目前使用广泛的调试器：IDE自带的调试器、Immunity Debugger、GDB以及Olly Dbg。

Immunity Debugger可以“Attach”一个进程来调试。OD，Immunity Debugger等动态调试器没有64位版，有钱就IDA pro。

软件使用验证：可可验证

使用PEID加强版说不是有效的PE文件，但是在linux下可以使用file进行识别

root@kali:~/Desktop# file babyre\(1\)

babyre(1): ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.24, BuildID[sha1]=c730f4424672fab1bd68cfb1826bdf9b6257a445, not stripped

这表示是linux下的可执行文件

壳、不同编译器编写的程序都有自己的特征，比较常见的入口地址处的特征。

Delphi也是种语言，并不是壳

易语言的程序使用PEID判断特征是VC

软件层面：

软件运行的时候应设置一个文件锁，可以避免程序被多次开启

**壳的经验：**

vmp壳 vm这个函数 里面的子函数

脱壳之后，配置文件在临时目录或者内存中

查壳软件（也可以查看出来是什么语言、编译器编写的程序）：

PEID

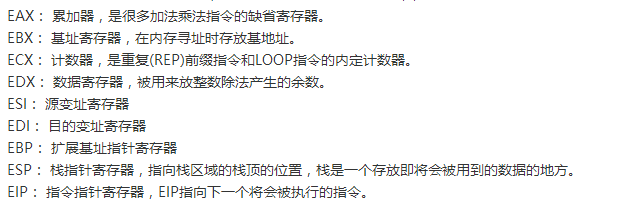
detect it easy

几种工具:

C32ASM反汇编工具)

PE Explorer 反汇编工具

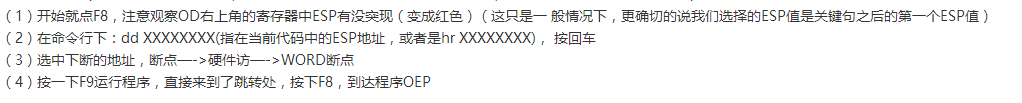
寄存器的作用：



脱壳手法：

ESP定律：

下一个ESP的硬件访问断点，从而找到OEP



PEID查到Microsoft Visual C++ 9.0 - VS 2008 \* vic4key {CiN1} \* ，这是没有壳的。

如果是使用Cygwin编译，我现在有的PEID无法识别。

反复加壳，很自然程序会崩溃。

特征码的方法可以过各种壳，特征码来源于官方提供的e语言模块，并没有编译好，所以编译使用的e语言程序版本也会有差异，搜不到很正常，总体还是搜字符串稳定些。金盾原来提供个e模块，但是里面包含一个dll，这个dll是编译好的，还有各种导出函数可以查，所以以前飘零很容易破解。但是高版本的dll很少流出。

scylla和loadpe脱壳后文件大小不一样，脱壳后都可以正常跑，这是怎么回事

有个功能叫修正镜像大小

e语言 按钮的事件特征码（一般有两个）: FF55FC5F5E

函数与dll之间的关系：（自动加载（可能因为环境变量的设置），在代码中手动加载）

MessageBox调用需要加载user32.dll

linux下查看二进制模式查看文件 hexdump，如果文件字节数为单数，那么最后为显示整齐会多出一个00字节：

