13] == 107
    && this[14] == 108
    && this[15] == 95
    && this[16] == 70
    && this[17] == 100
    && this[18] == 102
    && this[19] == 125 )
  {
    printf("yes,this is a flag");
    getchar();
LABEL_22:
    result = 0;
  }
  return result;
}
```

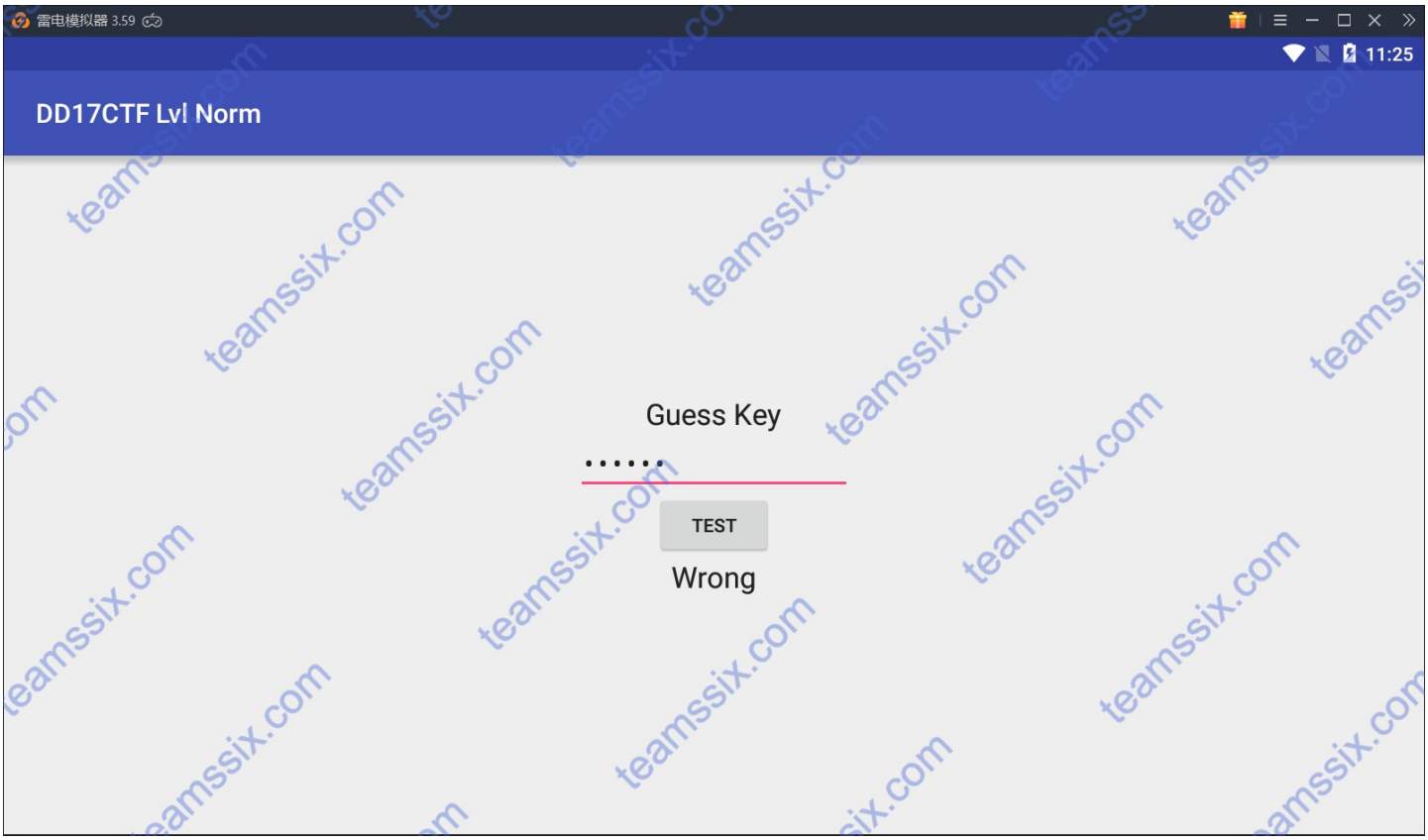
5、利用在线网站转换即可获得flag，网站地址：http://ctf.ssleye.com/jinzhi.html

十六进制	66 6c 61 67 7b 73 41 64 66 5f 66 44 66 6b 6c 5f 46 64 66 7d

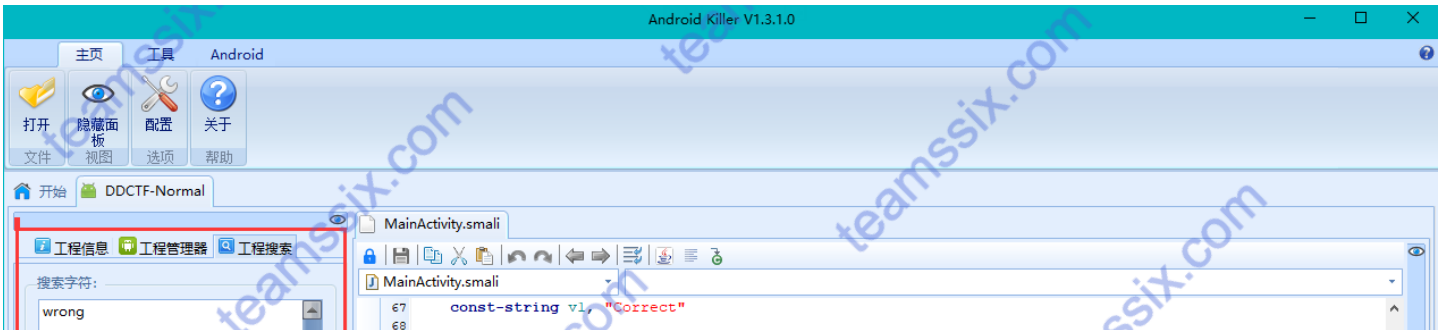
0x09 Android RE

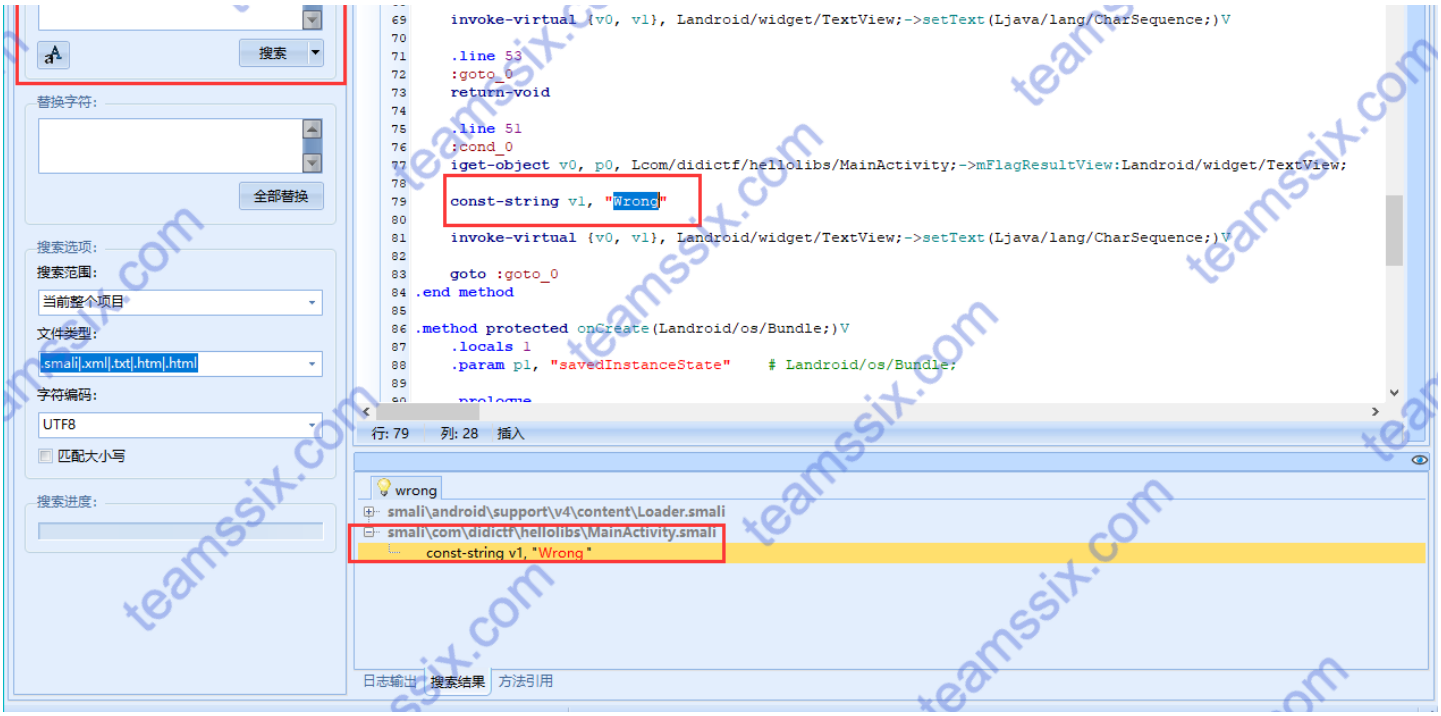
```
flag:DDCTF-397a90a3267641658bbc975326700f4b@didichuxing.com

题目：安卓逆向

flag格式：DDCTF-

Hint:flag中包含chuxing

题目来源：https://xz.aliyun.com/t/1103
```

1、这道题是滴滴出行的一道CTF，下载题目可以看到一个apk文件，先在模拟器中运行看看是个什么东西

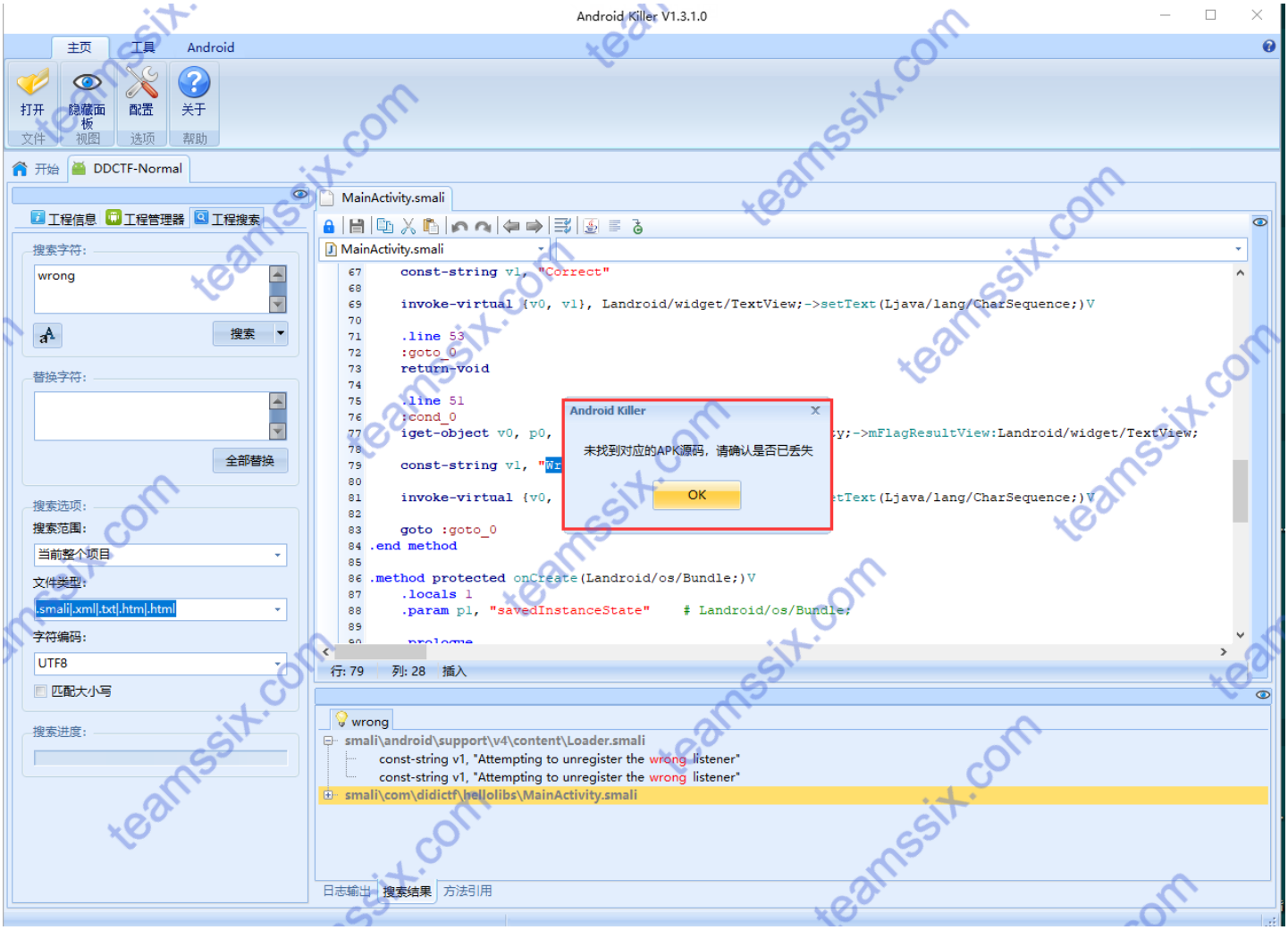


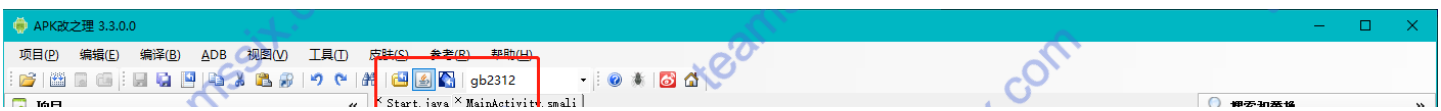
2、功能很简单，一个输入框，输错会提示Wrong，那么利用Android killer给它反编译一下，查找字符“Wrong”

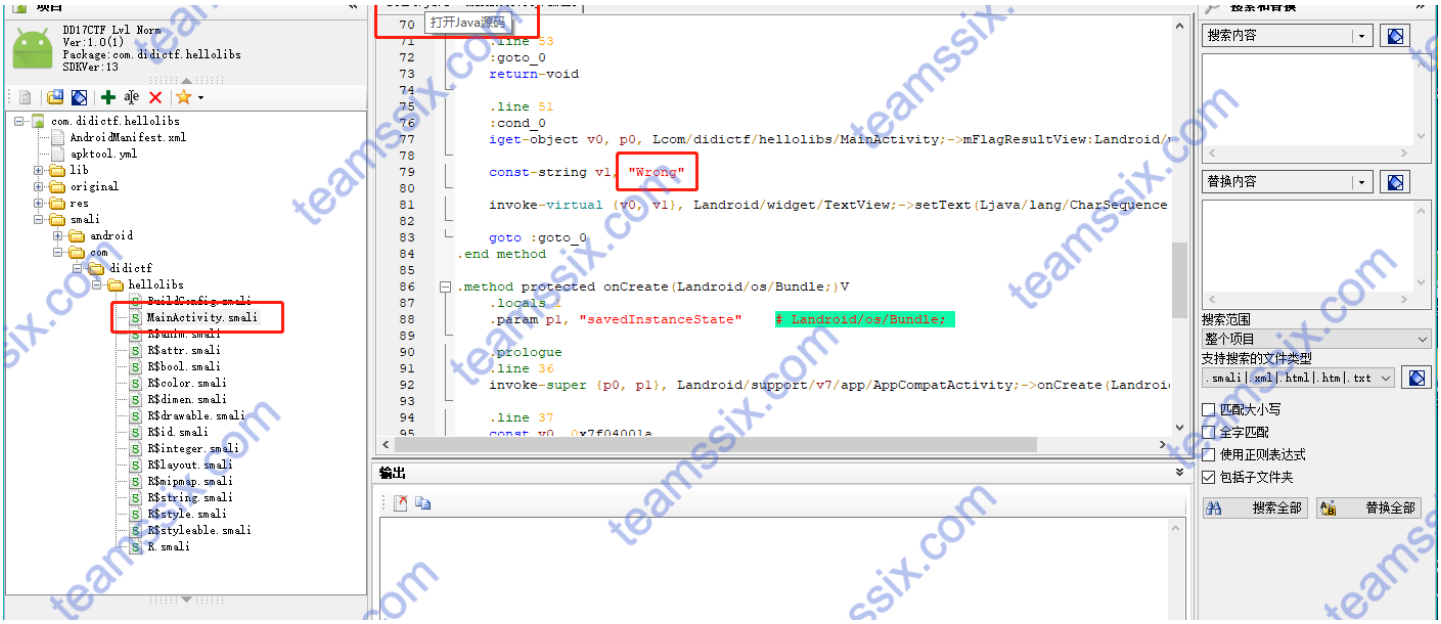




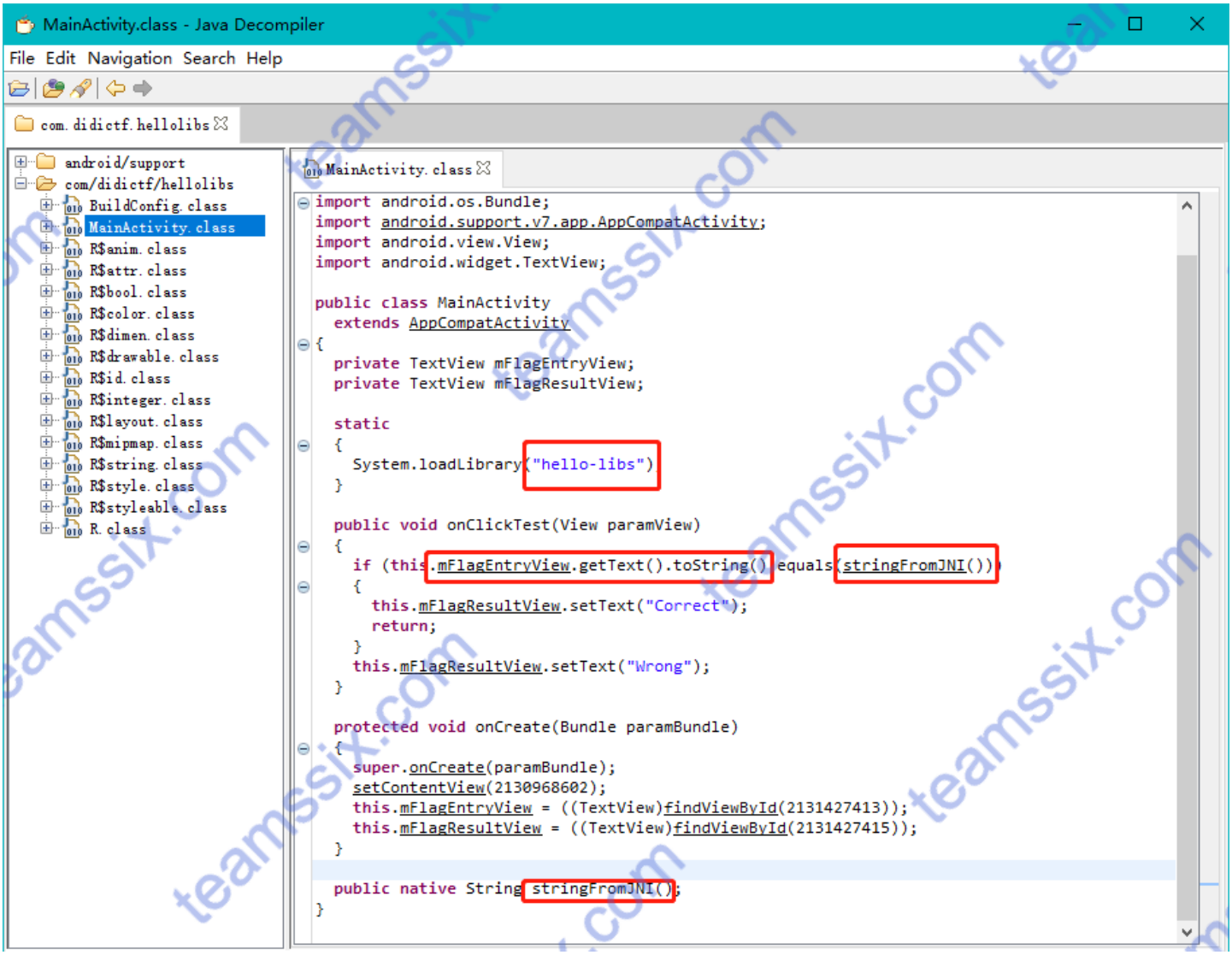
3、可以看到Wrong字符的路径，接下来进行反编译，不过可能由于本身软件的文件，反编译提示未找到对应的APK源码，没关系，换ApkIDE对其进行编译



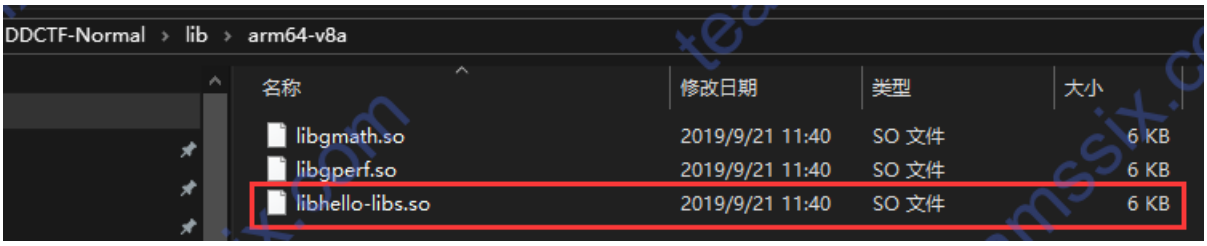




4、等待一段时间后，可以看到对应源码，简单分析就可以知道该代码从hello-libs.so文件加载，并且对mFlagEntryView.getText().toString()函数的内容即我们输入的内容和stringFromJNI()函数的内容做判断，如果一致就Correct，即正确，不一致就返回Wrong，即错误，那么接下来只需要分析stringFromJNI()的内容就行了，因此我们需要知道系统从hello-libs.so文件加载了什么



5、将APK解压，找到hello-libs.so文件，由于现在手机都是用arm64位的CPU（我也不知道是不是的啊，听别人说的），因此我们找到arm64-v8a文件夹下的libhello-libs.so文件，用IDA打开



其中，mips、armeabi、armeabi-v7a和x86都表示CPU的类型。一般的手机或平板都是用arm的cpu。

armeabi 是针对普通的或旧的arm v5 cpu，32位

armeabi-v7a 是针对有浮点运算或高级扩展功能的arm v7 cpu，32位

arm64-v8a 针对64位的

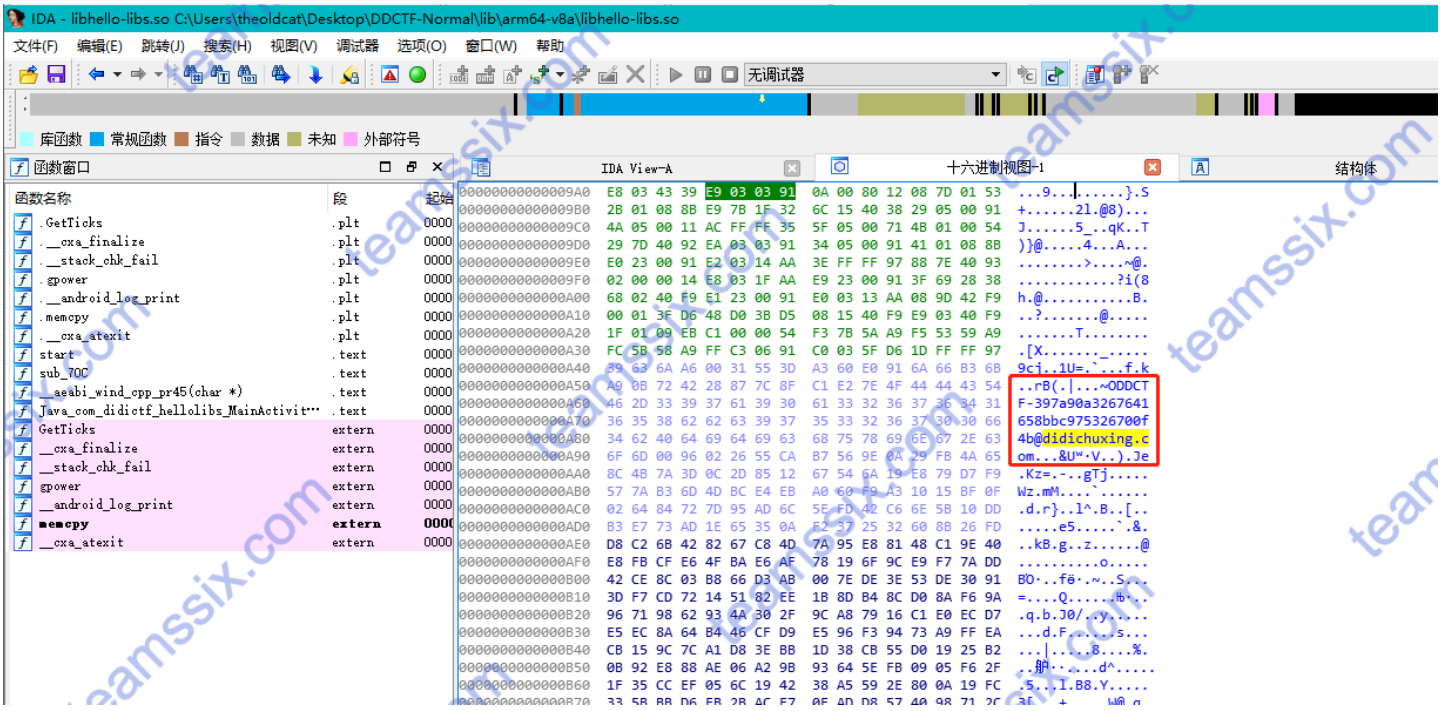
mips 是一种采取精简指令集（RISC）的处理器架构，32位

mips64 64位

x86 IA-32位指令集

x86_64 64位

6、打开IDA后，根据题目提示，Alt +T 查找chuxing



7、成功找到flag（DDCTF-397a90a3267641658bbc975326700f4b@didichuxing.com

）输入到模拟器中看到提示Correct，说明flag正确。

0x10 Easy_dump

```
flag：F0rens1cs_St2rt

题目：Easy_dump

flag格式：LCTF{}

Hint：volatilty了解一下

题目来源：https://www.tr0y.wang/2016/12/16/MiniLCTF/index.html
```

解法一：

1、下载题目文件，提示利用volatilty工具，同时结合文件后缀为vmem（VMWare的虚拟内存文件），因此判断是一个内存取证的题目，关于volatilty的使用可以参考官方手册：https://github.com/volatilityfoundation/volatility/wiki/Command-Reference，废话不多说，先看看镜像信息

```
# volatility -f xp.vmem imageinfo
```

```
root@kali:~/Desktop# volatility -f xp.vmem imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/root/Desktop/xp.vmem)
                      PAE type : PAE
                           DTB : 0xb2a000L
                          KDBG : 0x8054e2e0L
                Number of Processors : 2
           Image Type (Service Pack) : 3
                     KPCR for CPU 0 : 0xffdff000L
                     KPCR for CPU 1 : 0xfc72b000L
                 KUSER_SHARED_DATA : 0xffdf0000L
               Image date and time : 2016-12-02 11:17:55 UTC+0000
         Image local date and time : 2016-12-02 19:17:55 +0800
```

2、可以看到该镜像信息的为WinXPSP2x86，接下来直接扫描查看一些系统文件中有没有flag文件

```
# volatility -f xp.vmem --profile=WinXPSP2x86 filescan | grep flag
```

```
root@kali:~/Desktop# volatility -f xp.vmem --profile=WinXPSP2x86 filescan | grep flag
Volatility Foundation Volatility Framework 2.6
0x0000000005ab74c8      1      0 RW-r-- \Device\HarddiskVolume1\Documents and Settings\Administrator\桌面\flag.txt
0x0000000007782ef8      1      0 RW-rw- \Device\HarddiskVolume1\Documents and Settings\Administrator\Recent\flag.lnk
```

3、将该flag.txt文件dump下来

```
# volatility -f xp.vmem --profile=WinXPSP2x86 dumpfiles -Q 0x0000000005ab74c8 -D ./ -u
```

```
root@kali:~/Desktop# volatility -f xp.vmem --profile=WinXPSP2x86 dumpfiles -Q 0x0000000005ab74c8 -D ./ -u
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x05ab74c8   None   \Device\HarddiskVolume1\Documents and Settings\Administrator\桌面\flag.txt
```

4、直接cat flag文件即可看到flag

```
root@kali:~/Desktop# ls
file.None.0xff876ac0.dat  xp.vmem
root@kali:~/Desktop# cat file.None.0xff876ac0.dat
LCTF{F0rens1cs_St2rt}root@kali:~/Desktop#
```

解法二：

因为该题作者将flag复制到了自己电脑的粘贴板里的，所以直接获取粘贴板的内容也是可以看到flag的，不过谁能想到这种操作[笑哭]

```
# volatility -f xp.vmem clipboard
```

```
root@kali:~/Desktop# volatility -f xp.vmem clipboard
Volatility Foundation Volatility Framework 2.6
Session    WindowStation Format                Handle Object     Data
---------- ------------- ------------- ---------- ---------- ----------------------
         0 WinSta0       0xc009L             0x7e009d 0xe1577c30
         0 WinSta0       CF_UNICODETEXT      0x1c00d1 0xe10ca7c0 LCTF{F0rens1cs_St2rt}
         0 WinSta0       0xc013L             0x1000b3 0xe1260b38
         0 WinSta0       CF_LOCALE           0x16008b 0xe15a08e8
         0 WinSta0       CF_TEXT                0x1 ----------
         0 WinSta0       CF_OEMTEXT             0x1 ----------
```

以上就是本次我为他们准备的CTF的全部内容，大多数都是很基础的题目，平时拿来练练手还是不错的，拓宽一下自己的了解面，发现一些自己以前不知道的东西，如果你也想拿上面的题目来玩玩，在公众号（TeamsSix）回复CTF就可以获取下载地址哦。