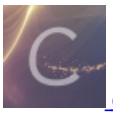
XCTF-Reverse-ExerciseArea-007-writeup

原创

y4ung 于 2019-08-03 08:52:06 发布 4126 收藏

分类专栏： ctf 文章标签： ctf ReverseEngineering



0x00 介绍

本题是xctf攻防世界中Reverse的新手第七题。题目来源： `9447 CTF 2014`

需要对该二进制文件insanity进行逆向分析，找到flag

实验环境：IDA Pro 7.0，gdb

0x01 解题过程

1.1 文件分析

1. 在Vscode中安装插件： `hexdump for VSCode` ，用Vscode打开，显示文件的十六进制：

可以看到文件的开头有 `ELF` ，说明这是一个在Linux下的可执行文件；

2. 在kali中用 file 命令，可以看到这是一个32bit的系统中编译的文件，同时可以看到该文件编译后符号表没有被strip掉

```
root@kali:~/hzy/ctf-learning# file insanity
insanity: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linu
x.so.2, for GNU/Linux 2.6.26, BuildID[sha1]=5b8ef7c72fce77481f4edd6802bbdb7c6100dc6e, not stripped
```

3. 修改文件权限为可执行，运行该文件。先输出字符串 `Reticulating splines, please wait..` ，然后暂停了几秒，再输出后面的字符串，然后就没了，也没让我输入什么字符串？？？一脸懵逼，而且每次运行输出还不一样。还是直接分析吧

```
root@kali:~/hzy/ctf-learning# ./insanity
Reticulating splines, please wait..
There aren't enough bits in my memory to represent how hard you fail.
root@kali:~/hzy/ctf-learning# ./insanity
Reticulating splines, please wait..
I've got a good feeling about this one..... wait no. Maybe next time.
root@kali:~/hzy/ctf-learning#
```

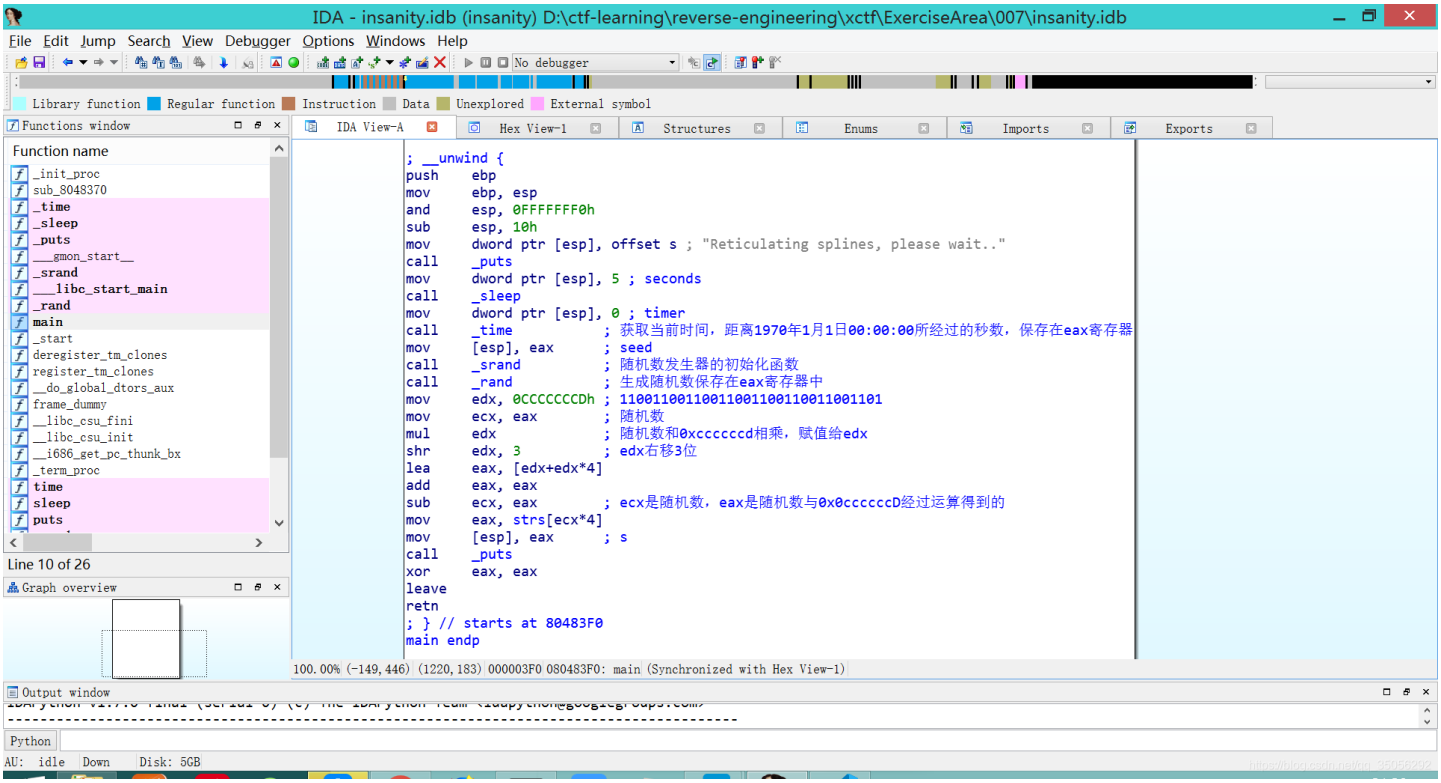
1.2 脱壳

用IDA打开，发现该二进制文件未被加壳，因此不需要进行脱壳操作

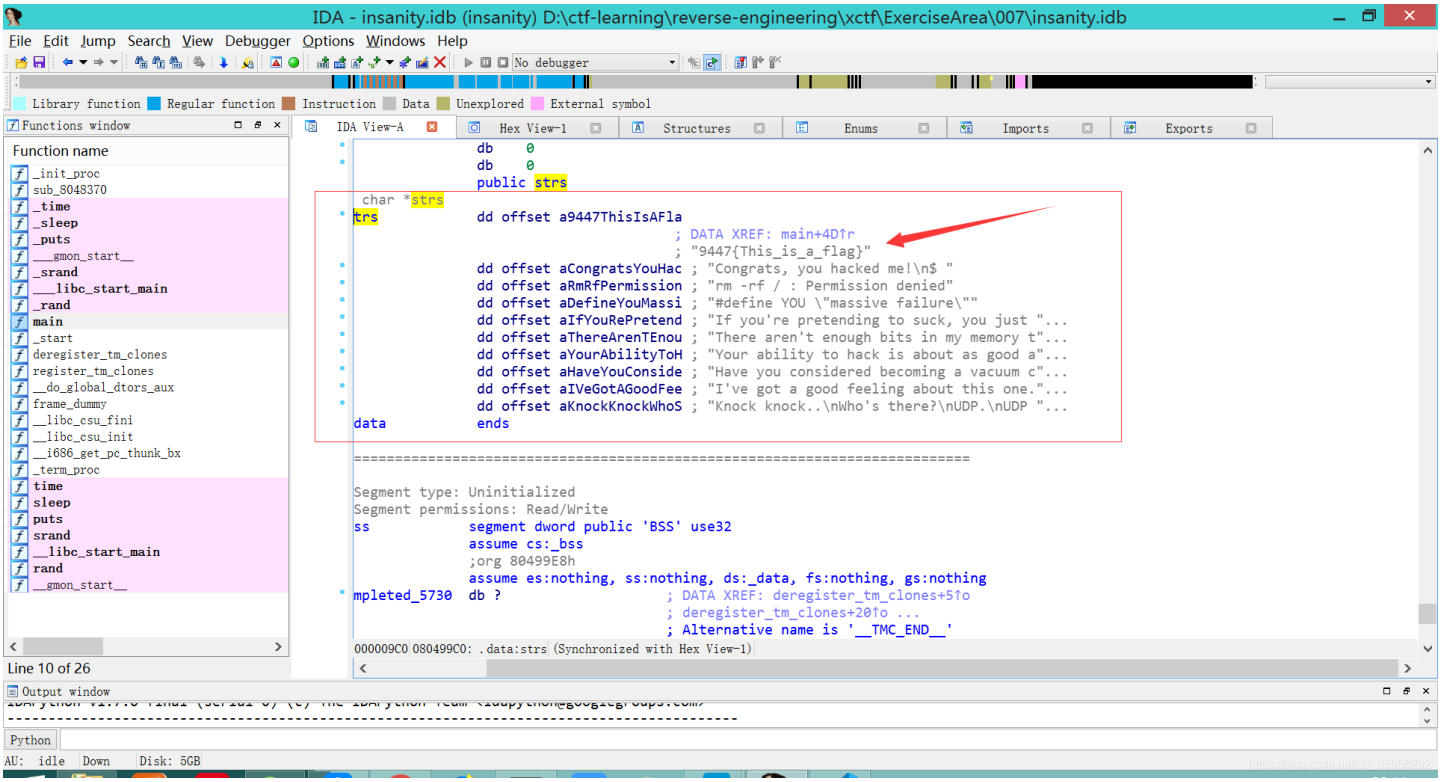
1.3 逆向分析

由于该文件的符号表未被去掉，因此直接用命令`b main`，在main函数处打断点进行调试

main函数中，通过获取当前时间作为随机数种子，生成随机数，然后根据随机数与0x0ccccccD进行相关运算，得到的结果作为字符串数组strs的索引。因此随着时间的变化，每次打印的字符串也是不一样的。



3. 在IDA中跳转到strs字符串数组查看，发现flag... 猜测只要试的次数多，就能打印出flag？ 于是此题结束



flag为：`9447{This_is_a_flag}`