攻防世界 - re-for-50-plz-50 - writeup

原创

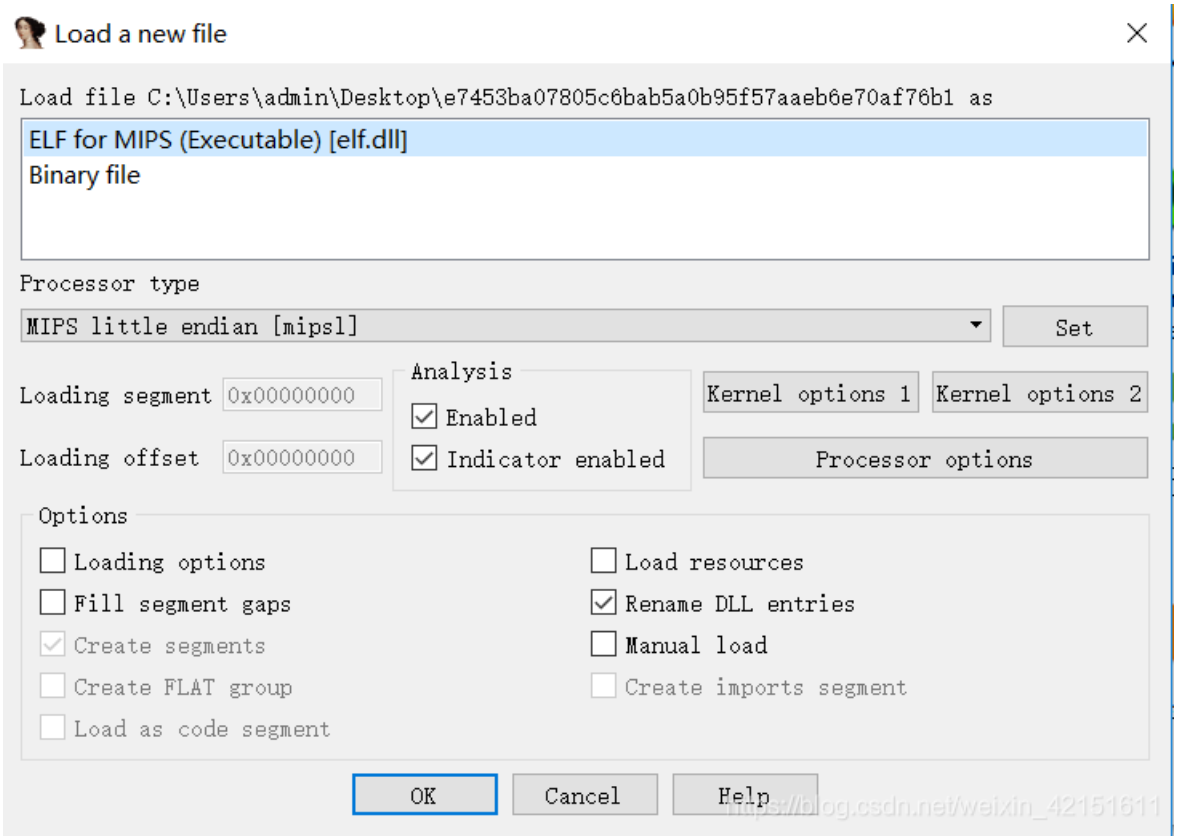
哒君 于 2019-05-29 21:52:58 发布 1048 收藏 1

分类专栏： CTF 文章标签： Reverse mips 汇编 反汇编

re-for-50-plz-50

> 文件链接 -> Github

初步分析



ELF for MIPS，第一次见，特此记录一下

知识梳理

关于Mips指令，参见 这篇博客

算法分析

主要代码段如下：

```
loc_4013C8:
lui     $v0, 0x4A # 加载0x4A到$v0寄存器的高16位
addiu   $v1, $v0, (meow - 0x4A0000)   # "cbtcqLUBChERV[[Nh@_X^D]X_YPV[CJ"
#此刻$v1寄存器的内容就是字符串的偏移量
lw      $v0, 0x28+var_10($fp)  #从$fp寄存器的0x28+var_10偏移处取出一个字（4字节）放入$v0寄存器，这里取出的是循环的i
addu    $v0, $v1, $v0  #$v0=$v1+$v0，此刻$v0中的内容就是字符串第1个字符的偏移量
lb      $v1, 0($v0)  #从$v0的内容+0的偏移量的内存中取出一字节放入$v1
lw      $v0, 0x28+arg_4($fp)  #从$fp的0x28+arg_4偏移处取出一个字放入$v0
addiu   $v0, 4
lw      $a0, 0($v0)  #取出输入的内容的偏移量
lw      $v0, 0x28+var_10($fp)  #取出循环的i
addu    $v0, $a0, $v0  #得到输入的内容的第1个字符的偏移量
lb      $v0, 0($v0)  #从输入的内容中取出一个字节放入$v0
xori    $v0, 0x37  #将其与0x37异或，存入$v0
sll     $v0, 24  #逻辑左移24位
sra     $v0, 24  #算术右移24位
#这里大概是在取得纯净的数据吧，总共32位的寄存器，左移右移24位，留下来的就是最低的8位即1字节
beq     $v1, $v0, loc_401428  #如果$v1与$v0的内容相等则跳转
move    $at, $at
```

求解

```python
a = "cbtcqLUBChERV[[Nh@_X^D]X_YPV[CJ"
f = ''
for i in range(len(a)):
    f += chr(ord(a[i])^0x37)
print(f)
```

这道题帮我了解了一下mips的汇编，感觉可读性不是那么高……