buuctf

原创

wgjfxj99 于 2022-02-19 21:00:57 发布 353 收藏
文章标签： python

本文链接：https://blog.csdn.net/wgjfxj99/article/details/123023830

java逆向解密

下载文件是一个class文件，上网查了以后下载jd-gui打开

```
public static void main(String[] args)
{
  Scanner s = new Scanner(System.in);
  System.out.println("Please input the flag : ");
  String str = s.next();
  System.out.println("Your input is : ");
  System.out.println(str);
  char[] stringArr = str.toCharArray();
  Encrypt(stringArr);
}

public static void Encrypt(char[] arr)
{
  ArrayList<Integer> Resultlist = new ArrayList();
  for (int i = 0; i < arr.length; i++)
  {
    int result = arr[i] + '@' ^ 0x20;
    Resultlist.add(Integer.valueOf(result));
  }
  int[] KEY = { 180, 136, 137, 147, 191, 137, 147, 191, 148, 136, 133
  ArrayList<Integer> KEYList = new ArrayList();
  for (int j = 0; j < KEY.length; j++) {
    KEYList.add(Integer.valueOf(KEY[j]));
  }
  System.out.println("Result:");
  if (Resultlist.equals(KEYList)) {
    System.out.println("Congratulations! ");
```


这是Python的代码

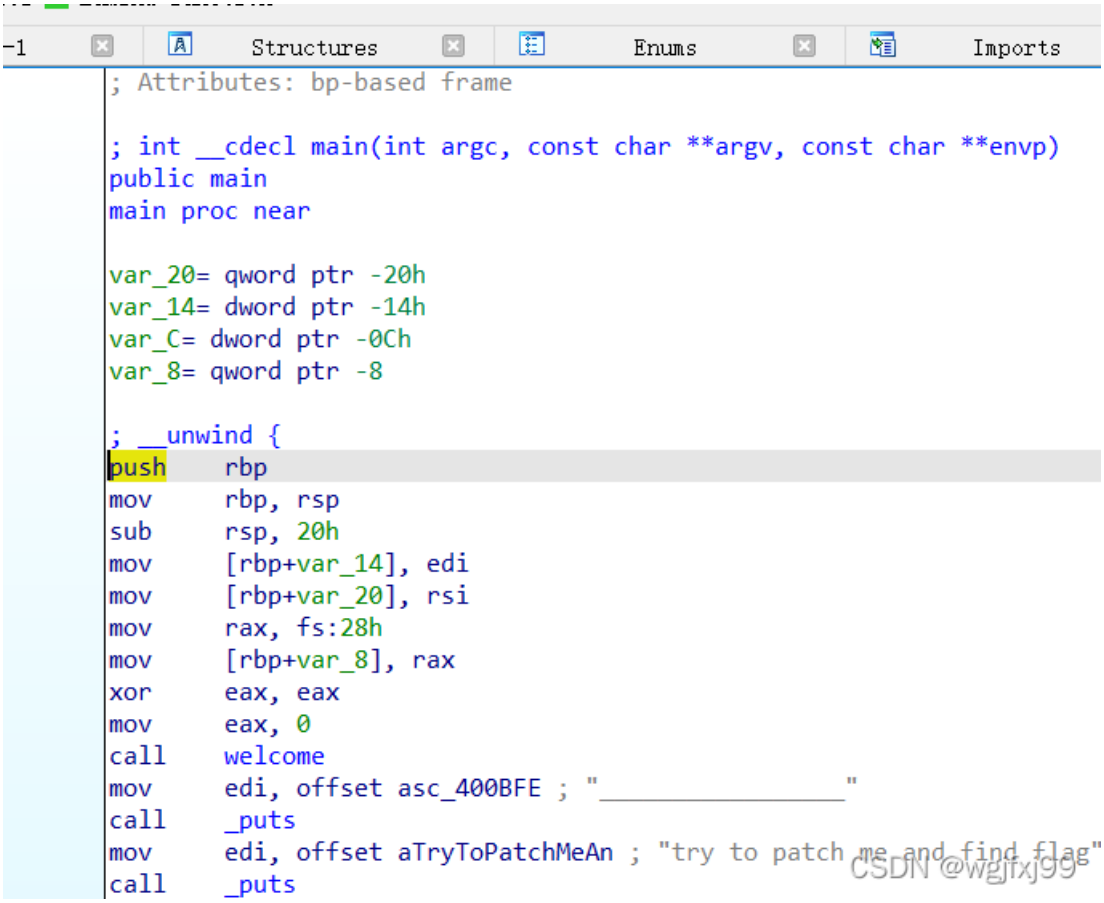
我们输入一个字符串 然后经过一个for循环进行异或 然后将得到的新字符串与KEY进行比较，看看是否相等

结果就得到了flag：flag{This_is_the_flag_!}

luck—guy

查看信息后用ida64位打开

找到main函数



反汇编并点到那个flag里

```
-1   Structures   Enums   Imports
; Attributes: bp-based frame

; int __cdecl main(int argc, const char **argv, const char **envp)
public main
main proc near

var_20= qword ptr -20h
var_14= dword ptr -14h
var_C= dword ptr -0Ch
var_8= qword ptr -8

; __unwind {
push    rbp
mov     rbp, rsp
sub     rsp, 20h
mov     [rbp+var_14], edi
mov     [rbp+var_20], rsi
mov     rax, fs:28h
mov     [rbp+var_8], rax
xor     eax, eax
mov     eax, 0
call    welcome
mov     edi, offset asc_400BFE ; "_____________________"
call    _puts
mov     edi, offset aTryToPatchMeAn ; "try to patch me and find flag"
call    _puts
```



```
ction  Data  Unexplored  External symbol  Lumina fur
IDA View-A   Pseudocode-A   Strings windo
 1 int __cdecl main(int argc, const char **argv, cc
 2 {
 3   unsigned int v4; // [rsp+14h] [rbp-Ch] BYREF
 4   unsigned __int64 v5; // [rsp+18h] [rbp-8h]
 5
 6   v5 = __readfsqword(0x28u);
 7   welcome(argc, argv, envp);
 8   puts("_____________________");
 9   puts("try to patch me and find flag");
10   v4 = 0;
11   puts("please input a lucky number");
12   __isoc99_scanf("%d", &v4);
13   patch_me(v4);
14   puts("OK,see you again");
15   return 0;
16 }
```

```
unsigned __int64 get_flag()
{
  unsigned int v0; // eax
  int i; // [rsp+4h] [rbp-3Ch]
  int j; // [rsp+8h] [rbp-38h]
  __int64 s; // [rsp+10h] [rbp-30h] BYREF
  char v5; // [rsp+18h] [rbp-28h]
  unsigned __int64 v6; // [rsp+38h] [rbp-8h]

  v6 = __readfsqword(0x28u);
  v0 = time(0LL);
  srand(v0);
  for ( i = 0; i <= 4; ++i )
  {
    switch ( rand() % 200 )
    {
      case 1:
        puts("OK, it's flag:");
        memset(&s, 0, 0x28uLL);
        strcat((char *)&s, f1);
        strcat((char *)&s, &f2);
        printf("%s", (const char *)&s);
        break;
      case 2:
        printf("Solar not like you");
        break;
      case 3:
        printf("Solar want a girlfriend");
        break;
      case 4:
        s = 0x7F666F6067756369LL;
        v5 = 0;
        printf("%s", (const char *)&s);
        break;
      case 2:
        printf("Solar not like you");
        break;
      case 3:
        printf("Solar want a girlfriend");
        break;
      case 4:
        s = 0x7F666F6067756369LL;
        v5 = 0;
        strcat(&f2, (const char *)&s);
        break;
      case 5:
        for ( j = 0; j <= 7; ++j )
        {
          if ( j % 2 == 1 )
            *(&f2 + j) -= 2;
          else
            --*(&f2 + j);
        }
        break;
      default:
        puts("emmm,you can't find flag 23333");
        break;
    }
  }
  return __readfsqword(0x28u) ^ v6;
}
```

先case4赋值，然后case5，进行加密操作，最后case1，进行剪切

用脚本得到flag