MISC

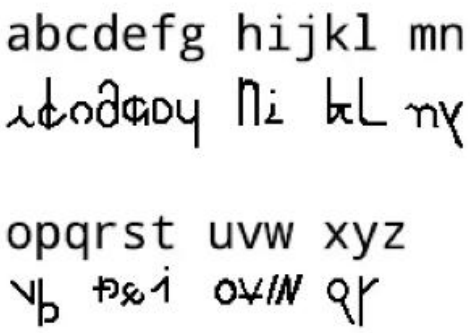
Hidden Image in LSB

利用 stegsolve

Broken Chest

Google 识图

可以得到

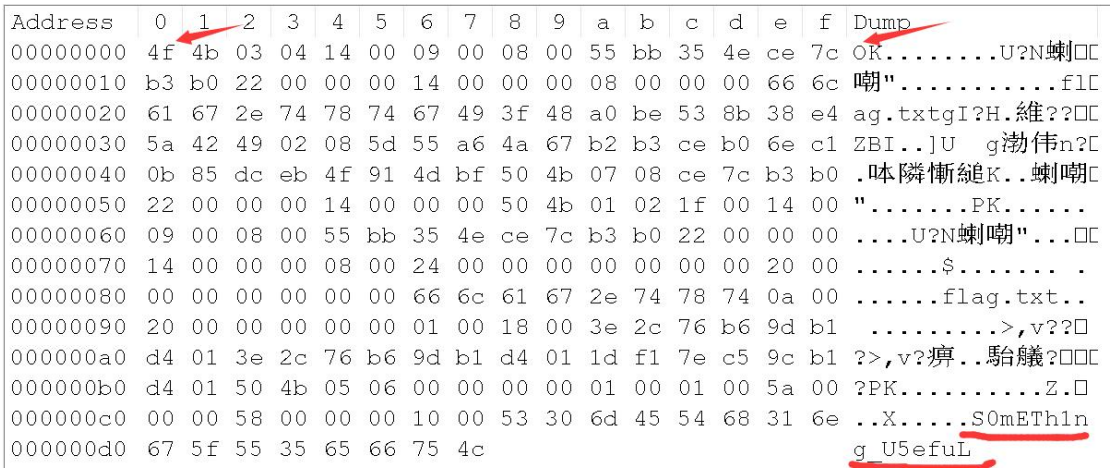


完整对于表

一一对照 由于是单词 可以猜出来一些未在图片中的符号
可得 hgame{My_vi0Let_tyPewRiter}

Broken Chest

Hex 查看



Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
00000000	4f	4b	03	04	14	00	09	00	08	00	55	bb	35	4e	ce	7c	OK.....U?N蜊□□
00000010	b3	b0	22	00	00	00	14	00	00	00	08	00	00	00	66	6c	嘲".....fl□
00000020	61	67	2e	74	78	74	67	49	3f	48	a0	be	53	8b	38	e4	ag.txtgI?H.維??□□
00000030	5a	42	49	02	08	5d	55	a6	4a	67	b2	b3	ce	b0	6e	c1	ZBI..]U g渤伟n?□
00000040	0b	85	dc	eb	4f	91	4d	bf	50	4b	07	08	ce	7c	b3	b0	.呫隣慚縋K..蜊嘲□
00000050	22	00	00	00	14	00	00	00	50	4b	01	02	1f	00	14	00	".......PK......
00000060	09	00	08	00	55	bb	35	4e	ce	7c	b3	b0	22	00	00	00	....U?N蜊嘲"...□□
00000070	14	00	00	00	08	00	24	00	00	00	00	00	00	00	20	00	......$..... .
00000080	00	00	00	00	00	00	66	6c	61	67	2e	74	78	74	0a	00	......flag.txt..
00000090	20	00	00	00	00	00	01	00	18	00	3e	2c	76	b6	9d	b1	.....>,v??□
000000a0	d4	01	3e	2c	76	b6	9d	b1	d4	01	1d	f1	7e	c5	9c	b1	?>,v?痹..駙艬?□□□
000000b0	d4	01	50	4b	05	06	00	00	00	00	01	00	01	00	5a	00	?PK.....Z.□
000000c0	00	00	58	00	00	00	10	00	53	30	6d	45	54	68	31	6e	..X.....S0mETh1n
000000d0	67	5f	55	35	65	66	75	4c									g_U5efuL

开头改为 50 同时注意到末尾 猜测是密码 更改后输入即可得到 flag
hgame{Cra2y_D1aM0nd}

CRYPTO

Mix

摩斯电码 → 两个一组对照 ascii →根据括号位置两个一组栅栏解码→凯撒解密

```
74  t
4B  K
73  s
5F  _
6D  m
6F  o
79  y
44  D
71  q
6B  k
7B  {
62  b
51  Q
66  f
34  4
30  0
65  e
7D  }
tK
s_
mo
yD
qk
{b
Qf
40
e}
tsmyq{Q4eK_oDkbf0}
hgame{E4sY_cRypt0}
```

PWN

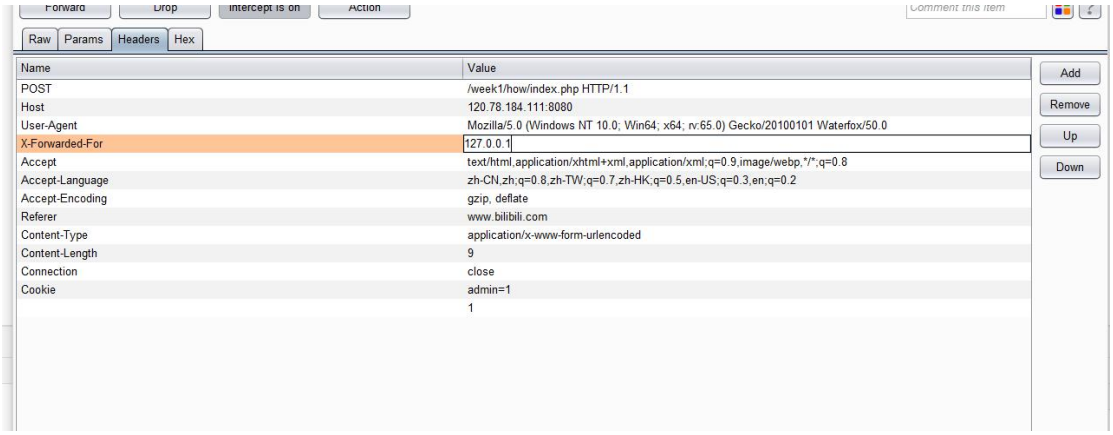
aaaaaaaaaaa
丢 ida　　nc 过去 a 就完事了

WEB

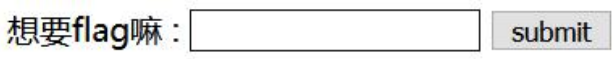
谁吃了我的 flag

Hint　vim 强退后会留有 swp 文件 改为/.index.html.swp 下载文件打开后可得 flag

换头大作战



利用 bp



hgame{hTTp_HeaDeR_iS_Ez}

very easy web[已完成]

是 bugku 的一道相似题型

```php
<?php
//如果是hackerDJ就echo，not allowed!并且推出程序
if(eregi("hackerDJ",$_GET[id])) {
    echo("not allowed!");
    exit();
}
//进行一个urldecode转码
$_GET[id] = urldecode($_GET[id]);
//如果还是一个hackerDJ就输出flag
if($_GET[id] == "hackerDJ")
{
    echo "Access granted!";
echo "flag";
}
?>
```

RE

Brainfxxker

可得伪代码

```
,>++++++++++        ///a[0]=getchar()   a[1]=10
|
[<---------->-]     ///while (a[1] ) {a[0] = a[0] - 10 a[1]--}
<++[+.]             ///a[0] = a[0] + 2 while( a[0] ) {a[0]++   putchar(a[0])}
,>+++++++++         ///a[0] = getchar()   a[1] = a[1] + 9

[<---------->-]     ///while (a[1]) { a[0] = a[0] - 9; a[1]--}
<-[+.]              ///a[0]-- ; while( a[0] )  {a[0]++; putchar(a[0])}
,>+++++++           /// a[0] = getchar  a[1] = a[1] + 7

[<------->-]        ///while(a[1]) {a[0] = a[0] - 7  a[1]--}
<---[+.]            ///a[0] = a[0]-3 while (a[0])  {a[0]++  putchar(a[0])
,>++++++            ///a[0] = getchar()   a[1] = a[1] + 6

[<------>-]         ///while(a[1]) {a[0] = a[0] - 6  a[1]--}
<+++[+.]            ///a[0] = a[0]+3 while (a[0])  {a[0]++  putchar(a[0])
,>++++++++          ///a[0] = getchar()   a[1] = a[1] + 8

[<---------->-]     ///while(a[1]) {a[0] = a[0] - 10  a[1]--}
<++[+.]             ///a[0] = a[0]+2 while (a[0])  {a[0]++  putchar(a[0])
,>++++++++++        ///a[0] = getchar()   a[1] = a[1] + 10

[<---------->-]     ///while(a[1]) {a[0] = a[0] - 10  a[1]--}
<--[+.]             ///a[0] = a[0]-2 while (a[0])  {a[0]++  putchar(a[0])
,>++++++++++        ///a[0] = getchar()   a[1] = a[1] + 10

[<--------->-]      ///while(a[1]) {a[0] = a[0] - 8  a[1]--}
<-----[+.]          ///a[0] = a[0]-5 while (a[0])  {a[0]++  putchar(a[0])
,>++++++++++        ///a[0] = getchar()   a[1] = a[1] + 10
```

因为 uint8_t 类型，要保证执行[+.]操作时 ，while 里面的条件为假，即 a[0]==0，往上推，即可退出输入的 char 是什么，可得 flag

HelloRe

丢 IDA

わかります

r & xor

r 之后可得到字符，因为储存顺序的关系，颠倒之后可得假 flag，

```
mov     [rbp+var_118], 1
mov     [rbp+var_110], 7
mov     [rbp+var_108], 92
mov     [rbp+var_104], 18
mov     [rbp+var_100], 38
mov     [rbp+var_FC], 11
mov     [rbp+var_F8], 93
mov     [rbp+var_F4], 43
mov     [rbp+var_F0], 11
mov     [rbp+var_EC], 23
mov     [rbp+var_E4], 23
mov     [rbp+var_E0], 43
mov     [rbp+var_DC], 69
mov     [rbp+var_D8], 6
mov     [rbp+var_D4], 86
mov     [rbp+var_D0], 44
mov     [rbp+var_CC], 54
mov     [rbp+var_C8], 67
mov     [rbp+var_C0], 66
mov     [rbp+var_BC], 85
mov     [rbp+var_B8], 126
mov     [rbp+var_B4], 72
mov     [rbp+var_B0], 85
mov     [rbp+var_AC], 30
```

可得异或的数组，注意开头的 1 和 7 占 8 字节，所以为 1，0，7，0，后面 23 同理

```
#include<stdio.h>
char a2[36]="hgame{Y0u_mayb3_need_th1s_0ne!!!!!}";
short int b[36]={0,0,0,0,0,0,
1,0,7,0,92,18,38,11,93,43,11,23,0,23,43,69,6,86,44,54,67,0,66,85,126,72,85,30,0
};
int main ()
{
    char aa[35];
    int i;
    for(i=0;i<35;i++)
    {
        aa[i] = a2[i]^b[i];
    }
    for (i=0;i<35;i++)
    {
        printf("%c",aa[i]);
    }
}
```

```
C:\Users\Mr.Zhou\Documents\123.exe
hgame{X0r_1s_interestlng_isn't_it?}
Process returned 125 (0x7D)   execution time : 1.910 s
Press any key to continue.
```

Pro 的 Python 教室(一)

```
enc1 = 'hgame{'
enc2 = 'SGVyZV8xc18zYXN5Xw=='
enc3 = 'Pyth0n}'
```

对 enc2 base64 解密 可得 Here_1s_3asy_