<https://adworld.xctf.org.cn> 的第一道pwn题

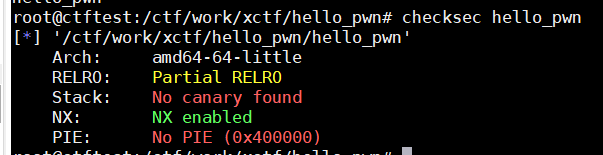


## Pwndocker

安装pwndocker并启动，过程略，自行百度

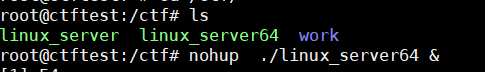
## 2、checksec

使用checksec命令查看为64位，开启了NX防护

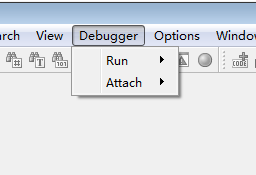


## 3、IDA

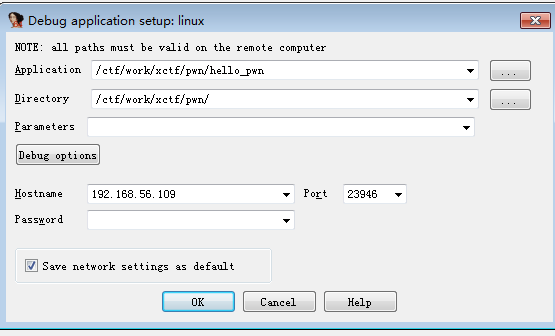
Linux上面启动linux\_server64远程调试服务



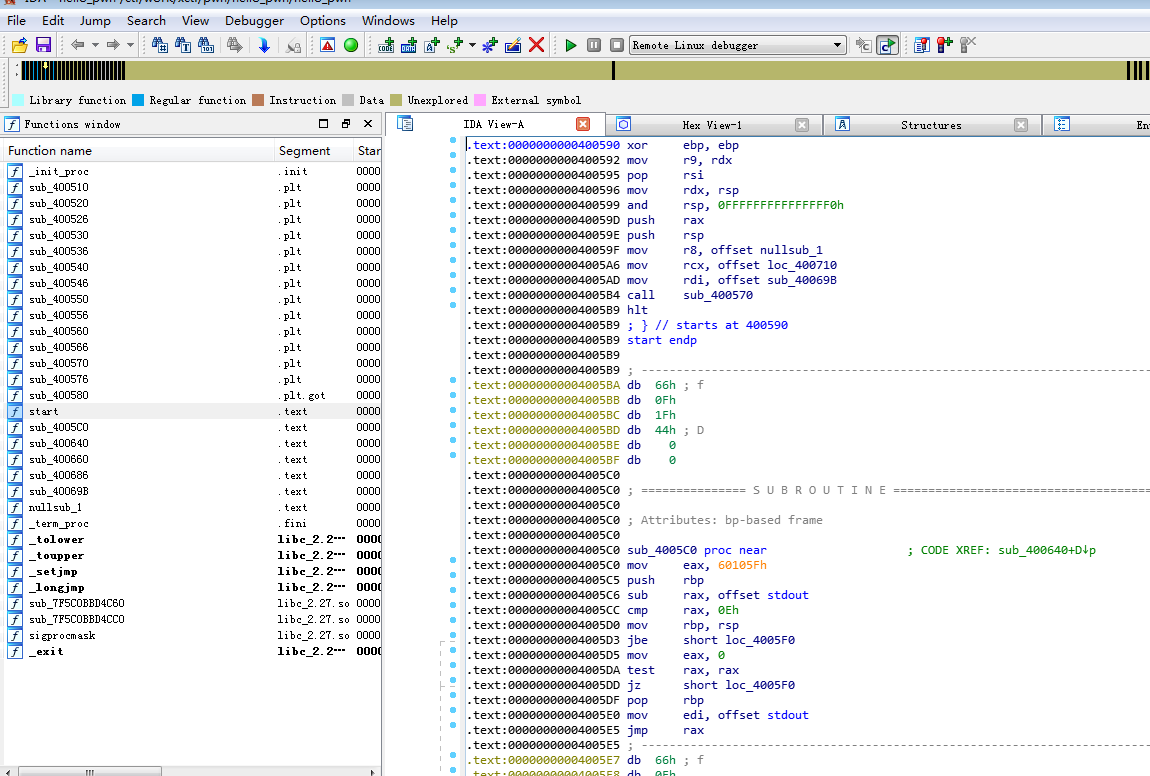
点IDA Debugger -> Run ->Remote Linux Debugger



填写如下



得代码如下：

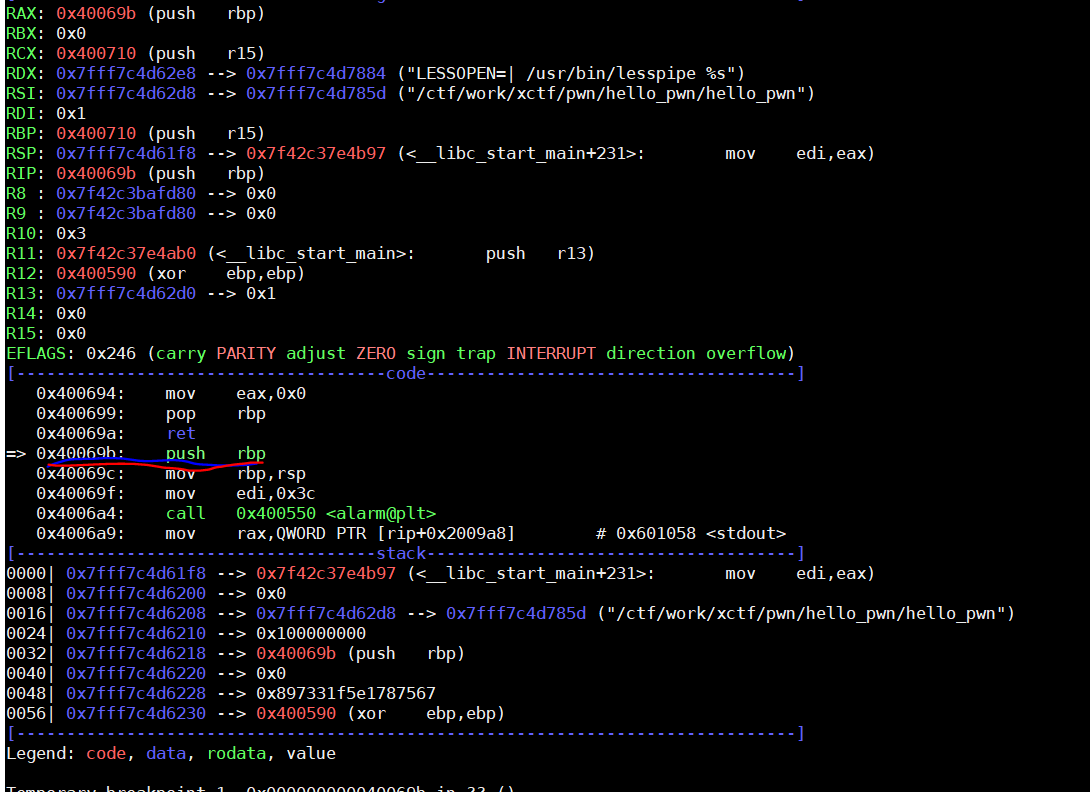


## 4、GDB+peda

GDB安装peda插件，自己百度

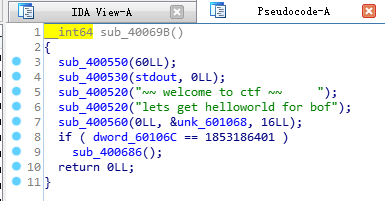
gdb hello\_pwn

然后输入start，即得到调试界面

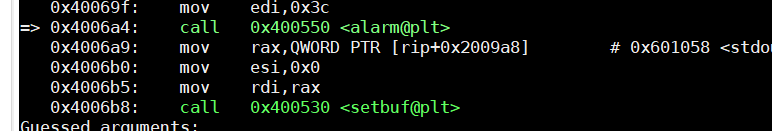


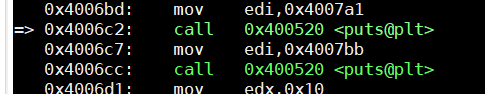
## 5、gdb和IDA共用

gdb加载后自动停在0x40069b这个地址，说明这个是起始地址，将IDA定位到该地址，然后按F5，得到伪代码，但是里面的函数名都是以地址开头的不好看，需要修正

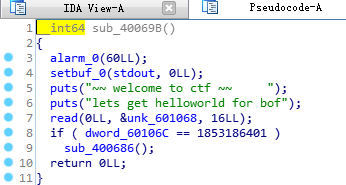


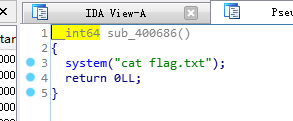
其中的一种方法是用gdb调试修改，可以看到函数真实名字





然后修改IDA函数名





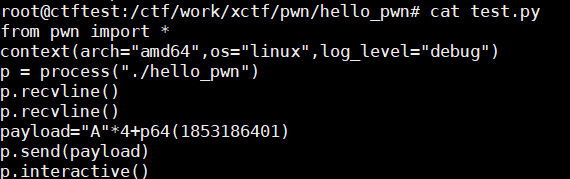
代码逻辑很简单，只要0x60106c里的内容为1853186401则可以执行system(“cat flag.txt”)极获取到flag

那么如何让0x60106c里面的值为1853186401（0x6E756161‬）呢

我们可以通过read那一行函数覆盖，它可以接收输入，然后覆盖从0x601068开始的16给字节，只要我们输入如下内容，即可保证0x60106c中的内容为0x6E756161（XX代表非00的任意字符，--代表任意字符）

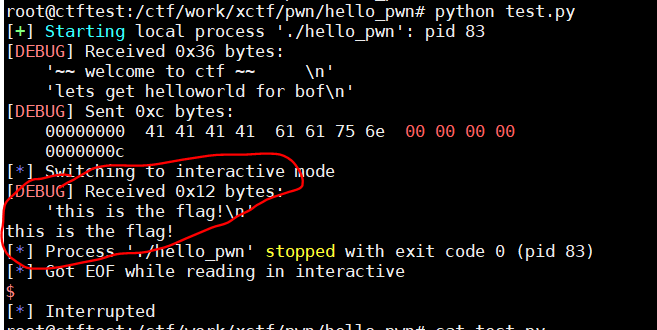
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0x601068 | | | | 0x60106c | | | | | | | | | | | |
| XX | XX | XX | XX | 61 | 61 | 75 | 6E | 00 | 00 | 00 | 00 | -- | -- | -- | -- |

## 6、Pwntools



主要是payload=”A”\*4+p64(1853186401)这一行，原因见上面的内容，p64函数自动处理大小端

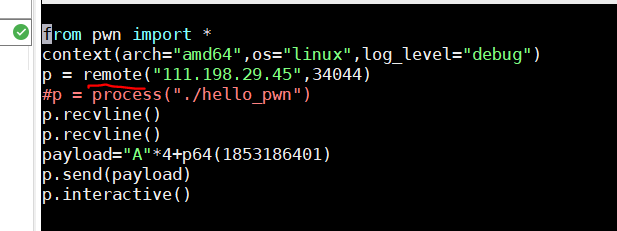
提前在本地建flag.txt，然后python执行，可获得flag



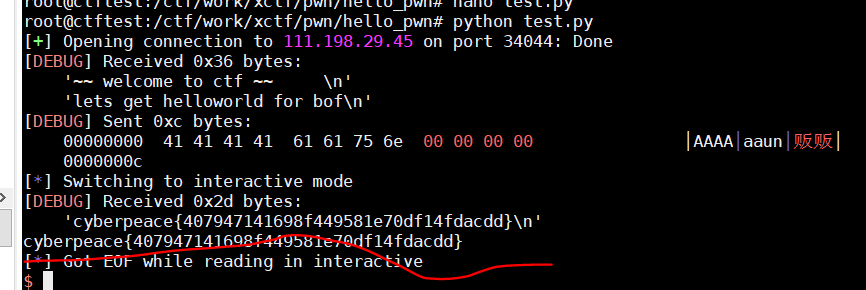
如果是在线题



把process函数换成remote函数



获取到flag



## 总结

注意：其实上面不是所有步骤都是必须的，只是为了演示工具的使用，单用一个gdb也可以做题