RE

1，secret

elf，用ida打开，从main中看不到什么，经过调试发现是个假的main找到真正的main

```
unsigned __int64 real_main()

  anti_debug();
  return flag_XTEA();
```

先看的第一个函数

```
v6 = __readfsqword(0x28u);
v0 = getppid();
snprintf(&s, 0x18uLL, "/proc/%d/cmdline", v0);
stream = fopen(&s, "r");
fgets(&v5, 256, stream);
fclose(stream);
v2 = strlen(&v5);
hash256((__int64)&v5, v2, (__int64)&unk_6032E0);// hash256
if ( !memcmp(&unk_6032E0, &unk_603100, 0x20uLL) )
{
```

在这里卡了好久，我开始一直以为是在这里加密的，加密后与unk_6032E0相比对，因为对加密算法不熟悉，对上图中hash256的分析耗了将近两天，当我分析出是hash256的时候简直崩溃～

不过得到这个结果，确定了此处不是关键，看下一函数

```
v5 = __readfsqword(0x28u);
sigemptyset((sigset_t *)&v2);
v1 = sub_4019EA;
v3 = 4;
sigaction(34, (const struct sigaction *)&v1, (struct sigaction *)&v4);
sigemptyset((sigset_t *)&v2);
v1 = sub_401A09;
v3 = 4;
sigaction(35, (const struct sigaction *)&v1, (struct sigaction *)&v4);
sigemptyset((sigset_t *)&v2);
v1 = sub_401A3A;
v3 = 4;
sigaction(36, (const struct sigaction *)&v1, (struct sigaction *)&v4);
v1 = sub_401AA4;
v3 = 0;
sigaction(37, (const struct sigaction *)&v1, (struct sigaction *)&v4);
v1 = sub_401AF6;
v3 = 0;
sigaction(38, (const struct sigaction *)&v1, (struct sigaction *)&v4);
v1 = sub_401B11;
v3 = 0;
sigaction(39, (const struct sigaction *)&v1, (struct sigaction *)&v4);
v1 = sub_401B66;
v3 = 0;
sigaction(40, (const struct sigaction *)&v1, (struct sigaction *)&v4);
v1 = check;
v3 = 0;
sigaction(41, (const struct sigaction *)&v1, (struct sigaction *)&v4);
return __readfsqword(0x28u) ^ v5;
```

在这个函数里找到了

```
signed int i; // [rsp+Ch] [rbp-4h]

for ( i = 0; i <= 13; ++i )
{
  if ( dword_603280[i] != dword_603200[i] )
  {
    puts("\nSorry, looks like you don't understand yet");
    exit(0);
  }
}
puts("\nIt seems you understand. Flag is your input");
exit(0);
```

对这个函数进行分析

```
v0 = (k[((unsigned int)sum >> 11) & 3] + sum) ^ ((((unsigned int)::v0 >> 5) ^ 16 * ::v0) + ::v0);
result = v0 + v1;
v1 += v0;
return result;
```

```
__int64 result; // rax

result = (unsigned int)(sum + delta);
sum += delta;
return result;
```

可以得出是XTEA加密，但是这一串sigaction打的我有点懵，经过了一天时间四处查，得到了我不用在意sigaction的结论（╯' – ')╯┻━┻

于是再次调试，得到了key

```
k[4]={0x42655F29,0x9E822EFC,0xDA278C92,0x4E355A62}
```

unk_603200为

```
E9C8A927 B473A9BA F972C0AA 0080FAA3 D3C2F4D9 C56B3FFB 5ED9D3D3
771D9686 3FC500E6 B927BC98 ACC3AA09 2424DC6A 04E30506 778CE765
```

之前的分析中我得到了每两个32位的数进行XTEA，所以解密时要改为

```
27A9C8E9 BAA973B4 AAC072F9 A3FA8000 D9F4C2D3 FB3F6BC5 D3D3D95E
86961D77 E600C53F 98BC27B9 09AAC3AC 6ADC2424 0605E304 65E78C77
```

```
#include <stdio.h>
#include <stdint.h>
void decipher(unsigned int num_rounds, uint32_t v[2], uint32_t
const key[4]) {
    unsigned int i;
    uint32_t v0=v[0], v1=v[1], delta=0x9E3779B9,
sum=delta*num_rounds;
    for (i=0; i < num_rounds; i++) {
        v1 -= (((v0 << 4) ^ (v0 >> 5)) + v0) ^ (sum +
key[(sum>>11) & 3]);
        sum -= delta;
```

```
        v0 -= (((v1 << 4) ^ (v1 >> 5)) + v1) ^ (sum + key[sum &
3]);
    }
    v[0]=v0; v[1]=v1;
}

int main()  {
    uint32_t v[2]={0x0605E304,0x65E78C77};
    uint32_t const k[4]={0x42655F29,0x9E822EFC,
0xDA278C92,0x4E355A62};
    unsigned int r=32;
    printf("加密前原始数据: %x %x\n",v[0],v[1]);
    decipher(r, v, k);
    printf("%u 解密后的数据: %x %x\n",r,v[0],v[1]);

    return 0;
}
```

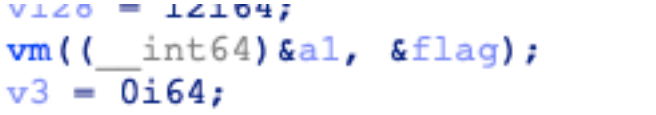
于是得到

```
6d616768 6f4e7b65    hgame{No
654e305f 4e34635f    _0Ne_c4N
5f30745f 405f6542    _t0_Be_@
6174245f 68542e52    _$taR.Th
735f7933 4c6c3174    3y_st1lL
6e41435f 4e34635f    _CAn_c4N
3148735f 7d2e336e    _sH1n3.}
```

得到flag

2,easyvm

拖进ida,



将一些数据，和我们的flag进行处理，vm函数有一些switch，可以看出是vm题（当然题目告诉我了）

a1就是加密的处理

```
v2 = a1;
v23 = flag;
stack = sub_7FF64C0F1000(0xC8ui64);
v21 = v2;
v22 = v2;
```

按正常思路分析虚拟机指令，可以看到要了一段内存，应该是作为栈，就从可以看出push和pop
本来想分析清楚每个操作，但在调试中发现了其实关键就是xor操作而异或的值在

```
mov     rcx, [rsi+8]
lea     rdx, [rax+rcx*8]
mov     rax, [rbp+57h+C]
xor     [rdx], rax
jmp     loc_7FF64C0F1A9C
; ---------------------------
```

就是此时rax的值，那么

```
c=[0x3A,0x54,0x2F,0x2A,0x2F,0x36,0x13,0x01,0x2E,
0x03,0x35,0x40,0x47,0x0E,0x5F,0x59,0x01,0x69,0x27,0x08,0x3D,0x4C,
0x33,0x1A,0x2D,0x0B,0x40,0x0E,0x4B,
0x24,0x41,0x27,0x25,0x28,0x29,0x2A,0x02,0x02,0x5D,0x24]
x=[0x52,0x33,0x4e,0x47,0x4a,0x4d,0x67,0x69,0x47,0x70,0x6a,
0x36,0x2a,0x51,0x36,0x2a,0x5e,0x36,0x54,0x67,0x4e,
0x23,0x40,0x75,0x5e,0x64,0x33,0x61,0x38,0x4b,
0x32,0x48,0x56,0x47,0x76,0x4f,0x63,0x71,0x24,0x59]

for i in range(len(c)):
    print(chr(c[i]^x[i]),end='')
```

得到flag（感觉有点对不起这道题（‘_‘））