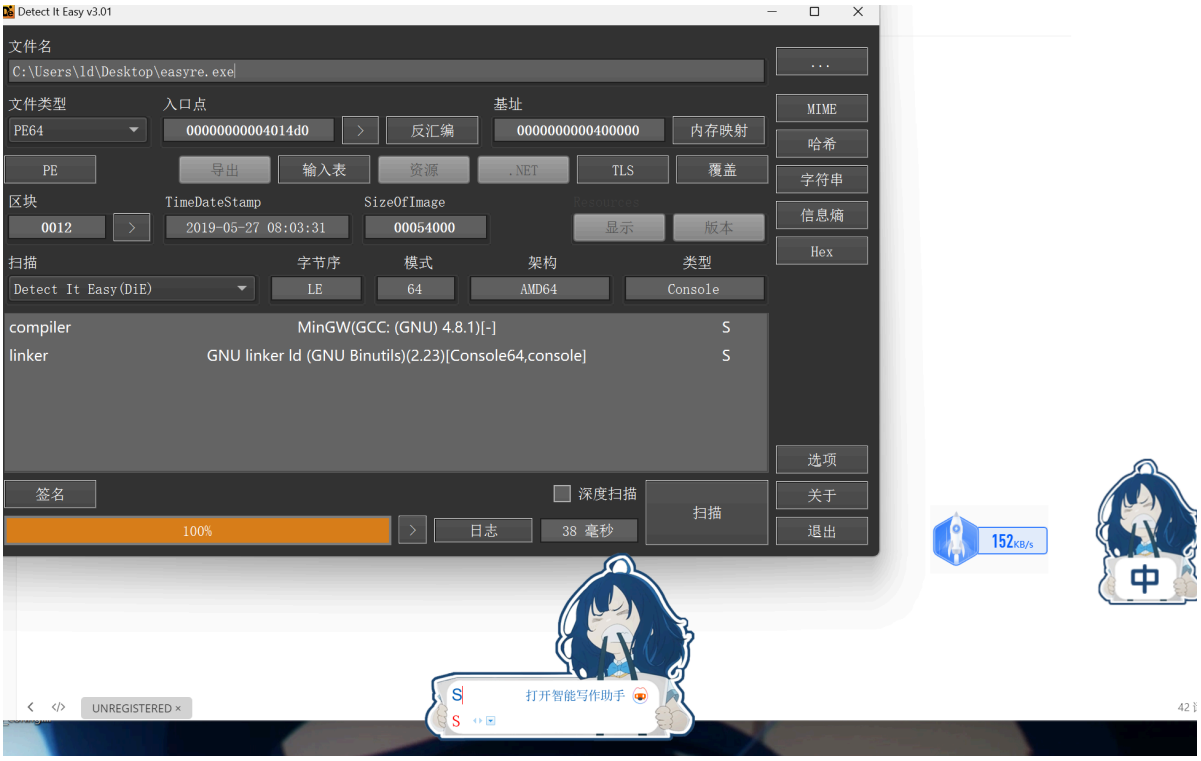
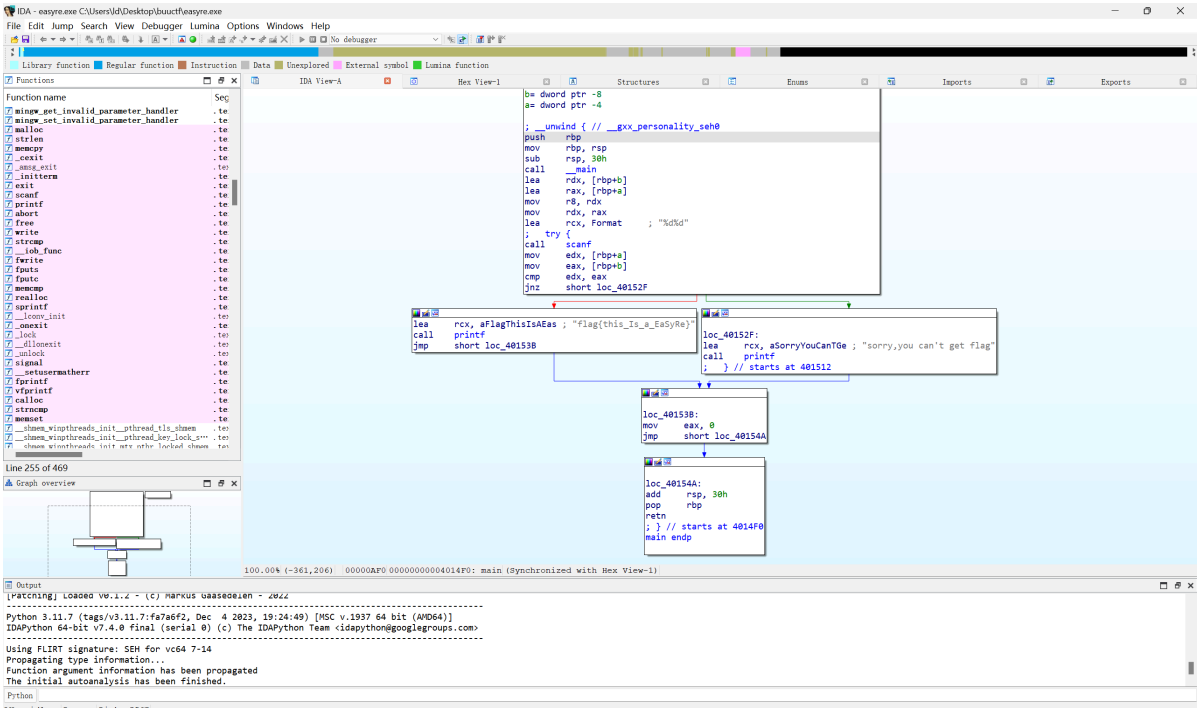
使用upx shell工具加壳

首先，我们随便在buuctf搞个题拿到一个二进制文件

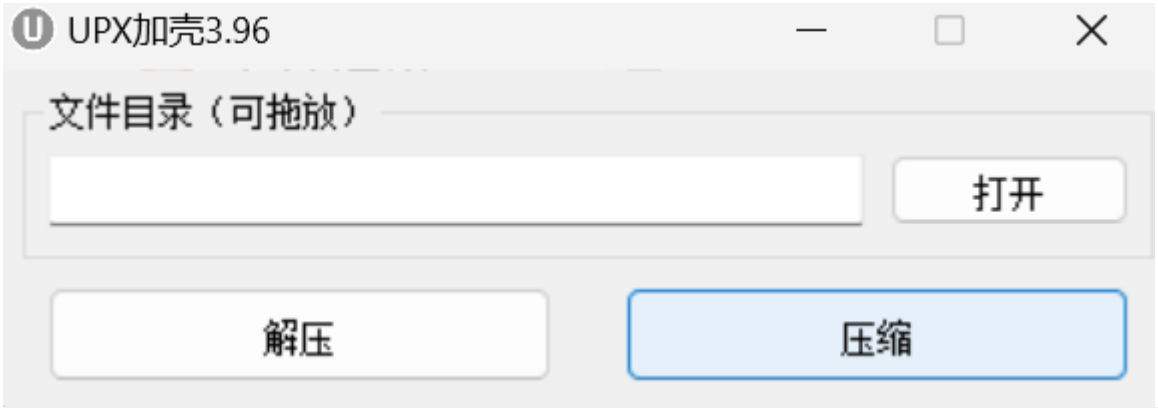
然后丢到DIE里面看看有没有壳

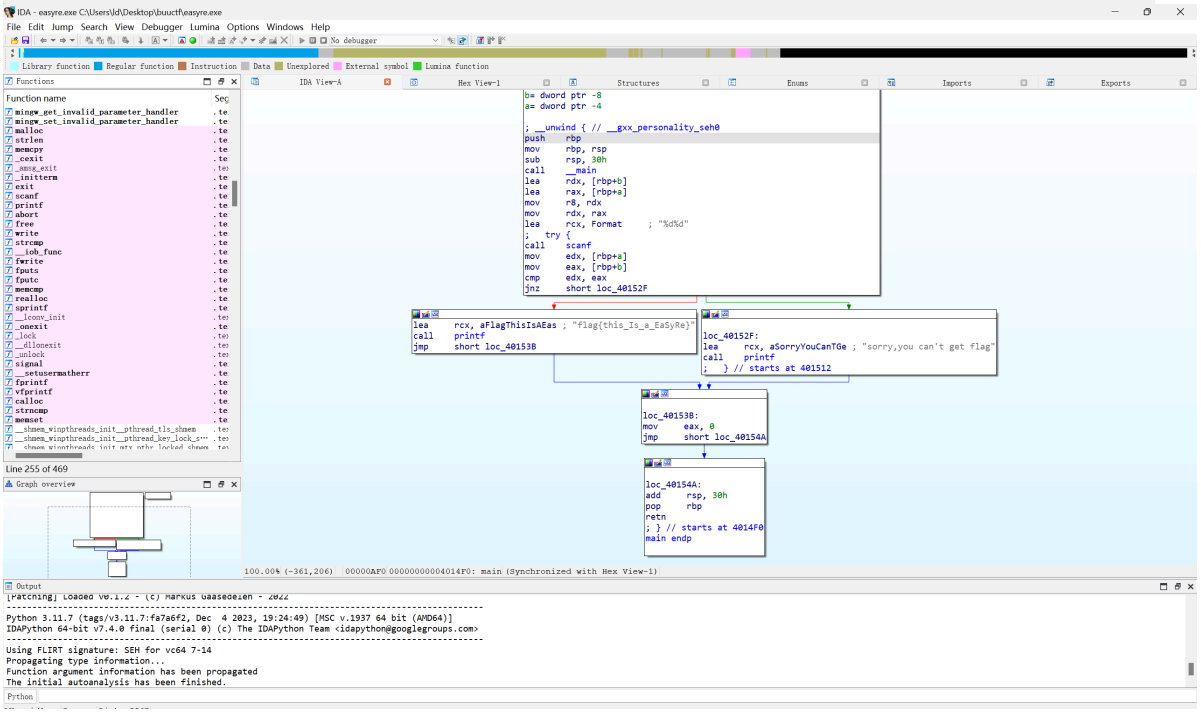


再丢进ida中看看



发现没壳，符合我们的需求，接下来我们就准备用upx shell加壳





附：后缀名为bak的文件是备份文件，你修改了原文件的内容后，保存了修改后的内容，那么修改前的内容会自动保存为后缀名为bak的备份文件（前提是设置为保留备份），如果你想查看或者恢复修改前的内容，就可以打开这个bak文件

这里补充一下upx加壳的原理：

upx的功能有两种描述。一种叫做给程序加壳，另一种叫压缩程序。其实这两种表述都是正确的，只是从不同的
角度 对upx的描述。
upx的工作原理其实是这样的：首先将程序压缩。所谓的压缩包括两方面，一方面在程序的开头或者其他合适的
地方 插入一段代码，另一方面是将程序的其他地方做压缩。压缩也可以叫做加密，因为压缩后的程序比较难看
懂，主要是 和原来的代码有很大的不同。最大的表现也就是他的主要作用就是程序本身变小了。变小之后的程
序在传输方面有很 大的优势。其次就是在程序执行时，实时的对程序解压缩。解压缩功能是在第一步时插入的
代码完成的功能。联起来 就是：upx可以完成代码的压缩和实时解压执行。且不会影响程序的执行效率。
upx和普通的压缩，解压不同点就算在于upx是实时解压缩的。实时解压的原理可以使用一下图形表示：
1==>2==>3==>4==>5==>6
假设1是upx插入的代码，2，3，4是压缩后的代码。5，6是随便的什么东西。
程序从1开始执行。而1的功能是将2，3，4解压缩为7，8，9。7，8，9就是2，3，4在压缩之前的形

式。
1==>7==>8==>9==>5==>6

连起来就是：

最初代码的形式就应该是：7==>8==>9==>5==>6
用upx压缩之后形式为：1==>2==>3==>4==>5==>6
执行时的形式变为：1==>7==>8==>9==>5==>6

类似的技术还有很多。这样的技术较多的应用于：木马和病毒躲避杀毒软件时，发布的程序防止被反编译或
破解时。 upx是一种典型的加壳程序或者压缩程序。因此已经有非常成熟的去壳程序或者解压缩程序。同时，很多的杀毒软件 也可以识别出加有upx壳的病毒和木马。而一些软件生产场上为了防止被破解所加的
壳也同时被轻易的破解。在这样 的情况下很多人想出了产生自己的加壳程序的想法。应此很多人都通过改
编一些成熟的加壳程序来产生自己的加壳程 序。 其实改编upx也是很简单的。因为upx的源代码是公开的
，所以可以下载它的源代码来瞧瞧。upx可以压缩很多种类型 的可执行文件。因此如果自己只是想要压缩exe
程序，则只需要阅读与压缩exe文件相关的内容。这些代码是很少的。

改写upx一般需要注意一下几点：
1、保证修改后的加壳程序不会产生upx产生的特征码。有很多软件可以通过这些特征码识别出程序是经过upx
加壳的。
2、保证加壳之后，程序仍然可以顺利执行。
3、在一定程度上保证效率不会下降太多。

`**点击压缩后即可给该可执行文件加壳**`

将加壳后的可执行文件再丢进DIE中查看,发现较加壳之前多了个packer也就是壳，观察可知加壳软件为UPX(3.96)

再将其丢进ida中进行反汇编，发现函数只有寥寥几个，自此我们可以得到一个解题经验

如果用ida打开一个可执行文件发现里面的函数非常少的话应该就是加壳了