# 十三届全国大学生信息大赛+强网杯+DASCTF八月V&N出题 赛-刷题笔记

# 原创

水星Sur ● 于 2020-08-27 13:04:56 发布 ● 1846 ◆ 收藏 2
 分类专栏: CTF Misc Web 文章标签: 信息安全 python 安全
 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/pone2233/article/details/108255940
 版权



CTF 同时被3个专栏收录

20 篇文章 0 订阅

订阅专栏



22 篇文章 0 订阅

订阅专栏



11 篇文章 0 订阅 订阅专栏

菜鸟的自白:

第十三届全国大学生信息安全竞赛

the\_best\_Ctf\_game

电脑被黑

第四届"强网杯"全国网络安全挑战赛

主动

upload

DASCTF 八月浪漫七夕战

双重图格

eeeeeeasyusb

参考文献

菜鸟的自白:

刚开始我还不知道什么是CTF到了大学,有学长带起,慢慢的步入这个信安大世界,我从基础小白,到现在入门小菜鸟,我觉得 学习CTF,可以锻炼自己写脚本,看bug,学渗透,不断的充实自己,这次这3个比赛,真的让我发现,自己还是很菜鸟,需要 继续锻炼,然后各位加油!下载文件猛如虎,一看战绩0-5,太难了。QWQ



## 第十三届全国大学生信息安全竞赛

### the\_best\_Ctf\_game

### 放入winhex里面就能看到flag

0003B0	0C	00	00	00	E0	FF	FF	FF	в0	FF	FF	FF	00	00	00	00	à	999°99	Ŷ
0003C0	66	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	f		
0003D0	01	00	00	00	00	00	00	00	0C	00	00	00	E0	FF	FF	FF			àyyy
0003E0	в0	$\mathbf{F}\mathbf{F}$	$\mathbf{FF}$	$\mathbf{FF}$	00	00	00	00	6C	00	00	00	00	00	00	00	°ŸŸŸ	1	
0003F0	00	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00			
000400	0C	00	00	00	E0	$\mathbf{F}\mathbf{F}$	$\mathbf{FF}$	FF	B0	$\mathbf{F}\mathbf{F}$	FF	$\mathbf{FF}$	00	00	00	00	à	ŸŸŸ°ŸŸ	Ÿ
000410	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	a		
000420	01	00	00	00	00	00	00	00	0C	00	00	00	E0	FF	FF	FF			àÿÿÿ
000430	B0	$\mathbf{F}\mathbf{F}$	$\mathbf{FF}$	FF	00	00	00	00	67	00	00	00	00	00	00	00	°ŸŸŸ	g	
000440	00	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00			
000450	0C	00	00	00	E0	FF	$\mathbf{F}\mathbf{F}$	FF	B0	FF	FF	$\mathbf{FF}$	00	00	00	00	à	YYY <sup>°</sup> YY?	Ŷ
000460	7B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	(		
000470	01	00	00	00	00	00	00	00	0C	00	00	00	E0	FF	FF	FF			àyyy
000480	в0	FF	$\mathbf{FF}$	FF	00	00	00	00	36	00	00	00	00	00	00	00	°yyy	6	
000490	00	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00			
0004A0	0C	00	00	00	E0	FF	FF	FF	в0	FF	FF	FF	00	00	00	00	à	999°99	Ŷ
0004B0	35	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	5		
0004C0	01	00	00	00	00	00	00	00	0C	00	00	00	E0	FF	FF	FF			àyyy
0004D0	в0	$\mathbf{F}\mathbf{F}$	$\mathbf{FF}$	$\mathbf{FF}$	00	00	00	00	65	00	00	00	00	00	00	00	°YYY	e	
0004E0	00	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00			
0004F0	0C	00	00	00	E0	$\mathbf{F}\mathbf{F}$	$\mathbf{F}\mathbf{F}$	FF	в0	$\mathbf{F}\mathbf{F}$	FF	$\mathbf{F}\mathbf{F}$	00	00	00	00	à	YYY°YY:	Ŷ
000500	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0		
000510	01	00	00	00	00	00	00	00	0C	00	00	00	E0	$\mathbf{FF}$	$\mathbf{F}\mathbf{F}$	FF			àÿÿÿ
000520	B0	$\mathbf{F}\mathbf{F}$	$\mathbf{FF}$	$\mathbf{FF}$	00	00	00	00	32	00	00	00	00	00	00	00	°ŸŸŸ	2	
000530	00	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00			
000540	0C	00	00	00	E0	FF	$\mathbf{FF}$	FF	B0	FF	FF	$\mathbf{FF}$	00	00	00	00	à	ŸŸŸ°ŸŸ	Ÿ
000550	66	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	f		
000560	01	00	00	00	00	00	00	00	0C	00	00	00	E0	$\mathbf{FF}$	$\mathbf{F}\mathbf{F}$	FF			àyyy
000570	в0	$\mathbf{F}\mathbf{F}$	$\mathbf{FF}$	$\mathbf{FF}$	00	00	00	00	32	00	00	00	00	00	00	00	°ŸŸŸ	2	
000580	00	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00			
000590	0C	00	00	00	E0	FF	FF	FF	в0	FF	FF	FF	00	00	00	00	à	YYY°YY	Υ
0005A0	36	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	6		
0005B0	01	00	00	00	00	00	00	00	0C	00	00	00	E0	FF	FF	FF			àyyy
0005c0	в0	FF	FF	FF	00	00	00	00	2D	00	00	00	00	00	00	00	°YYY	-	
0005D0	00	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00			
0005E0	0C	00	00	00	ЕÛ	FF	FF	FF	в0	FF	FF	FF	00	00	00	00	à	999°99	Ŷ
0005F0	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0 og	sdn ne	t/none22
000600	01	00	00	00	00	00	00	00	0C	00	00	00	E0	FF	FF	FF			àyyy

#### 把他复制出来,删除多余的东西,就有了flag了

flag{65e02f26-0d6e-463f-bc63-2df733e47fbe}

电脑被黑

放入取证大师中找到被删除文件



打开一看是某种加密

☐ flag.txt - 记事本 文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
D\*□? / b□1N笾霅f\$\窝鲎M□jA 飷Z~[炝鹪a□□E敫

然后使用binwalk分离一下发现了3样子东西



```
发现demo里面是一个文件加密程序,我们反编译一下
         int __cdecl main(int argc, const char **argv, const char **envp)
         {
           int result; // eax
           char v4; // [rsp+1Dh] [rbp-13h]
           char v5; // [rsp+1Eh] [rbp-12h]
           char v6; // [rsp+1Fh] [rbp-11h]
           FILE *v7; // [rsp+20h] [rbp-10h]
           FILE *stream; // [rsp+28h] [rbp-8h]
           v4 = 34;
           v5 = 0;
           v7 = fopen(argv[1], "rb");
           if (v7)
           {
             stream = fopen(argv[1], "rb+");
             if ( stream )
             {
               while (1)
               {
                 v6 = fgetc(v7);
                 if ( v6 == -1 )
                  break;
                 fputc(v4 ^ (v5 + v6), stream);
                                                   1 重点算法
                 v4 += 34;
                 v5 = (v5 + 2) \& 0xF;
               }
               fclose(v7);
               fclose(stream);
               result = 0;
             }
             else
             {
               printf("cannot open file", "rb+", argv);
               result = 0;
             }
           }
```

```
#include <iostream>
#include <stdio.h>
using namespace std;
int main(int argc, char *argv[]) {
 FILE *stream; // [rsp+28h] [rbp-8h]
 v4 = 34;
 v7 = fopen("flag.txt", "rb");
  stream = fopen("flag.txt", "rb+");
  if ( stream )
   fclose(stream);
   printf("cannot open file", "rb+", argv);
   result = 0;
```

### 就得到了flag

flag{e5d7c4ed-b8f6-4417-8317-b809fc26c047}

## 第四届"强网杯"全国网络安全挑战赛

### 主动

打开一看发现了system,那就说明可以执行linux代码指令

我们先使用一下Is查看一下又什么文件然后发现了flag.php

```
← → C ① 不安全 | 39.96.23.228:10002/?ip=%0a%20ls
```

https://blog.csdn.net/pone2233

然后试一下使用cat flag.php 不行,给拦截了,那我就换成cat \*.php 就成功了

← → C ① 不安全 | view-source:39.96.23.228:10002/?ip=%0a%20cat%20\*.php

```
<code><span style="color: #000000">
    1
                     <span style="color: #0000BB">&lt;?php<br />highlight_file</span><span style="color: #007700">(</span</pre>
     2
                     #0000BB">preg_match</span><span style="color: #007700">(</span><span style="color: #DD0000">"/flag/i"</spa
                      <span style="color: #DD0000">"ip"</span><span style="color: #007700">]))<br />{<br />&nbsp;&nbsp;&nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp;%nbsp
                    style="color: #0000BB">system</span><span style="color: #007700">(</span><span style="color: #DD0000">"pir
#0000BB">ip</span><span style="color: #007700">]</span><span style="color: #DD0000">"</span><span style="color: #DD0000">"</span></span style="color: #DD0000">"</span style="color: #DD0000""</span style="color: #DD0000">"</span style="color: #DD0000""</span style="color: #DD0000"">"</span style="color: #DD0000""</span style="color: #DD0000"">"</span style="color: #DD0000""</span style="color: #DD0000""">"<
                   </code><?php
$flag = "flag{I_like_qwb_web}"; <?php
     3
     4
                    highlight file("index.php");
     5
    6
                    if(preg_match("/flag/i", $_GET["ip"]))
    7
                     {
    8
                                               die("no flag");
    9
 10 }
 11
                    system("ping -c 3 $_GET[ip]");
 12
 13
                    ?>
14
 15
 16
```

https://blog.csdn.net/pone2233

ip=%0a cat \*.php

### upload

下载文件发现是文件分析题目

🌆 upload.zip

文件(F) 命令(C) 工具(S)	) 收藏夹(O) 选项(N)	帮助(H)										
添加 解压到 测		重找	▲	信息	(3)	<b>唐</b> 注释	自解压格	武				
in the second s												
名称	大小	压缩后大小	类型			修改时间		CRC32				
J			文件夹									
🛅 data.pcapng	36,416	33,287	Wiresha	ark captu	re file	2020/8/14	22:43	10E2A7A4				

\_

# 老规矩追溯一下tcp然后发现了一个图片

2006100772460252102270066002
Content-Disposition: form-data; name= file; filename= stegnide.jpg
Content-Type: image/jpeg
JFIFC.
······}······}······!IA·.Qd. q.2····#B···K·.>30Г.
<pre>%&amp; ()*456789:CDEFGHIJSTUVWXYZcdefghiJstuvwxyz</pre>
wl1AQ.aq."2B #3Rbr.
.\$4.
%&'()*56789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz
QA.Pi.R.ZC@R.hQEQEQIE4.j).ZSY3p8
F.@SH.i)(.
((A](6.,({Rb.,E.T.,U,b,(
$A_{1} = A_{2} = A_{1} = A_{2} = A_{2$
$\frac{1}{2}$ dn $\frac{1}{2}$ n $\frac{1}{2}$
D D b b b b b b b b b b b b b b b b b b
.ZDM.EXV.rW[.y.hr]U.{8}.6S)V.QHj:ZZ.AEFJR*3@
11.@.4.N".x.B.:H)qMjv.#b,%1almC(.'.ugbj:% .4(4R.E.
%R).SMQE
TuiiY3J.)R4.9p.k@.R.U(.F).b.G4E=jB0.*.#8.
FNj.ucI.YHGZ."y{.eE.ePvd1f5.
9.[.:.3nUS.3WH5.e.I.M.H{z.?\$8f9[.y# by.I6h.;.0}
9@'.k.k0xs\m.j:Uig0.h.X.fA.N7S
·B. ·a)····7tsc.··i··
F. iF: VHi&Y FV6
$ \begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 $
HØ 2.2 2 C/2 H H2 L HAR HØ 1 tops(c) 3 ###

### 导出来还挺骚





重点来了就是他的图片名字,就是一个隐写名字

-----29061997724503521033788658882 Content-Disposition: form-data; name="file"; filename="steghide.jpg" Content-Type: image/jpeg

弱密码试试就成功了密码 123456

E:\>cd E:\桌面\CTF工具包\[隐写工具]steghide

E:\桌面\CTF工具包\[隐写工具]steghide>steghide.exe extract -sf E:\桌面\8-22\data\1.jpg Enter passphrase: steghide: could not extract any data with that passphrase!

stegnide. Could not extract any data with that passphrase:

E:\桌面\CTF工具包\[隐写工具]steghide>steghide.exe extract -sf E:\桌面\8-22\data\1.jpg Enter passphrase: wrote extracted data to "flag.txt".

🥘 flag.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(

flag{te11\_me\_y0u\_like\_it}

还好手快,抢了题目,到这里为止发现了这个就是PWN杯!!!!太欺负人了



### DASCTF 八月浪漫七夕战

### 双重图格

下载文件,我们首先检查一下xls,里面是否又夹带 里里外外看了一次,就发现了一个东西

# Insert:OFF5ET28354h

# 就看的很清楚了 Insert:OFF5ET 28354h 这个词汇翻译一下,就知道位移,在28345H 这个为止上面需要位移。

起始页 test cipher.png×		< > ▽ 工作区	
▼ 编辑方式: 十六进制(H) > 运行脚本 > 运行模板: PNG.bt > ▷		▶ 打开的文件	
<u>0123456789ABCDEF</u>	0123456789ABCDEF	□ cipher.png E:\桌面\8	
5:03E0h: C6 32 D2 18 63 8C B1 8C 34 C6 18 63 2C 23 8D 31	#20.c@±@4#.c,#.1	PNG bt 1 C·\Users\/	
5:03F0h: C6 18 CB 48 63 8C 31 C6 32 D2 18 63 8C B1 8C 34	E.EHCCE1E2O.CCE±CE4	DNG bt:2 Cillions/	
5:0400h: C6 18 63 2C 23 8D 31 C6 18 CB 48 63 8C 31 C6 58	E.C, #.1E.EHCCE1EX	DNC ht2 Cillians	
5:0410h: 46 1A 63 8C 31 96 91 C6 18 63 8C 65 A4 31 C6 18	F. ccl - E. cce alk	PING.DL:3 C:\Users\/	
5:0420h: 63 19 69 8C 31 C6 58 46 1A 63 8C 31 96 91 C6 18	C.1ClEXF.CCl-'E.	test E:\杲囬\8-	
5:0430h: 63 8C 65 A4 31 C6 18 63 19 69 8C 31 C6 58 46 1A	cœe¤1æ.c.1œ1æxF.	📂 收藏的文件	
5:0440h: 63 8C 31 96 91 C6 18 63 8C 65 A4 31 C6 18 63 19	CEI-'E.CEe¤IE.C.	🦻 最近的文件	
5:0450h: 69 8C 31 C6 18 CB 48 63 8C 31 C6 32 D2 18 63 8C	1CLR.EHCCLR2O.CC	□ Octobd.txt E:\桌面\8-	
5:0460n: BI 8C 34 C6 18 63 2C 23 8D 31 C6 18 CB 48 63 8C	tuste.c, #.1R.EHCu	isteahide.php E:\桌面\8-	
5:0470n: 31 C6 32 D2 18 63 8C B1 8C 34 C6 18 63 2C 23 8D	IRZO.CUETUAR.C,F.	2 ipg F\桌面\8-	
5:0480h: 31 C6 18 CB 48 63 8C 31 C6 32 D2 18 63 8C B1 8C	An - A in - inin	IPG bt Cillions	
5:0490h: 34 C6 18 63 2C 23 8D 31 C6 18 63 19 69 8C 31 C6 5:0480b: 59 46 18 63 9C 31 66 61 66 19 63 9C 65 84 31 66	4E.C, #.IE.C.IEIE	Line Line Line Line Line Line Line Line	
5:04R0h: 10 63 10 60 90 31 66 59 46 18 63 80 31 66 91 66	AF.CELT R.CECMIR	key.jpg	
5:0420h: 18 63 80 65 M 31 C6 18 63 19 69 80 31 C6 58 46	CREATER C IRIEVE		
5:0400h: 10 63 80 31 96 91 06 18 63 80 81 80 34 06 18 63	$c(\mathbb{R}^{1}) = \sum_{k=1}^{N} c(\mathbb{R}^{k}) R_{k} c_{k}$	□ ⊥ □ □ □ 页源自理語	
5:04E0h: 2C 23 8D 31 C6 18 CB 48 63 8C 31 E6 DF E1 FF 03	# 18 EHc@1#8áÿ	变量	
5:04F0b: 99 2F 15 4E D5 A1 1B D9 00 00 00 49 45 4E 44	$\mathbf{v}' \in \mathbf{N}^{\circ}$	名称	值 开始 大小 颜色
5:0500h: AR 42 60 82 66 63 54 4C 00 00 00 01 00 00 02 61		> struct PNG_CHUNK chunk	[7] IDAT (Critic 18091b 800Ch Eq: B
5:0510h: 00 00 02 0A 00 00 00 10 00 00 1C 00 01 00 5F		struct DNC_CUUNK shunk	[9] IDAT (Critic 2000Db 800Cb For D
5:0520h: 00 00 5D E5 6B 03 00 00 80 04 66 64 41 54 00 00	]åk€.fdAT	Struct PING_CHOINK chulk	[0] IDAT (CHIC 2009DH 600CH FG. B
5:0530h: 00 02 78 DA EC 7D 79 9C 14 C5 F9 FE F3 56 75 F7	xÚì}yœ.ÅùþóVu÷	struct PNG_CHUNK chunk	[9] IDAT (Critic 280A9h 2A7h Fg: B
5:0540h: CC CE 1E 2C EB 8A 88 88 06 09 22 A2 41 44 82 8A	ÌÎ.,ëŠ^^"¢AD,Š	struct PNG_CHUNK chunk	[10] fcTL (Ancill 28350h 26h Fg: B
5:0550h: 88 88 17 F1 BE C5 DB 78 8B B7 51 63 12 FD 19 43	^^.ñ¾ÅÛx‹ Qc.ý.C	> struct PNG_CHUNK chunk	[11] fdAT (Ancill 28376h 8010h Fa: B
5:0560h: 8C 31 7E 8D F1 8A F7 7D E2 85 4A BC 45 44 62 50	Œ1∼.ñŠ÷}âJ¾EDbP	> struct PNG_CHUNK chunk	[12] fdAT (Ancill 30386h 8010h Eq. B
5:0570h: D1 78 06 B9 44 04 E4 86 5D D8 63 A6 AB EA FD FD	Ñx.¹D.ä†]Øc¦«êýý	struct DNC_CITUNK should	[12] fart (recent 2000ch 0010h Fg. D
5:0580h: 51 BD 3D B3 B3 B3 27 0B 5E F5 7C 5A EC ED E9 BB	Q <sup>1</sup> 2= <sup>3 3 3 '</sup> .^õ Zìíé≫	> struct PING_CHOINK chunk	[15] IdAT (Ancill 56596n 6010n Fg: D
5:0590h: AB EA A9 F7 26 66 86 83 83 83 83 83 83 43 13 08	≪ê©÷&f†fffffC	struct PNG_CHUNK chunk	[14] fdAT (Ancill 403A6h 8010h Fg: B
5:05A0h: F7 0A D6 15 4A B9 77 E0 E0 E0 E0 E0 38 D2 A1 31	÷.Ö.J¹wààààà8Ò;1	struct PNG_CHUNK chunk	[15] fdAT (Ancill 483B6h 8010h Fg: B
5:05B0h: 35 3E FF 3C 0E 39 04 45 45 28 2A C2 16 5B E0 97	5>ÿ<.9.EE(*Ã.[à-	> struct PNG_CHUNK chunk	[16] fdAT (Ancill 503C6h 132h Fa: B
5:05C0h: BF C4 01 07 E0 8F 7F C4 D2 A5 EE F5 38 38 38 38	88888îf¥lõ8888	> struct Diagon HUNK chunk	LL/L LENU) (( mu 504F8h Ch Fa: B
5:05D0h: FC 08 40 4E D7 DA 6E 7C F1 05 EE B9 07 0F 3F DC	ü.@N×Un ñ.1º?U	at the Church Street	
5:05E0h: 2C 17 26 93 38 FA 68 5C 78 21 B6 DF DE BD AD F5	,.&"8ùh\x!¶ß₽≒=ő	struct PNG_CHONK chunk	[10] 50504n Un •9 B
5:05F0h: 82 97 5F 46 BF 7E D8 72 4B F7 26 7E 0C C8 64 50	,F;~ØrK÷&~.EdP		· · · · · · · · · · · · · · · · · · ·
5:0600h: 5D 8D CA 4A F7 26 IC 1C 47 FE CO 31 63 06 8E 3F	J.EJ÷&GpAIC.Z?	▲ チャ 函数 👘 安里	
输出			
サ/「##F /C/11/ A douting to the A D			

模板执行成功。

我们	门首先,	让如0	10里	面看	,发	现一	个没	有访	别的	的东西	<b>§</b> , ₹	<b>戈猜</b> 》	则应i	亥是法	这个法	东西	需要信	立移,应该是插入这个位置
文	件( <u>F</u> ) 编	辑( <u>E</u> )	搜索	( <u>S</u> )	视图	(⊻)	格式	( <u>O</u> )	脚本	·( <u>I)</u> ;	模板(	<u>L)</u>	Τ昗(	<u>I)</u> 1	ĝ□( <u>\</u>	<u>N)</u>	帮助( <u> </u>	<u>–</u> )
I I	` ~ 📂	~ 🔒 🚦	1	1 🖿	1	i %		<b>-</b> 1	5	L R	ړ 🔎	р 🍃	• >	I A	м 📐		<b></b>	Π 📰 🔤 📄 🐢 🔽 🗽 🐼 🐉
起	始页	test	cip	oher.	. png	×												
Ŧ	编辑方式	式: 十六	、进制	(H) `	√ ž	s行脚	本 ~	⁄ 运	行模	扳: F	PNG.Ł	ot 🗸	⊳					
		Ò		2			5	Ģ	7		9	Ą	B	Ċ	Ď	E		0123456789ABCDEF
2:	8310h	: 01	40	80	01	00	01	06	00	01	06	00	04	18	00	04	18	.@€
2:	8320h	: 00	10	60	00	10	60	00	40	80	01	40	80	01	40	80	01	``.@€.@€.@€.
2:	8330h	: 00	01	06	00	01	<u>⁄</u> ⁄6	00	04	18	00	04	18	00	10	60	00	
2:	8340h	: 10	60	00	40	80	01	20	61	0D	4A	15	90	26				.`.@€. a.J&=¢»
2:	8350h	: 00	00	00	1A	66	63	54	4C	00	00	00	0В	00	00	02	61	fcTLa
2:	8360h	: 00	00	02	0A	00	00	00	10	00	00	00	1C	00	01	00	5F	
2:	8370h	: 00	00	в0	CA	FB	CC	00	00	80	04	66	64	41	54	00	00	°ÊûÌ€.fdat
2:	8380h	: 00	0C	78	DA	EC	BD	79	9C	<b>A</b> 4	55	75	FF	FF	39	E7	DE	xÚì½yœ¤Uuÿÿ9ç₽
2:	8390h	: A7	AA	D7	Е9	EE	D9	17	60	06	90	4D	59	86	01	86	75	§ª×éîÙ.`MY†.†u
2:	83A0h	: 08	AB	в2	28	06	D4	28	2A	18	8D	89	1A	F5	17	в7	24	.≪²(.Ô(*‰.õ. ·\$
2:	83B0h	: 26	9A	68	4C	62	30	51	BF	EE	5F	25	<b>A</b> 0	22	8A	F2	25	&šhLb0Q;î_% "Šò%
2:	83C0h	: 2E	91	60	8C	41	33	2E	20	<b>A</b> 8	AC	1A	18	(76)	D9	66	EF	.``ŒA3. <sup>~</sup> ¬(vĎfï
2:	83D0h	: E9	9E	DE	AA	EA	79	EE	3D	9F	$\mathbf{DF}$	1F	в7	AA	A7	19	11	éžÞªêyî=Ÿß. ∙ª§
2:	83E0h	: D9	D1	99	FB	7E	F5	6B	5E	ЗD	D5	DD	D5	5D	F5	2C	9F	ÙÑ™û~õk^=ÕÝÕ]õ,Ÿ
2:	83F0h	: 7B	CE	ЗD	E7	73	84	24	32	99	4C	26	93	C9	$\mathbf{FC}$	0A	9A	{Î=çs"\$2™L&``Éü.š
2:	8400h	: DF	82	4C	26	93	C9	64	В2	46	66	32	99	4C	26	93	35	β,L&"Éd²Ff2™L&"5
2:	8410h	: 32	93	C9	64	32	99	AC	91	99	4C	26	93	C9	64	8D	CC	2"Éd2™¬ '™L&"Éd.Ì
2:	8420h	: 64	32	99	4C	26	6B	64	26	93	C9	64	32	59	23	33	99	d2™L&kd&``Éd2Y#3™
<u>-</u> -	04201	- 10	26	02	00	17	00	00	61	22	00	40	DE	00	40	26	02	

Éd²Ff2™L&``52``Éd2 ™\_`™L&``Etps://blog.csdn.net/pone2233

插入之后发现,似乎还是没什么变化,然后在010中发现了

2:8440h: C9 64 B2 46 66 32 99 4C 26 93 35 32 93 C9 64 32 2:8450h: 99 AC 91 99 4C 26 93 C9 64 B2 46 66 32 99 4C 26



https://blog.csdn.net/pone2233

这个fdAT是什么东西然后查一下,找到了APNG,他们说这个是很相似gif然后放入谷歌和火狐里面就可以查看了

变量				
名称	值	开始	大小	
struct PNG_CHUNK chunk[9]	IDAT (Critic	280A9h	2A7h	Fg
struct PNG_CHUNK chunk[10]	fcTL (Ancill	28350h	26h	Fg
struct PNG_CHUNK chunk[11]	fdAT (Ancill	28376h	8010h	Fg
struct PNG_CHUNK chunk[12]	fdAT (Ancill	30386h	8010h	Fg
struct PNG_CHUNK chunk[13]	fdAT (Ancill	38396h	8010h	Fg
struct PNG_CHUNK chunk[14]	fdAT (Ancill	403A6h	8010h	Fg
struct PNG_CHUNK chunk[15]	fdAT (Ancill	483B6h	8010h	Fg
struct PNG_CHUNK chunk[16]	fdAT (Ancill	503C6h	8010h	Fg
struct PNG_CHUNK chunk[17]	fdAT (Ancill	583D6h	8010h	Fg
struct PNG_CHUNK chunk[18]	fdAT (Ancill	603E6h	8010h	Fg
struct PNG_CHUNK chunk[19]	fdAT (Ancill	683F6h	449Eh	Fg
struct PNG CHUNK chunk[20]	fcTL (Ancill	6C894h	26h	Fq
<			<u> </u>	_
f0 函数 变量	https://	//blog.csdn.	net/pone22	33

# 哎呀9让你发现3呢~







手快就行,然后发现这个点点,起初以为是哪个猪圈密码,后门查一下,发现了是DotCode,也是一种二维码,真的是长见识了 大家可以了解一下DotCode|斐泰二维码

然后,这里就卡住了,然后多谢一个大佬,送的破解网址



密文1: U2FsdGVkX1/mLyhDqehTlmxmPoamVfr7h1El3iWRVvuJQodh1HvxMeQ2F8lgHfXzq70N4U/ZcjYtjLbXE8HRmw==

然后接下来就是破解key.jpg 当时尝试了很多jpg的加密都不对,然后这个二维码扫描不出来





### 然后拖入010觉得很奇怪

▼ 编辑方	式: -	十六进	推制()	H) 🗸	- 运行	行脚Z	× 7	」运行	ī模板	: JF	'G.bt						
	Ò				4	5	Ģ				Ą	B	Ċ	Ď			0123456789ABCDEF
0000h:	FF	D8	$\mathbf{F}\mathbf{F}$	E0	00	10	4A	46	49	46	00	01	01	01	01	5E	ÿØÿàJFIF
0010h:	01	5E	00	00	FF	E1	FF	FF	45	78	69	66	00	00	4D	4D	│ .^ÿá <mark>1</mark> 13:
0020h:	00	2A	00	00	00	08	00	01	01	12	00	03	00	00	00	01	*
0030h:	00	01	00	00	00	00	00	1A	00	06	01	03	00	03	00	00	
0040h:	00	01	00	06	00	00	01	1A	00	05	00	00	00	01	00	00	
0050h:	00	68	01	1B	00	05	00	00	00	01	00	00	00	70	01	28	.hp. (
0060h:	00	03	00	00	00	01	00	02	00	00	02	01	00	04	00	00	
0070h:	00	01	00	00	00	78	02	02	00	04	00	00	00	01	00	01	x
0080h:	DE	В8	00	00	00	00	00	00	00	48	00	00	00	01	00	00	₽,H
0090h:	00	48	00	00	00	01	FF	D8	FF	ЕO	00	10	4A	46	49	46	.HÿØÿàJFIF
00A0h:	00	01	01	01	01	5E	01	5E	00	00	FF	E1	00	22	45	78	^.^ÿá."Ex 🧿 🤈 🔨
00B0h:	69	66	00	00	4D	4D	00	2A	00	00	00	08	00	01	01	12	ifMM.*
00C0h:	00	03	00	00	00	01	00	01	00	00	00	00	00	00	FF	DB	ÿû
00D0h:	00	43	00	02	01	01	02	01	01	02	02	02	02	02	02	02	.C
00E0h:	02	03	05	03	03	03	03	03	06	04	04	03	05	07	06	07	
00F0h:	07	07	06	07	07	80	09	0B	09	80	80	0A	08	07	07	0A	
0100h:	0D	0A	0A	0B	0C	0C	0C	0C	07	09	0E	0F	0D	0C	0E	0B	
0110h:	0C	0C	0C	FF	DB	00	43	01	02	02	02	03	03	03	06	03	ÿÛ.C
0120h:	03	06	0C	80	07	80	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	
0130h:	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	
0140h:	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	
0150h:	0C	0C	0C	0C	0C	0C	0C	0C	$\mathbf{F}\mathbf{F}$	C0	00	11	08	02	58	01	ÿÀX.
0160h:	E8	03	01	22	00	02	11	01	03		01	FF	C4	00	1F	00	è"ÿÄ
0170h:	00	01	05	01	01	01	01	01	01	00	00	00	00	00	00	00	
0180h:	00	01	02	03	04	05	06	07	80	09	0A	0в	FF	C4	00	в5	ÿÄ.μ
0190h:	10	00	02	01	03	03	02	04	03	05	05	04	04	00	00	01	https://blog.opdp.pat/sepag222
01A0h:	7D	01	02	03	00	04	11	05	12	21	31	41	06	13	51	61	}!lAQa

### 首先删除一个头看看





删除头1,就获得了这个然后这个很明显就是二维码还没有结合 使用PS,进行修复把类似参考这个



https://blog.csdn.net/pone2233

### 我们首先把2个图片何在一起





### 我可真实二维码修复第一人哈哈哈,这里因为他给的一个图片有残缺我送大家完好的



和大家说一下流程,首先2张图片对在一起,比对首先先把下面没有的,给取出来



然后就可以操作了,把除了红框里面的进行比对如果颜色深那就把他变成白色,如果浅色变成黑色,然后弄完之后,在加上这个 底部进行合并,用魔棒工具取白色变成红色,然后把黑色变成白色,在把红色变成黑色,就成功了

https://blog.csdn.net/pone22:

QR Research	- 🗆 X
文件(F) 工具(T) 帮助(H)	
📄 🔛 🔝 🔌 📀	
	纠错等级 H(30%) 从uto 从本 Auto Auto Auto
	~
已解码数据 1:	^
位置:(6.7,0.4)-(362.3,6.2)-(-0.5,366.4)-(355.0,374.7) 颜色正常,正像 版本:1 纠错等级:L,掩码:1 内容: key:apngisamazing	
解码完成	mups//bioglosamnevponezzos

密钥1: apngisamazing



<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<cp:coreProperties xmlns:cp="http://schemas.openxmlformats.org/package/2006/metadata/core-properties" <mlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:dcmitype="http://purl.org/dc/dcmitype/" <mlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><dc:creator>G a1@xy</dc:creator><cp:lastModifiedBy>G a1@xy</cp:lastModifiedBy><dcterms:created xsi:type="dcterms:W3CDTF">2020-07-29T08:58:43Z</dcterms:created><dcterms:modified <si:type="dcterms:W3CDTF">2020-07-29T09:27:15Z</dcterms:modified></cp:coreProperties>

https://blog.csdn.net/pone2233

Ga1@xy师傅出的tql

### eeeeeeasyusb

这道题目和之前做的知识点就在一起了零宽度加密,本来是有网址的,现在似乎给拦截了



### 浙江省公安温馨提醒:

您访问的330k.github.io/misc\_tools/unicode\_steganography.html 该网站被大量用户举报,含有未经证实的信息,可能造成您的损失,建议谨慎访问!

https://blog.csdn.net/pone2233

那拿出比赛的时候的图把

JavaScript library is below. http://330k.github.io/misc\_tools/unicode\_steganography.js

#### Text in Text Steganography Sample

Original Text: Clear (length: 32)	Steganography Text: Clear (length: 56)
神秘代码, DYcbU-gQz_TZCBjh8rID/JmTjTw	神秘代码: DYcbU-gQz_TZCBjh8rID/JmTjTw Encode »
Hidden Text: Clear (length: 3)	
nut	« Decode
	Download Stego Text as File

nut翻译一下就是坚果

https://www.jianguoyun.com/p/DYcbU-gQz\_TZCBjh8rID 这个则是密码JmTjTw 然后下载下来,发现2个都是usb数据流量包,我推荐参考这个文章做 https://www.cnblogs.com/ECJTUACM-873284962/p/9473808.html 不过不太准确需要后期修改, 我们先把usb的信息提取出来 tshark -r part1.pcapng -T fields -e usb.capdata >data1.txt tshark -r part2.pcapng -T fields -e usb.capdata >data2.txt

这边,我卡了很久,应为他的脚本都很有问题,然后技能尚浅,所以脚本有点不太会改 我这边推荐一个师傅的博客大家可以看一下 http://www.fzwjscj.xyz/index.php/archives/38/#eeeeeeeasyusb 在这个师傅这个里边学边敲,还是很方便的,我建议可以保存一波,然后我这边送大家一个画图的脚本把

```
#!/usr/bin/python
# coding: utf-8
import matplotlib.pyplot as plt
import numpy as np
import matplotlib as mpl
mpl.rcParams['font.family'] = 'sans-serif'
mpl.rcParams['font.sans-serif'] = 'NSimSun,Times New Roman'
x, y = np.loadtxt('res.txt', delimiter=' ', unpack=True)
plt.plot(x, y, '*', label='Data', color='black')
plt.xlabel('x')
plt.vlabel('y')
plt.title('Data')
plt.legend()
plt.show()
```



#### 推荐反一下看: 166433882cd04aaa

在然后分析第二个文件,第二个文件很明显就少很多信息,就应该是键盘了,然后键盘脚本,我自己码,把虚拟机码炸了,太恐 怖了,我太菜了

然后还是哪里哪个师傅的脚本,存了,感谢师傅

```
all.deb 404 Not Found [IP: 192.99.200.113 80]
E: 无法下载 http://http.kali.org/kali/pool/main/w/wireshark/libwsutil11_3.2.3-1_i386.d
eb 404 Not Found [IP: 192.99.200.113 80]
E: 无法下载 http://http.kali.org/kali/pool/main/w/wireshark/libwiretap10_3.2.3-1_i386.
deb 404 Not Found [IP: 192.99.200.113 80]
E: 无法下载 http://http.kali.org/kali/pool/main/w/wireshark/libwireshark13_3.2.3-1_i38
6.deb 404 Not Found [IP: 192.99.200.113 80]
E: 无法下载 http://http.kali.org/kali/pool/main/w/wireshark/wireshark-qt_3.2.3-1_i386.
    404 Not Found [IP: 192.99.200.113 80]
deb
E: 无法下载 http://http.kali.org/kali/pool/main/w/wireshark/tshark_3.2.3-1_i386.deb 4
04 Not Found [IP: 192.99.200.113 80]
E: 无法下载 http://http.kali.org/kali/pool/main/w/wireshark/wireshark-common_3.2.3-1_i
386.deb 404 Not Found [IP: 192.99.200.113 80]
E: 无法下载 http://http.kali.org/kali/pool/main/w/wireshark/wireshark_3.2.3-1_i386.deb
 404 Not Found [IP: 192.99.200.113 80]
E: 有几个软件包无法下载,要不运行 apt-get update 或者加上 ---fix-missing 的选项再试试?
pjy@admin:~/桌面/8-25/Y0u_f0u1d_1t$ python2 data2.py
File "data2.py", line 8
SyntaxError: Non-ASCII character '\xe9' in file data2.py on line 8, but no encoding de clared; see http://python.org/dev/peps/pep-0263/ for details
                /8-25/Y0u_f0u1d_1t$ python2 data2.py
:~/桌面/8-25/Y0u_f0u1d_1t$
```

	第一種	中方式											
A aaaaa	H aabbb	0 abbba	V babab										
B aaaab	I abaaa	P abbbb	W babba										
C aaaba	J abaab	Q baaaa	X babbb										
D aaabb	K ababa	R baaab	Y bbaaa										
E aabaa	L ababb	S baaba	Z bbaab										
F aabab	M abbaa	T baabb											
G aabba	N abbab	U babaa											
	第二种方式												
a AAAAA	g AABBA	n ABBAA	t BAABA										
D AAAAB	h AABBB	o ABBAB	u-v BAABB										
c AAABA	i-j ABAAA	p ABBBA	w BABAA										
d AAABB	k ABAAB	q ABBBB	x BABAB										
e AABAA	1 ABABA	r BAAAA	y BABBA										
f AABAB	m ABABB	s BAAAB	z BABBB <sup>2233</sup>										

056bd4ad29bb522b

最终flag是: 166433882cd04aaa056bd4ad29bb522b

### 参考文献

这里由衷的感谢这位师傅,让我没卡住! 师傅1: http://www.fzwjscj.xyz/index.php/archives/38/

师傅2: https://www.cnblogs.com/ECJTUACM-873284962/p/9473808.html