RE

1,babypy

对字节码硬怼，就可分析出大致逻辑是倒序后异或

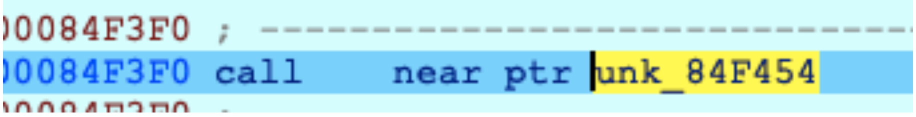
```
c=[0x7d,0x03,0x7d,0x04,0x57,0x17,0x72,0x2d,0x62,0x11,0x4e,0x6a,
0x5b,0x04,0x4f,0x2c,0x18,0x4c,0x3f,0x44,0x21,0x4c,0x2d,0x4a,0x22]
c.reverse()
flag=[]
for i in range(len(c)):
    if i!=len(c)-1:
        print(chr(c[i]^c[i+1]),end='')
    else:
        print(chr(c[i]))
```

得到flag

2,unpack

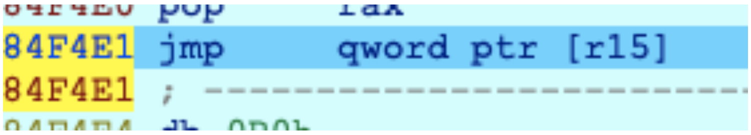
题如其名，手脱upx壳，虽然简单，但linux上还是第一次脱

用ida调试，f8不跑飞就不用f7，一直到



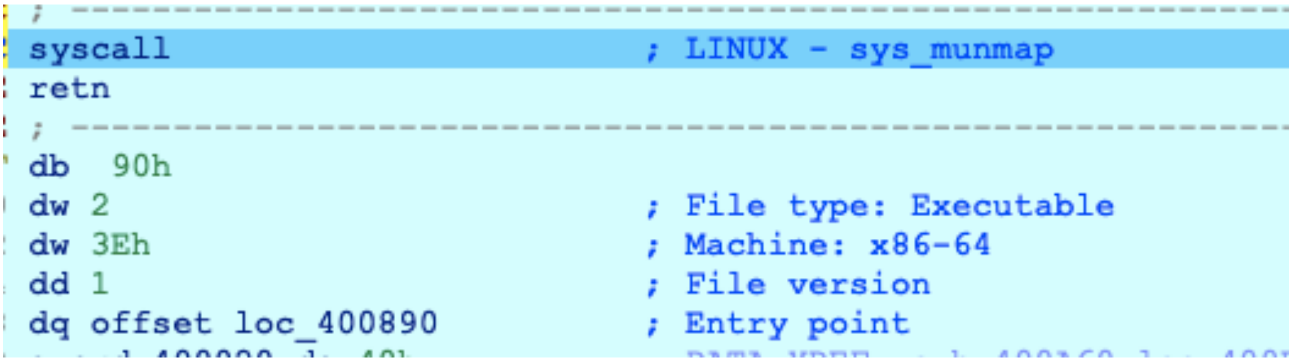
```
00084F3F0 ; ---------------------------
00084F3F0 call     near ptr unk_84F454
```

f7进去，再一直f8



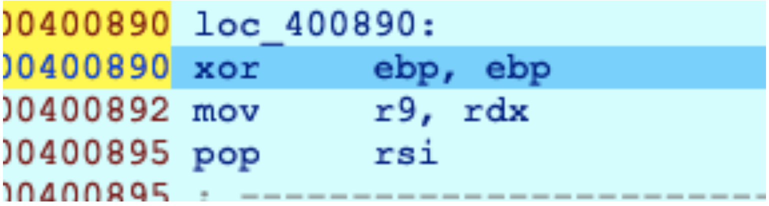
```
84F4E0 pop     rax
84F4E1 jmp     qword ptr [r15]
84F4E1 ; ----------------------
```

到了一个跳转，f8



```
 ; ---------------------------------------------------------
 syscall                          ; LINUX - sys_munmap
 retn
 ; ---------------------------------------------------------
 db  90h
 dw  2                            ; File type: Executable
 dw  3Eh                          ; Machine: x86-64
 dd  1                            ; File version
 dq offset loc_400890             ; Entry point
```

可以看到入口点0x400890，继续f8

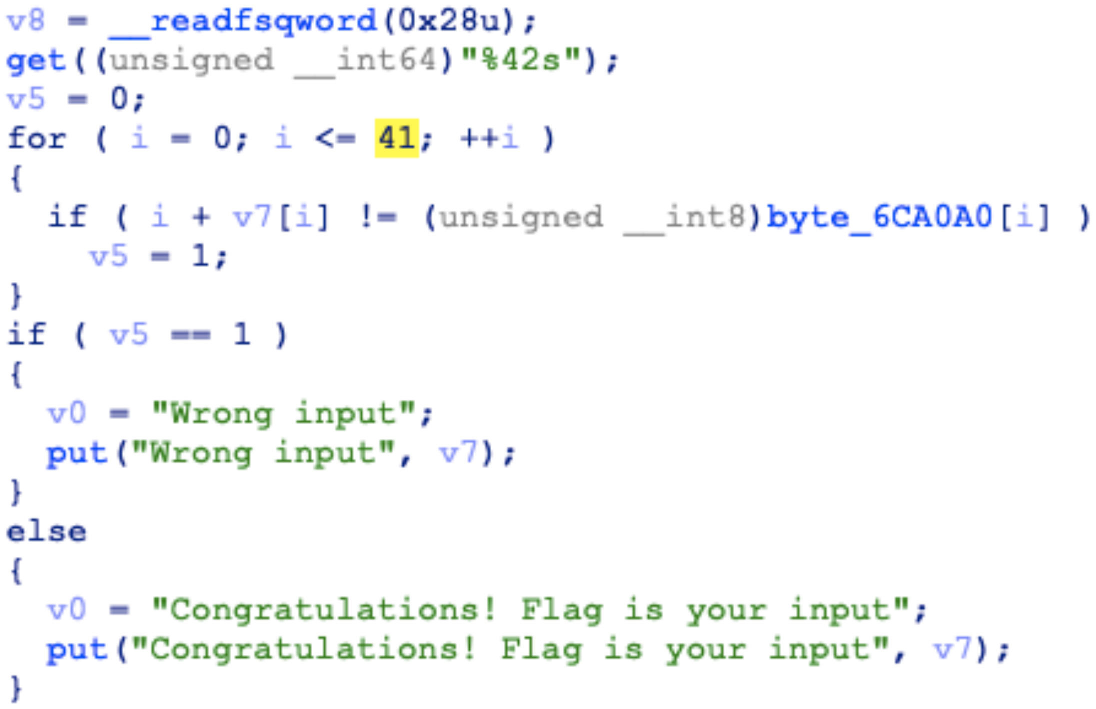


```
00400890 loc_400890:
00400890 xor      ebp, ebp
00400892 mov      r9, rdx
00400895 pop      rsi
00400895 ; ----------------------
```

来到了入口，dump出来

```
#include <idc.idc>
#define PT_LOAD              1
#define PT_DYNAMIC           2
static main(void)
{
         auto ImageBase,StartImg,EndImg;
         auto e_phoff;
         auto e_phnum,p_offset;
         auto i,dumpfile;
         ImageBase=0x400000;
         StartImg=0x400000;
         EndImg=0x0;
         if (Dword(ImageBase)==0x7f454c46 ||
Dword(ImageBase)==0x464c457f )
  {
     if(dumpfile=fopen("G:\\dumpfile","wb"))
    {
      e_phoff=ImageBase+Qword(ImageBase+0x20);
      Message("e_phoff = 0x%x\n", e_phoff);
      e_phnum=Word(ImageBase+0x38);
      Message("e_phnum = 0x%x\n", e_phnum);
      for(i=0;i<e_phnum;i++)
      {
         if (Dword(e_phoff)==PT_LOAD ||
Dword(e_phoff)==PT_DYNAMIC)
                         {
                                 p_offset=Qword(e_phoff+0x8);
                                 StartImg=Qword(e_phoff+0x10);

EndImg=StartImg+Qword(e_phoff+0x28);
                                 Message("start = 0x%x, end =
0x%x, offset = 0x%x\n", StartImg, EndImg, p_offset);

dump(dumpfile,StartImg,EndImg,p_offset);
                                 Message("dump segment %d ok.
\n",i);
                         }
         e_phoff=e_phoff+0x38;
      }

      fseek(dumpfile,0x3c,0);
      fputc(0x00,dumpfile);
      fputc(0x00,dumpfile);
      fputc(0x00,dumpfile);
      fputc(0x00,dumpfile);

      fseek(dumpfile,0x28,0);
      fputc(0x00,dumpfile);
      fputc(0x00,dumpfile);
      fputc(0x00,dumpfile);
      fputc(0x00,dumpfile);
      fputc(0x00,dumpfile);
      fputc(0x00,dumpfile);
      fputc(0x00,dumpfile);
      fputc(0x00,dumpfile);

      fclose(dumpfile);
        }else Message("dump err.");
 }
}
static dump(dumpfile,startimg,endimg,offset)
{
        auto i;
        auto size;
        size=endimg-startimg;
        fseek(dumpfile,offset,0);
        for ( i=0; i < size; i=i+1 )
        {
        fputc(Byte(startimg+i),dumpfile);
        }
}
```

再看dump出来的文件，用ida打开，



```
v8 = __readfsqword(0x28u);
get((unsigned __int64)"%42s");
v5 = 0;
for ( i = 0; i <= 41; ++i )
{
  if ( i + v7[i] != (unsigned __int8)byte_6CA0A0[i] )
    v5 = 1;
}
if ( v5 == 1 )
{
  v0 = "Wrong input";
  put("Wrong input", v7);
}
else
{
  v0 = "Congratulations! Flag is your input";
  put("Congratulations! Flag is your input", v7);
}
```

逻辑就十分简单了，脚本就不放了。

1,crackme

c#写的，关键代码

```
private void button1_Click(object sender, EventArgs e)
{
    if (this.status == 1)
    {
        MessageBox.Show("你已经激活成功啦，快去提交flag吧~~~");
        return;
    }
    string text = this.textBox1.Text;
    if (text.Length != 46 || text.IndexOf("hgame{") != 0 ||
text.IndexOf("}") != 45)
    {
        MessageBox.Show("Illegal format");
        return;
    }
    string base64iv = text.Substring(6, 24);
    string str = text.Substring(30, 15);
    try
    {
        Aes aes = new Aes("SGc0bTNfMm8yMF9XZWVLMg==", base64iv);
        Aes aes2 = new Aes("SGc0bTNfMm8yMF9XZWVLMg==",
"MFB1T2g5SWxYMDU0SWN0cw==");
        string text2 =
aes.DecryptFromBase64String("mjdRqH4d1O8nbUYJk+wVu3AeE7ZtE9rtT/
8BA8J897I=");
        if (text2.Equals("Same_ciphertext_"))
        {
            byte[] array = new byte[16];
            Array.Copy(aes2.EncryptToByte(text2 + str), 16, array,
0, 16);
            if
(Convert.ToBase64String(array).Equals("dJntSWSPWbWocAq4yjBP5Q=="))
            {
                MessageBox.Show("注册成功！");
                this.Text = "已激活，欢迎使用！";
                this.status = 1;
            }
            else
            {
                MessageBox.Show("注册失败！\nhint：" +
aes2.DecryptFromBase64String("mjdRqH4d1O8nbUYJk+wVu3AeE7ZtE9rtT/
8BA8J897I="));
            }
        }
        else
        {
            MessageBox.Show("注册失败！\nhint：" +
aes2.DecryptFromBase64String("mjdRqH4d1O8nbUYJk+wVu3AeE7ZtE9rtT/
8BA8J897I="));
        }
    }
    catch
    {
        MessageBox.Show("注册失败！");
    }
```

```
}
```

可见又是将输入分为两半，第一部分作为初始向量，使加密后的mjdRqH4d1O8nbUYJk+wVu3AeE7ZtE9rtT/8BA8J897I=变为Same_ciphertext_

第二部分将Same_ciphertext_和第二部分相加后加密为byte后将后十六位转为base64等于dJntSWSPWbWocAq4yjBP5Q==

就第一部分，已经得到了明文，密文，和密钥，根据aes加密的方式，只要将明文作为向量对密文进行解密就可得到真实的初始向量，

第二部分，因为只给了后半部分base64，将其转为十六进制，为了知道前面的部分，我以Same_ciphertext_123456789012345的格式进行加密，从而得到了其半部分的十六进制，拼在一起后转为base64，再进行aes解密，得到第二部分

拼起来得到flag

```
hgame{L1R5WFl6UG5ZOyQpXHdlXw==DiFfer3Nt_w0r1d}
```