

# CTFmisc图像题（zsteg取zip、压缩包重组、IDAT数据隐写、Markdown编写LaTeX、零宽字节隐写）

原创

[Hardworking666](#) 于 2022-01-02 20:02:32 发布 2978 收藏

分类专栏: [CTF](#) 文章标签: [zsteg](#) [压缩包重组](#) [IDAT隐写](#) [latex](#) [零宽字节隐写](#)

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/Hardworking666/article/details/122274926>

版权



[CTF 专栏收录该内容](#)

21 篇文章 2 订阅

订阅专栏

## 文章目录

一、题目描述

二、题目分析与完成

[zsteg取zip](#)

[压缩包重组](#)

[IDAT数据隐写](#)

[Markdown编写LaTeX](#)

[二维码Version](#)

[零宽字节隐写](#)

三、BMZCTF其他赛题wp

## 一、题目描述

BMZCTF第二届网络安全公开赛, 主办单位: 白帽子社区·WHT战队, 2022年1月1日, 白帽杯 misc

题目名称: 游戏秘籍

题目说明: 做为程序员我的, 竟然无法调出30条命, 我用笔记记录下来了。

游戏秘籍hint: 压缩包注意标识, 再重组。

游戏秘籍hint2: 笔记是用markdown记录的。

下载一张名为30.png的魂斗罗图片：



## 二、题目分析与完成

zsteg取zip

zsteg 30.png查看一下（也可以zsteg -a 30.png）：

```
root@kali:~# zsteg /root/桌面/30.png
[?] 945 bytes of extra data after image end (IEND), offset = 0x9f55d
extradata:0
00000000: 4e 65 65 64 5f 46 69 6e 64 5f 54 68 65 5f 50 61 |Need_Find_Th
e_Pa|
00000010: 73 73 77 64 5f 74 6f 5f 75 6e 7a 69 70 5f 66 69 |sswd_to_unzi
p_fi|
00000020: 6c 65 30 50 4b 01 02 3f 00 14 00 01 00 08 00 07 |le0PK..?....
....|
00000030: 6e 8f 53 9f d2 0e ea fc 02 00 00 f7 23 00 00 08 |n.S.....
# ... |
00000040: 00 24 00 00 00 00 00 00 00 20 00 00 00 00 00 00 |.$.....
....|
00000050: 00 46 6c 61 67 2e 74 78 74 0a 00 20 00 00 00 00 |.Flag.txt..
....|
00000060: 00 01 00 18 00 c3 6f 6c 59 77 f1 d7 01 46 35 e8 |.....oYw..
.F5.|
00000070: 59 77 f1 d7 01 90 34 f3 37 42 f1 d7 01 05 06 00 |Yw...47B..
....|
00000080: 00 00 00 01 00 01 00 5a 00 00 00 22 03 00 00 00 |.....Z..."
....|
00000090: 00 03 04 14 00 01 00 08 00 07 6e 8f 53 9f d2 0e |.....n.
S ... |
000000a0: ea fc 02 00 00 f7 23 00 00 08 00 00 00 46 6c 61 |.....#....
.Fla|
000000b0: 67 2e 74 78 74 be 5b f6 13 c9 94 ea b4 4e 91 12 |g.txt.[....
.N..|
000000c0: 01 de e0 ff 8e 6a 6e 84 83 dc 77 3f 13 51 58 cd |.....jn ...w?
.QX.|
000000d0: ee 41 0c 2b 52 fa 28 1e e5 bc f2 53 67 a2 e8 ec |.A.+R.(...S
g ... |
000000e0: 35 ae 35 e9 7b 25 a9 36 af 1e 04 f3 0b 3b ef 0c |5.5.{%.6....
.; .. |
000000f0: 30 f3 52 90 02 04 6b ff 1c 71 35 28 de dd d1 aa |0.R ... k.. q5(
....|
root@kali:~#
```

CSDN @Hardworking666

发现有PK开头的字符，可能是压缩包，隐藏的数据位置处于 extradata:0  
提取压缩包：

```
zsteg -E "extradata:0" 30.png > 30.zip
```

## 压缩包重组

010editor打开，发现一段提示：Need\_Find\_The\_Passwd\_to\_unzip\_file0

```
30.zip x
0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0000h: 4E 65 65 64 5F 46 69 6E 64 5F 54 68 65 5F 50 61 Need_Find_The_Pa
0010h: 73 73 77 64 5F 74 6F 5F 75 6E 7A 69 70 5F 66 69 sswd_to_unzip_fi
0020h: 6C 65 30 50 4B 01 02 3F 00 14 00 01 00 08 00 07 le0PK..?.....
0030h: 6E 8F 53 9F D2 0E EA FC 02 00 00 F7 23 00 00 08 n.SY0.êü...÷#...
0040h: 00 24 00 00 00 00 00 00 00 20 00 00 00 00 00 00 .$.....
0050h: 00 46 6C 61 67 2E 74 78 74 0A 00 20 00 00 00 00 .Flag.txt.. ....
0060h: 00 01 00 18 00 C3 6F 6C 59 77 F1 D7 01 46 35 E8 .....Äo1Ywñ×.F5è
0070h: 59 77 F1 D7 01 90 34 F3 37 42 F1 D7 01 05 06 00 Ywñ×..4ó7Bñ×....
0080h: 00 00 00 01 00 01 00 5A 00 00 00 22 03 00 00 00 .....Z..."....
0090h: 00 03 04 14 00 01 00 08 00 07 6E 8F 53 9F D2 0E .....n.SY0.
00A0h: EA FC 02 00 00 F7 23 00 00 08 00 00 00 46 6C 61 êü...÷#.....Fla
00B0h: 67 2E 74 78 74 BE 5B F6 13 C9 94 EA B4 4E 91 12 SBNt@Hardworking666
```

删除504B0102前的数据后，双击打开发现压缩包损坏。

正常压缩包开头是50 4B 03 04，还包括50 4B 01 02和50 4B 05 06等字段。

一个 ZIP 文件由三个部分组成：

压缩源文件数据区+压缩源文件目录区+压缩源文件目录结束标志

CTF压缩包隐写类（zip、RAR）

50 4B 01 02: 目录中文件文件头标记

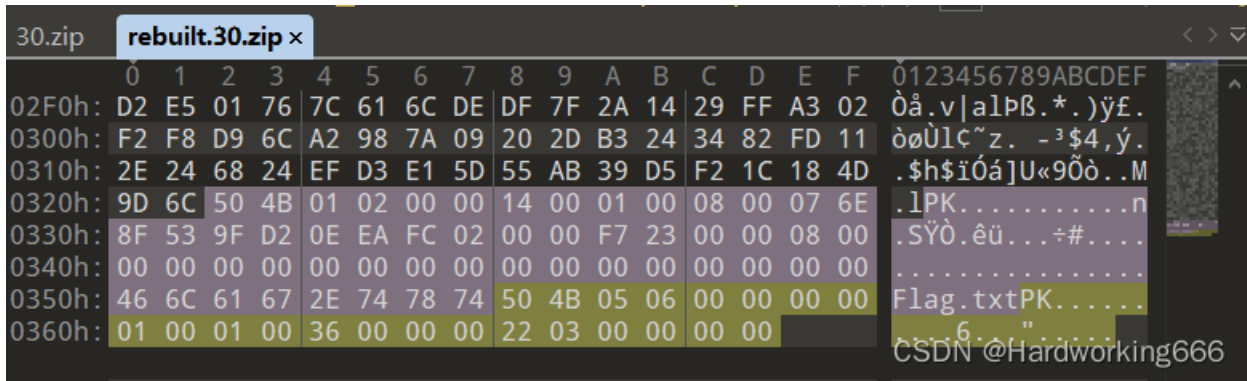
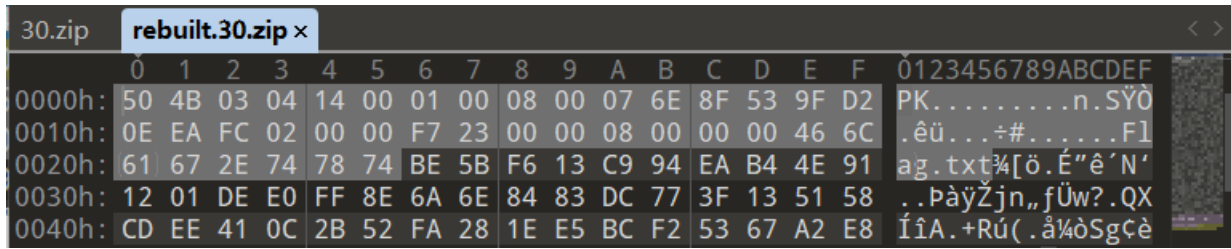
3F 00: 压缩使用的 pkware 版本

14 00: 解压文件所需 pkware 版本

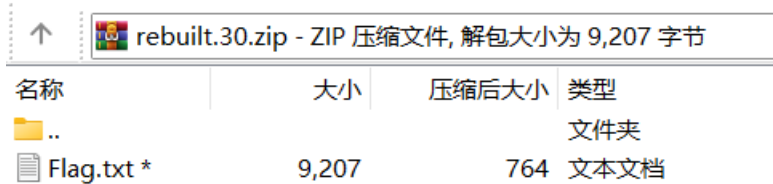
00 00: 全局方式位标记（有无加密，这里的 01 00为真加密）

08 00: 压缩方式

将此压缩包重组，搜索05 06、03 04 字段，发现均缺少50 4B，插入这两比特字节，然后用WinRAR自带的修复重组顺序。



压缩包可以正常打开，但需要密码。



## IDAT数据隐写

用010editor打开，发现很多类似这样的错误：

```
*ERROR: CRC Mismatch @ chunk[17]; in data: 5468655f; expected: 4a57f189
*错误: CRC不匹配 @ chunk[17]; 在数据: 5468655f; 预计: 4a57f189
```



## 输出

```
执行模板 'C:\Users\hong\OneDrive\文档\SweetScape\010 Templates\Repository\PNG.bt'  
*ERROR: CRC Mismatch @ chunk[17]; in data: 5468655f; expected: 4a57f189  
*ERROR: CRC Mismatch @ chunk[18]; in data: 68696e74; expected: 8f06e43a  
*ERROR: CRC Mismatch @ chunk[19]; in data: 5f69733a; expected: cd1cdd26  
*ERROR: CRC Mismatch @ chunk[20]; in data: 55316b6b; expected: 8b046c25  
*ERROR: CRC Mismatch @ chunk[21]; in data: 67473a56; expected: 0e3d8f3c  
*ERROR: CRC Mismatch @ chunk[22]; in data: 62444533; expected: 06a7a131  
*ERROR: CRC Mismatch @ chunk[23]; in data: 5b316f79; expected: ea364598  
*ERROR: CRC Mismatch @ chunk[24]; in data: 5b314565; expected: b0b27c70  
*ERROR: CRC Mismatch @ chunk[25]; in data: 60444579; expected: ab149ebf  
*ERROR: CRC Mismatch @ chunk[26]; in data: 60306768; expected: c221892a  
*ERROR: CRC Mismatch @ chunk[27]; in data: 537a463e; expected: 12d08536  
*ERROR: CRC Mismatch @ chunk[28]; in data: 3e3e3ee4; expected: f4b14517  
*ERROR: CRC Mismatch @ chunk[29]; in data: b88ae99d; expected: e08fdc44  
*ERROR: CRC Mismatch @ chunk[30]; in data: a2e5ad97; expected: a99df4de  
*ERROR: CRC Mismatch @ chunk[31]; in data: e7aca6e4; expected: ea4fd52e  
*ERROR: CRC Mismatch @ chunk[32]; in data: b8b2e5bc; expected: c7256870  
*ERROR: CRC Mismatch @ chunk[33]; in data: 82e68896; expected: 73c6ff48  
*ERROR: CRC Mismatch @ chunk[34]; in data: e99bb6e5; expected: b2f0bea8  
*ERROR: CRC Mismatch @ chunk[35]; in data: 8f89e99b; expected: 629f78e3  
*ERROR: CRC Mismatch @ chunk[36]; in data: b6e4b889; expected: 78f28831  
*ERROR: CRC Mismatch @ chunk[37]; in data: e784b6e5; expected: aae6059c  
*ERROR: CRC Mismatch @ chunk[38]; in data: 908ee59c; expected: 38ca3606  
*ERROR: CRC Mismatch @ chunk[39]; in data: a8424153; expected: 5c869776  
*ERROR: CRC Mismatch @ chunk[40]; in data: 453634e8; expected: 6579e7c3  
*ERROR: CRC Mismatch @ chunk[41]; in data: a7a3e7a0; expected: 069493c4  
*ERROR: CRC Mismatch @ chunk[42]; in data: 81e38082; expected: fe342207  
*ERROR: CRC Mismatch @ chunk[43]; in data: 596f755f; expected: ae426082  
*ERROR Line 332: 模板已通过变量 'data' 的文件结尾。 CSDN @Hardworking666
```

将in data:后的数据拷贝出来，按照顺序生成十六进制文件，

```
5468655f68696e745f69733a55316b6b67473a56624445335b316f795b3145656044457960306768537a463e3e3ee4b88ae  
99da2e5ad97e7aca6e4b8b2e5bc82e68896e99bb6e58f89e99bb6e4b889e784b6e5908ee59ca8424153453634e8a7a3e7a08  
1e38082596f755f
```

即新建16进制文本文档，然后在文本选项里选择导入16进制文件，最后在右下角改成UTF8编码。

```
 E F 0 1 2 3 4 5 6 7 8 9 A B C D E F  
6B 6B The _ hint _ is : U 1 k k  
45 65 g G : V b D E 3 [ 1 o y [ 1 E e  
3E E4 ` D E y ` 0 g h S z F > > > >  
E5 BC 上 面 字 符 串  
B8 89 异 或 零 叉 零 三  
34 E8 然 后 在 B A S E 6 4  
解 码 。 Y o u _ |
```

得到hint:

```
The_hint_is:U1kkgG:VbDE3[1oy[1EeDEy`0ghSzF>>>>>上面字符串异或零叉零三然后在BASE64解码。You_
```

写python脚本获得密码: What?That\_is\_a\_passwd?!

```
import base64  
  
str1 = "U1kkgG:VbDE3[1oy[1Ee`DEy`0ghSzF>>>>>"  
flag_base64 = ""  
for i in range(len(str1)):  
    a = ord(str1[i]) ^ 0x03  
    flag_base64 += chr(a)  
print(flag_base64)  
flag = base64.b64decode(flag_base64)  
print(flag)  
# What?That_is_a_passwd?!
```



我们知道在markdown语法中两个 `$$` 中可以加入LaTeX代码，就会表示成公式。

## # 游戏笔记

```


$$\begin{matrix} \uparrow\uparrow & 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 1 & 1 & 1 \\ 3 & 1 & 0 & 1 & 1 & 1 \\ 4 & 1 & 0 & 1 & 1 & 1 \end{matrix}$$


$$\begin{matrix} \downarrow\downarrow & 0 & 1 & 2 & 3 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 & 1 \\ 3 & 0 & 1 & 0 & 0 \\ 4 & 0 & 1 & 0 & 0 \end{matrix}$$


$$\begin{matrix} \rightarrow\rightarrow & 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 1 & 0 \\ 3 & 1 & 1 & 0 & 1 & 0 \\ 4 & 0 & 0 & 0 & 1 & 0 \end{matrix}$$


$$\begin{matrix} \leftarrow\leftarrow & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 0 & 1 & 1 \\ 3 & 1 & 1 & 1 & 1 \end{matrix}$$


$$\begin{matrix} \uparrow\uparrow & 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 3 & 1 & 1 & 1 & 0 & 1 \end{matrix}$$


$$\begin{matrix} \leftarrow\leftarrow & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 1 \end{matrix}$$


$$\begin{matrix} \rightarrow\rightarrow & 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 & 1 & 0 \\ 3 & 1 & 0 & 0 & 1 & 0 \end{matrix}$$


$$\begin{matrix} \leftarrow\leftarrow & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 1 & 1 \end{matrix}$$


$$\begin{matrix} \uparrow\uparrow & 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 2 & 1 & 0 & 1 & 1 & 1 \\ 3 & 0 & 1 & 1 & 1 & 1 \end{matrix}$$


$$\begin{matrix} \downarrow\downarrow & 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 2 & 0 & 0 & 0 & 1 \\ 3 & 1 & 1 & 0 & 1 \\ 4 & 0 & 1 & 0 & 1 \end{matrix}$$


$$\begin{matrix} \rightarrow\rightarrow & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 2 & 1 & 1 & 0 & 1 & 1 \\ 3 & 1 & 0 & 1 & 0 & 1 \\ 4 & 1 & 0 & 0 & 0 & 1 \end{matrix}$$


$$\begin{matrix} \leftarrow\leftarrow & 0 & 1 & 2 & 3 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 2 & 0 & 1 & 0 & 1 \\ 3 & 0 & 0 & 1 & 0 \\ 4 & 1 & 1 & 1 & 1 \end{matrix}$$


$$\begin{matrix} \rightarrow\rightarrow & 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 0 & 1 & 0 & 0 & 0 \\ 3 & 1 & 1 & 0 & 1 & 0 \\ 4 & 1 & 1 & 0 & 0 & 0 \end{matrix}$$


```

## 游戏笔记

$\uparrow\uparrow$	0	1	2	3	4	$\downarrow\downarrow$	0	1	2	3
0	1	1	1	1	1	0	1	1	0	0
1	1	0	0	0	0	1	0	1	0	0
2	1	0	1	1	1	2	0	1	0	1
3	1	0	1	1	1	3	0	1	0	0
4	1	0	1	1	1	4	0	1	0	0

$\uparrow\uparrow$	0	1	2	3	4	$\leftarrow\leftarrow$	0	1	2	3
0	1	0	0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1	0	1	0
2	0	0	0	0	0	2	1	0	1	1
3	1	1	1	0	1	3	1	1	0	1

CSDN @Hardworking666

## 游戏笔记

$\uparrow\uparrow$	0	1	2	3	4	$\downarrow\downarrow$	0	1	2	3	$\rightarrow\rightarrow$	0	1	2	3	4	$\leftarrow\leftarrow$	0	1	2	3	ABAB	1	2	3	4	5
0	1	1	1	1	1	0	1	1	0	0	0	1	1	0	1	1	0	0	0	1	0	0	1	1	1	1	1
1	1	0	0	0	0	1	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1
2	1	0	1	1	1	2	0	1	0	1	2	0	1	0	1	0	2	1	0	1	1	3	1	1	1	0	1
3	1	0	1	1	1	3	0	1	0	0	3	1	1	0	1	0	3	1	1	0	1	4	1	1	1	0	1
4	1	0	1	1	1	4	0	1	0	0	4	0	0	0	1	0	4	1	0	0	0						

$\uparrow\uparrow$	0	1	2	3	4	$\leftarrow\leftarrow$	0	1	2	3	$\rightarrow\rightarrow$	0	1	2	3	4	$\downarrow\downarrow$	0	1	2	3	ABAB	1	2	3	4	5
0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	1	
1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	
2	0	0	0	0	0	2	1	0	1	1	2	1	1	0	0	0	2	0	0	0	1	3	0	0	0	0	
3	1	1	1	0	1	3	1	1	0	1	3	1	0	1	1	0	3	1	1	1	1	3	0	0	1	0	

$\uparrow\uparrow$	0	1	2	3	4	$\downarrow\downarrow$	0	1	2	3	$\rightarrow\rightarrow$	0	1	2	3	4	$\leftarrow\leftarrow$	0	1	2	3	ABAB	1	2	3	4	5
0	1	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	
1	1	1	1	0	1	1	0	0	1	1	1	1	0	0	0	1	1	0	1	0	0	1	0	1	0	0	
2	1	0	1	1	1	2	1	1	1	1	2	0	0	1	1	0	2	0	0	1	0	2	1	0	1	0	
3	0	1	1	1	1	3	0	0	0	0	3	1	0	0	1	0	3	0	1	1	1	3	1	1	0	0	

$\uparrow\uparrow$	0	1	2	3	4	$\downarrow\downarrow$	0	1	2	3	$\leftarrow\leftarrow$	0	1	2	3	ABAB	1	2	3	4	5	$\rightarrow\rightarrow$	0	1	2	3	4
0	1	0	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	1	
1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	
2	1	0	1	1	1	2	0	1	0	1	2	1	1	0	0	0	2	0	1	1	0	2	1	1	1	1	
3	1	0	0	0	0	3	0	1	0	1	3	0	1	1	0	3	0	0	1	1	0	3	1	0	1	0	
4	1	1	1	1	1	4	1	1	0	1	4	1	1	0	0	4	0	0	0	0	1	4	1	0	0	1	

CSDN @Hardworking666



看到5行5列的表格，全是由01组成，想到二维码。

根据题目魂斗罗调30条命，百度发现秘籍是：上上下下左右左右BABA（用上上下下左左右右ABAB来排列）。

但是，23x23尺寸的二维码不合理。

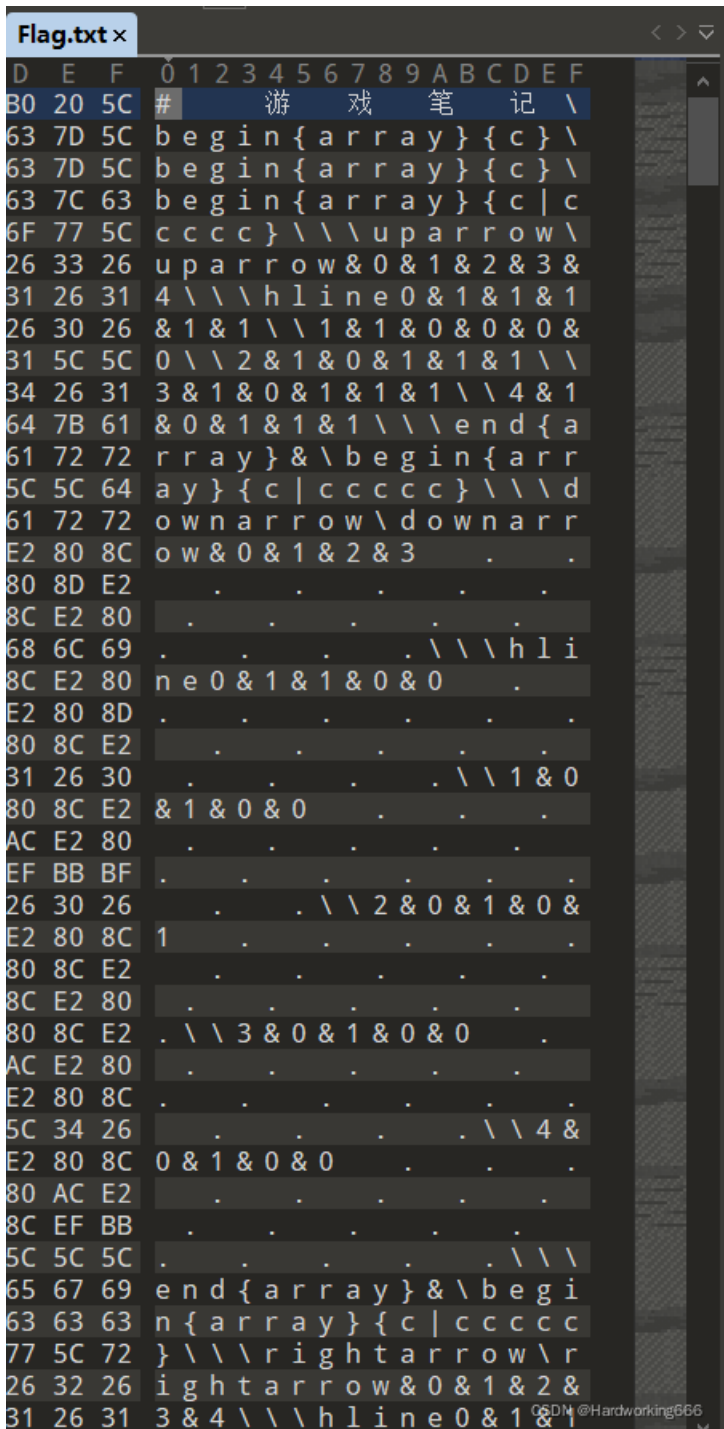
二维码官方叫版本为Version。

二维码一共有40个尺寸。Version 1是21 x 21的矩阵，Version 2是 25 x 25的矩阵，Version 3是29的尺寸，每增加一个version，就会增加4的尺寸，公式是： $(V-1)*4 + 21$ （V是版本号）最高Version 40， $(40-1)*4+21 = 177$ ，所以最高是177 x 177 的正方形。

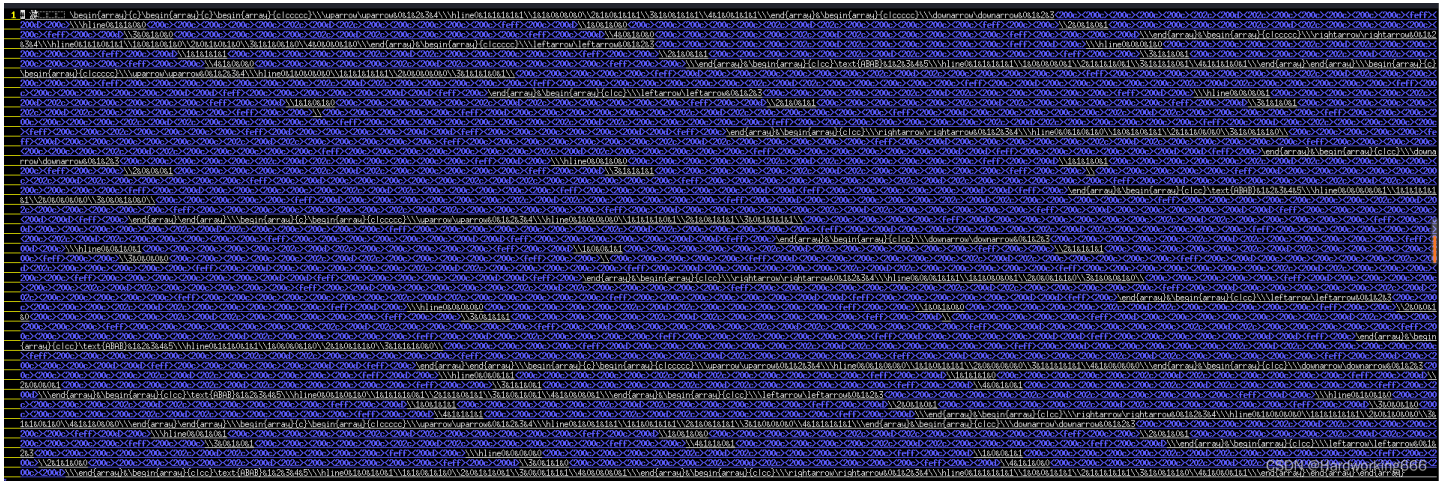
version2 二维码的标准大小应该是25x25。想到零宽字节隐写。

## 零宽字节隐写

这里用010editor打开，或者在kali下的vim打开字里行间填充的 `<200c><200c><200c>` 正是零宽字节的隐写特征。



```
Flag.txt x
D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F
B0 20 5C # 游 戏 笔 记 \
63 7D 5C begin{array}{c} \
63 7D 5C begin{array}{c} \
63 7C 63 begin{array}{c|c
6F 77 5C cccc} \ \ \uparrow \
26 33 26 uparrow&0&1&2&3&
31 26 31 4 \ \ \hline 0&1&1&1
26 30 26 &1&1 \ \ 1&1&0&0&0&
31 5C 5C 0 \ \ 2&1&0&1&1&1 \ \
34 26 31 3 &1&0&1&1&1 \ \ 4&1
64 7B 61 &0&1&1&1 \ \ \end{a
61 72 72 rray}& \ begin{arr
5C 5C 64 ay}{c|cccc} \ \ d
61 72 72 ownarrow \ downarr
E2 80 8C ow&0&1&2&3 .
80 8D E2 .
8C E2 80 .
68 6C 69 . \ \ \hli
8C E2 80 ne 0&1&1&0&0 .
E2 80 8D .
80 8C E2 .
31 26 30 . \ \ 1&0
80 8C E2 &1&0&0 .
AC E2 80 .
EF BB BF .
26 30 26 . \ \ 2&0&1&0&
E2 80 8C 1 .
80 8C E2 .
8C E2 80 .
80 8C E2 . \ \ 3&0&1&0&0 .
AC E2 80 .
E2 80 8C .
5C 34 26 . \ \ 4&
E2 80 8C 0&1&0&0 .
80 AC E2 .
8C EF BB .
5C 5C 5C . \ \ \
65 67 69 end{array}& \ begi
63 63 63 n{array}{c|cccc
77 5C 72 } \ \ \rightarrow \ r
26 32 26 ightarrow&0&1&2&
31 26 31 3&4 \ \ \hline 0&1&
```



零宽字节隐写解密网站: [https://330k.github.io/misc\\_tools/unicode\\_steganography.html](https://330k.github.io/misc_tools/unicode_steganography.html)

注意不要全部复制, 要单行解密。

从 `\begin{array}` 到 `\end{array}`

## Text in Text Steganography Sample

Original Text:  (length: 0)

Hidden Text:  (length: 2)

&0

Encode »

« Decode

Steganography Text:  (length: 16)

CSDN @Hardworking666

用上上下下左左右右ABAB来排列 (以此类推):

```
11111 # 上上的第一行
11011 # 下下的第一行
00101 # 左左的第一行
11011 # 右右的第一行
11111 # ABAB的第一行
```



BMZCTF官方靶场:

<http://www.bmzclub.cn/>

第二届“BMZCTF”网络安全公开赛CFS题目WriteUp:

<https://www.bll-l.com/index.php/archives/12/>

WHT战队博客 (BMZCTF往年真题):

[https://blog.csdn.net/qq\\_26243045](https://blog.csdn.net/qq_26243045)

第二届“BMZCTF”网络安全公开赛官方解题思路 (全部题):

[https://mp.weixin.qq.com/s/ZEfY-u9TgkWOgX\\_\\_BK40mw](https://mp.weixin.qq.com/s/ZEfY-u9TgkWOgX__BK40mw)