XCTF babymips

原创

pipixia233333 于 2019-05-10 16:55:48 发布 810 收藏
分类专栏： 逆向之旅

这个题 一看就有点不对劲。

拖入ida 里面这个指令有感觉有点问题。

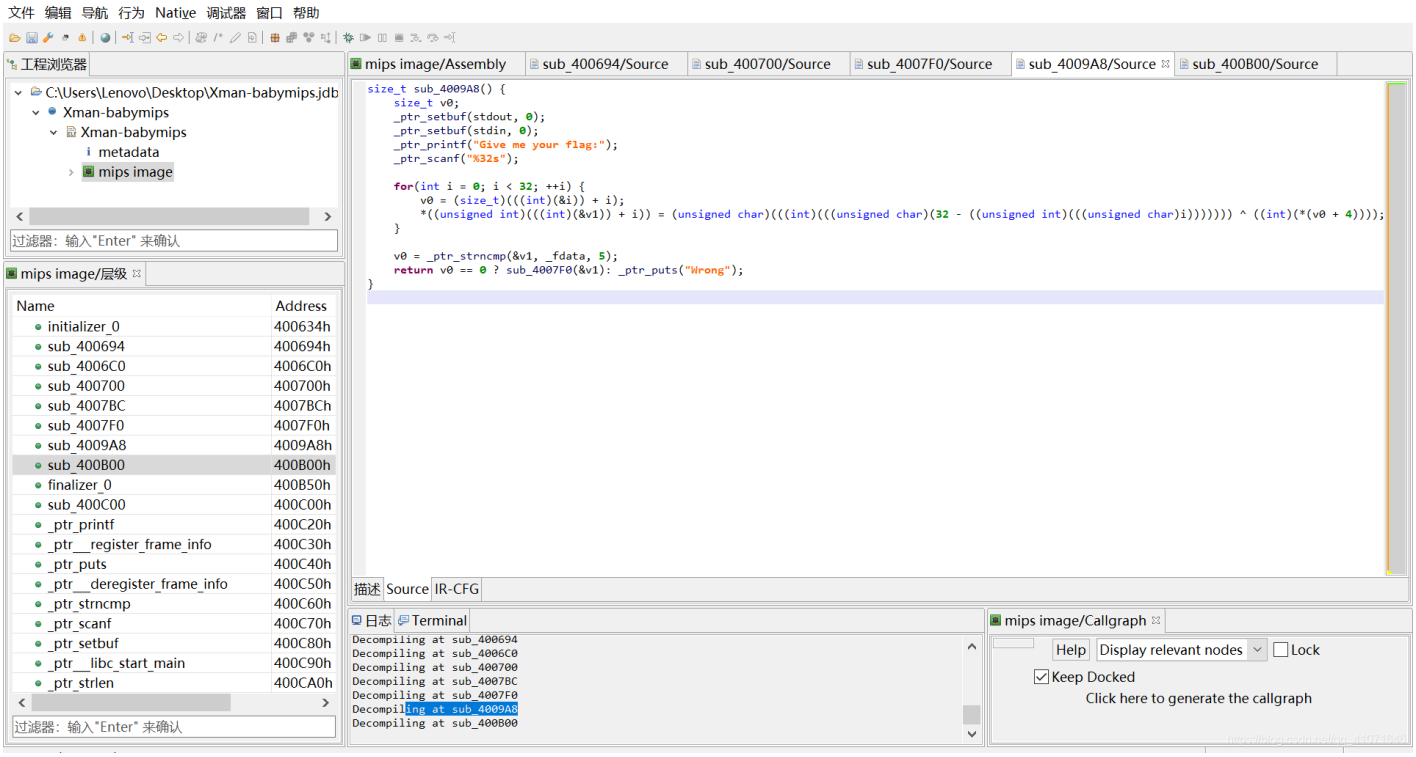
```
                li      $v0, 5
                sw      $v0, 0x28+var_10($fp)
                b       loc_400910
                nop
-------------------------------------------------------------------------------

loc_400814:                             # CODE XREF: sub_4007F0+13C↓j
                lw      $v0, 0x28+var_10($fp)
                nop
                andi    $v0, 1
                beqz    $v0, loc_400898
                nop
                lw      $v0, 0x28+var_10($fp)
                lw      $v1, 0x28+arg_0($fp)
                nop
                addu    $v0, $v1, $v0
                lb      $v0, 0($v0)
                nop
                sra     $v0, 2
                sll     $a0, $v0, 24
                sra     $a0, 24
                lw      $v0, 0x28+var_10($fp)
                lw      $v1, 0x28+arg_0($fp)
                nop
                addu    $v0, $v1, $v0
                lb      $v0, 0($v0)
                nop
                sll     $v0, 6
                sll     $v1, $v0, 24
                sra     $v1, 24
                lw      $v0, 0x28+var_10($fp)
                lw      $a1, 0x28+arg_0($fp)
                nop
                addu    $v0, $a1, $v0
```


这是啥。。。。 后来 看了一下官方的题解 说是另一种 架构 mips

什么鬼 然后说是ida 有一个插件 但是我没有搞定。。。 要是有大佬搞定了 还麻烦指教一下

然后我们就下载另一个插件 jeb（这玩意还挺难搞定的）



搞成之后 看起来就舒服很多了

不过 还是要吐槽这个 东西确实不是很好用。（或许我没有get到这个工具的正常使用方法把）

我去看 字符串什么的还要去ida 里面看

然后分析一下他这里的算法

上面的算法很好看的出来 就是

```
char s[32];
scanf("%s",s);
for(int i=0;i<32;i++)
{
    s[i]^=(32-i);
}
```

前五位 和 Qljfg 这个做比较

然后去看下一个函数

```
size_t sub_4007F0(char* __s) {
    char* __s1 = __s;
    unsigned int min = 5;
    size_t i;

    for(i = _ptr_strlen(__s); i > min; i = _ptr_strlen(__s1)) {

        if((min & 1) != 0) {
            i = (size_t)(min + ((int)__s1));
            __s = (int)(((unsigned char)(((int)(*i)) / 4)));
            i = (size_t)(min + ((int)__s1));
            __s1[min] = (unsigned char)(((unsigned int)(((((int)(*i)) * 1073741824) >> 24) | ((int)__s))));
        }
        else {
            i = (size_t)(min + ((int)__s1));
            __s = (((int)(*i)) * 67108864) >> 24;
            i = (size_t)(min + ((int)__s1));
            __s1[min] = (unsigned char)(((unsigned int)(((int)(((unsigned char)(((int)(*i)) / 64)))) | ((int)__s))));
        }

        ++min;
    }

    i = _ptr_strncmp(__s1 + 5, gvar_410D04, 27);
    return i == 0 ? _ptr_puts("Right!"): _ptr_puts("Wrong!");
}
```

描述 Source IR-CFG

这个函数貌似难懂一点

仔细看看 也不算是很很难吧~

这里的 1073741824 >>24 其实就是 *64 也就是<<6 那么 67108864>>24 也就是 /4 也就是 <<2

那么 我们就很明了了

写出脚本即可

```
#include <stdio.h>
#include<iostream>
#include<iomanip>
#include<stdio.h>
#include<string.h>
#include<algorithm>
#include<vector>
#include<iostream>
#include<map>
#include<time.h>
#include<queue>
#include <Windows.h>
#include "windows.h"
using namespace std;
char s[]="Q|j{g";
unsigned char ida_chars[] =
{  0x51, 0x7C, 0x6A, 0x7B, 0x67,
  0x52, 0xFD, 0x16, 0xA4, 0x89, 0xBD, 0x92, 0x80, 0x13, 0x41,
  0x54, 0xA0, 0x8D, 0x45, 0x18, 0x81, 0xDE, 0xFC, 0x95, 0xF0,
  0x16, 0x79, 0x1A, 0x15, 0x5B, 0x75, 0x1F, 0x00
};
int main()
{
    for(int i=5;i<32;i++)
    {
        if((i&1)!=0)
        {
            ida_chars[i]=(ida_chars[i]>>6)|(ida_chars[i]<<2);
        }
        else
        {
            ida_chars[i]=(ida_chars[i]>>2)|(ida_chars[i]<<6);
        }
    }
    for(int i=0;i<32;i++)
    {
        ida_chars[i]^=(32-i);
    }
    printf("%s\n",ida_chars);
    return 0;
}
```