

**恶意代码分析与防治课程实验报告**

**实验九**

****

专 业 信息安全

学 号 2113662

姓 名 张丛

班 级 信安一班

1. **实验目的**

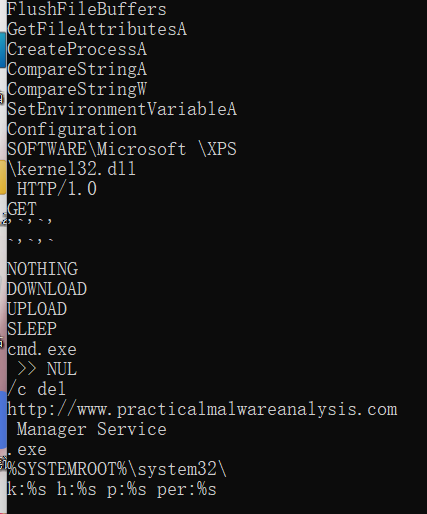
完成课本Lab 9的实验内容，编写Yara规则，并尝试IDA Python的自动化分析。

1. **实验过程**

### Lab 9-1

### 分析：

查看字符串：

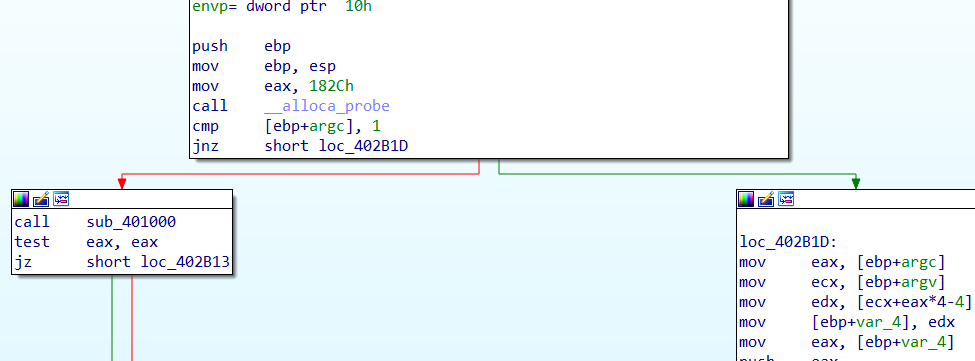


可以看到该程序对注册表和环境变量进行了一些操作，并且具有网络特征；

出现了cmd.exe可能具有远程shell的功能；

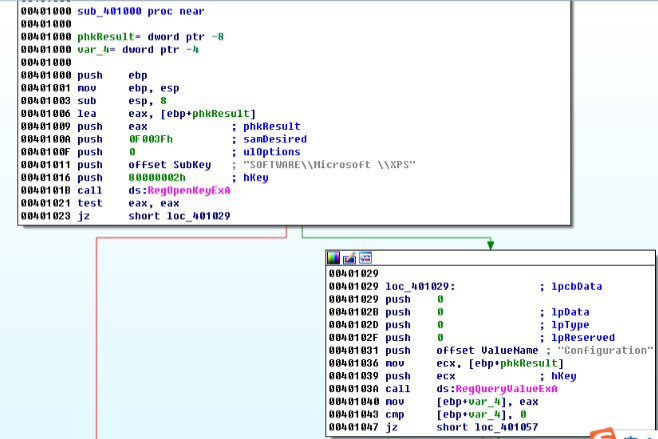
创建了文件，可能下载了什么东西。

从程序起点开始分析：



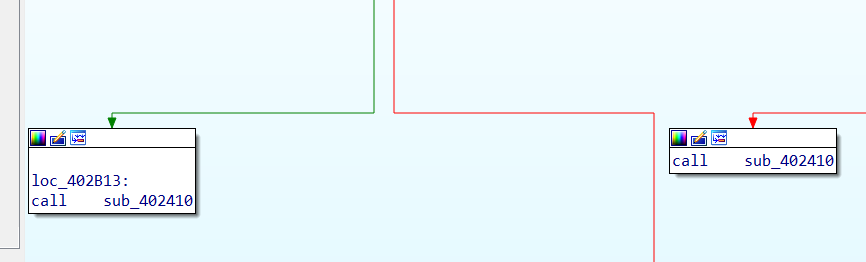
首先，在地址0x402AFD查看[命令行参数](https://so.csdn.net/so/search?q=%E5%91%BD%E4%BB%A4%E8%A1%8C%E5%8F%82%E6%95%B0&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_61823031/article/details/_blank)的数量是否等于1，等于1，ZF等于1，此时jnz无法跳转，cal了函数 0x401000。

查看此函数：

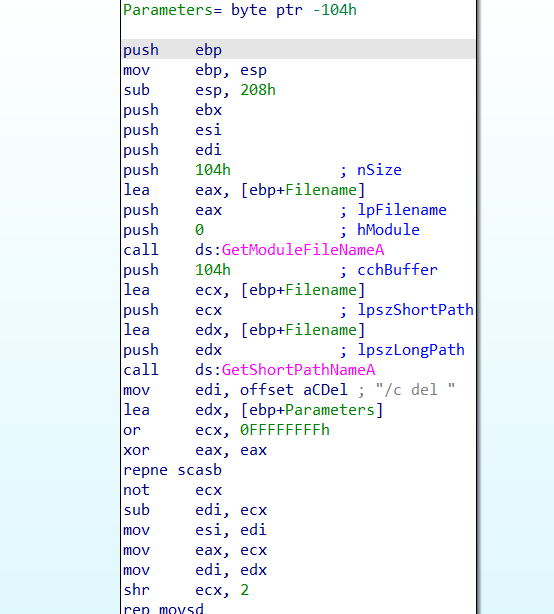


发现函数 RegOpenKeyExA ，打开指定的注册表，RegQueryValueExA 检索与打开的注册表项关联的指定值名称的类型和数据。该函数综合功能是查找是否存在它指定的注册表。

没有查找到指定的注册表，反返回了0.进入了0x402410：



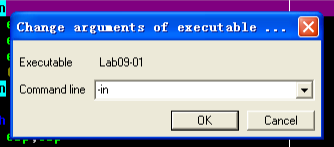
查看此函数：



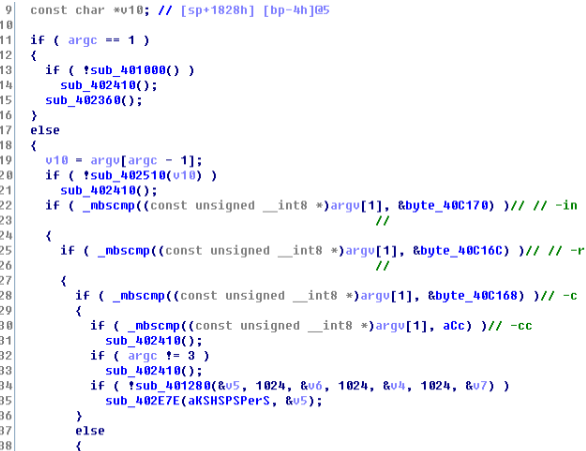
GetModuleFileName 函数获取当前可执行文件的路径，shellExeute函数对指定文件进行操作，我们看得到有“cmd.exe”“>> NUL”“/c del”字符串，可能是对自身进行了删除。

### 如何让这个恶意代码安装自身？

首先需要跳过\_\_alloca\_probe函数，通过加入一个参数跳过。

  
 还需要跳过0x402510函数，进入这个函数分析可以得知，需要参数为abcd。

### 这个恶意代码的命令行选项是什么？它要求的密码是什么？



（1）若带的参数是-in，则安装服务。



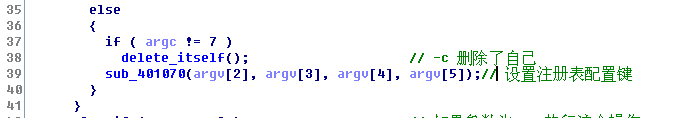
(2) 参数-re

若带的参数是-re,则卸载服务



(4) 参数-c

若带的参数是-c，则删除自身并设置注册表配置键



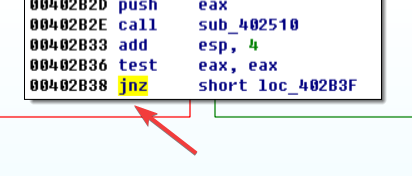
(5) 参数-cc

若带的参数是-cc，则打印注册表配置键

IMG_256

该恶意程序还能能执行多种指令 ‘-in’ 进行安装，‘-re’ 进行恶意程序的彻底清除，‘-cc’ 进行注册表打印，‘-c’ 进行恶意代码更新配置；要求的密码是 ‘abcd’。

### 如何利用OllyDbg永久修补这个恶意代码，使其不需要指定的命令行密码？



将此处的jnz改为je，既可在密码错误时跳转。

### 这个恶意代码基于系统的特征是什么？

创建一个指定注册表项 “HKLM\SOFTWARE\Microsoft \XPS” 的 “configuration”键”；

同时会创建一个服务，该服务的名称由安转时传入的参数决定； 该程序还会自我赋值到Windows系统目录下。

### 这个恶意代码通过网络命令执行了哪些不同操作？

SLEEP

UPLOAD

DOWNLOAD

NOTHING

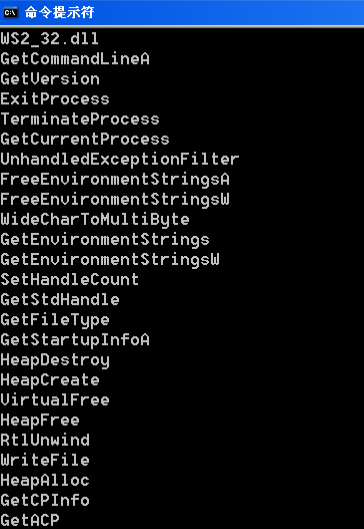
### 这个恶意代码是否有网络特征？

网址请求：http://www.practicalmalwareanalysis.com

### Lab 9-2

### 在二进制文件中，你看到的静态字符串是什么？

查看字符串：



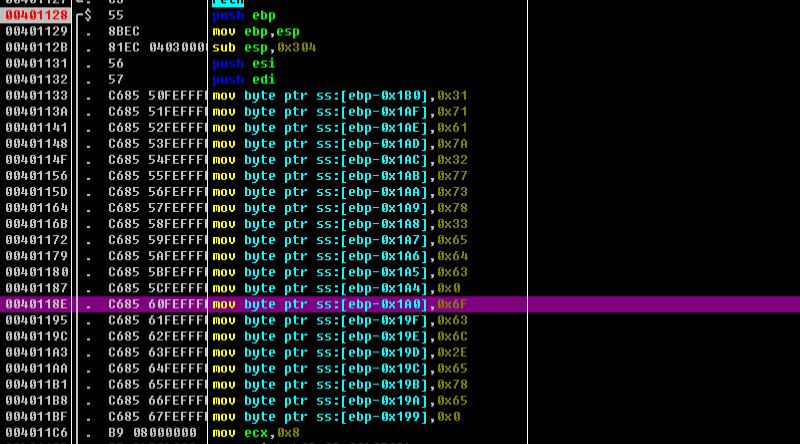
### 当你运行这个二进制文件时，会发生什么？

没有发生什么就终止了。

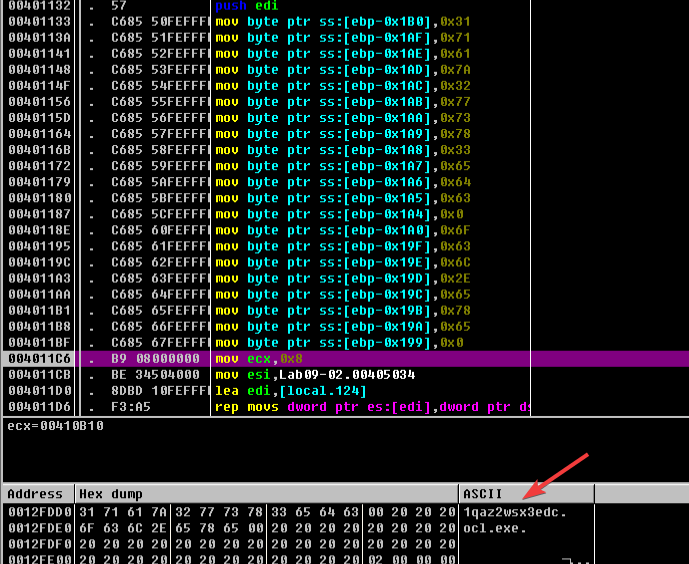
### 怎样让恶意代码的攻击负载（payload）获得运行？



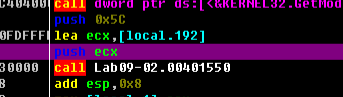
使用OllyDbg进行分析，首先定位到程序入口。



使用OllyDbg进行跟随：

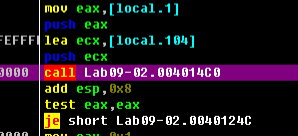


继续运行，发现调用了GetMoudleFileName函数，最终运行到这个位置：



调用了一个函数，有两个参数，其中一个是上面所返回的路径，另一个是‘\’，这个函数的作用是将当前可执行文件的文件名找出来。

运行到下一个call调用函数的地址。



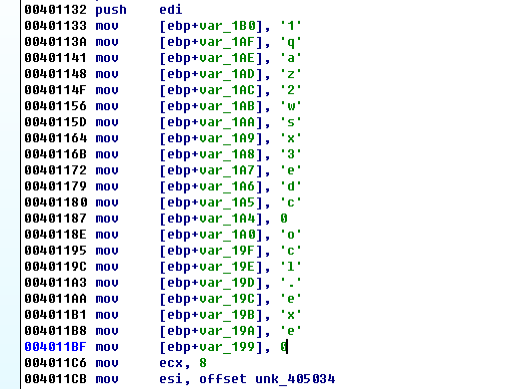
可以看到栈上有两个参数。

IMG_256

结合IDA我们可以得知这就是一个字符串比较函数。如果这个函数发现两个字符串不一样，那么就会退出程序。

将该可执行文件名称改为ocl.exe就可以运行该样本。

### 在地址0x00401133处发生了什么？



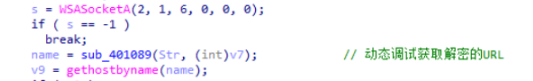
将两个字符串压入了栈中。

### 传递给子例程（函数）0x00401089的参数是什么？

参数是字符串和一个数据缓冲区，分别是“1qaz2wsx3edc”和解密密钥。

### 恶意代码使用的域名是什么？

通过此解密函数获取：



practticalmalwareanalysis.com

### 恶意代码使用什么编码函数来混域名？

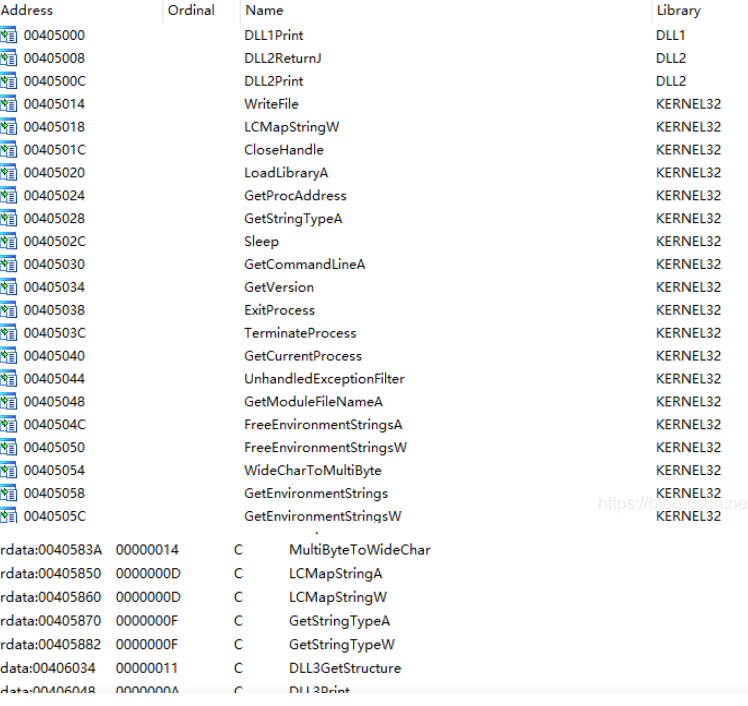
“1qaz2wsx3edc”异或加密混淆域名

### 恶意代码在0x0040106E处调用CreateProcessA函数的意义是什么？

创建和攻击者的反向连接。

### Lab 9-3

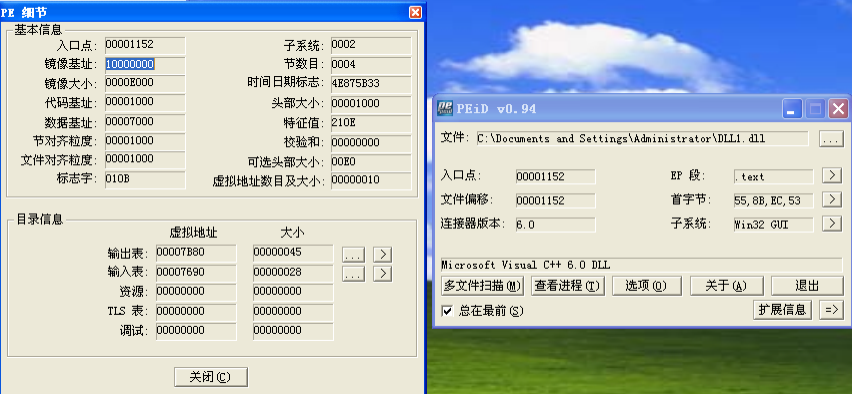
### Lab09-03.cxe导入了哪些DLL？



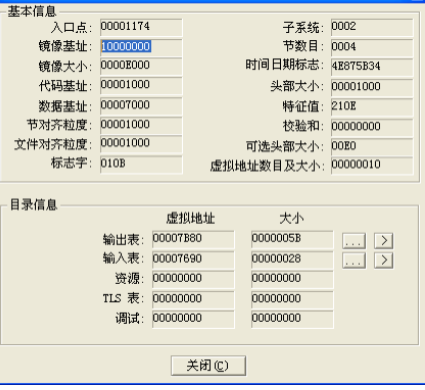
直接导入了DLL1、DLL2、Kernel32.dll。虽然没有直接导入DLL3但是字符串中有DLL3相关的字符串。

### DLL.dll、DLL2.dll、DLL3.dll 要求的基地址是多少

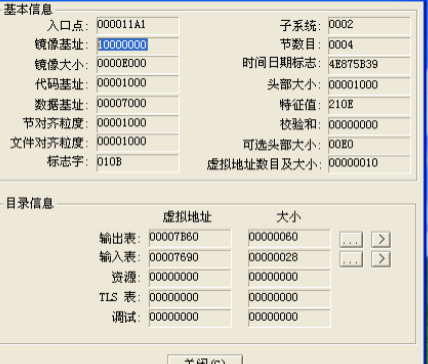
dll.all:



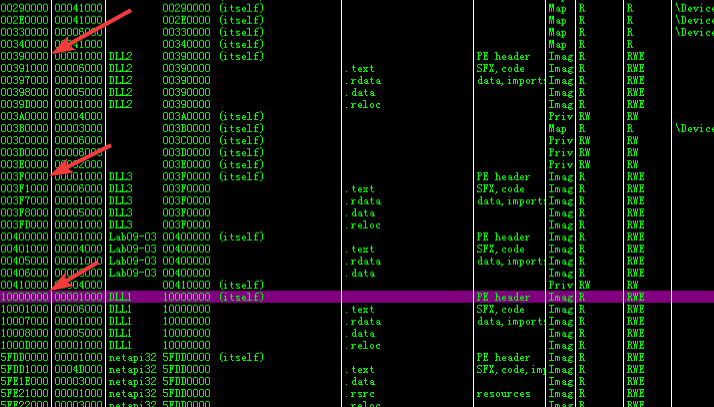
dll2.dll:



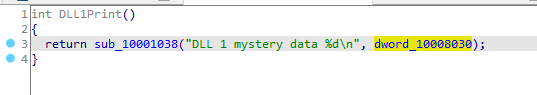
dll3.dll:

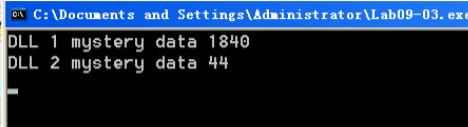


### 当使用OllyDbg调试Lab09-03.exe时，为DLL.d、DLL2.d、DLL3.du分配的基地址是什么？



### 当Lab09-03.exe调用DLL1.dll中的一个导入函数时，这个导入函数都做了些什么





DLL1Print函数输出进程PID。

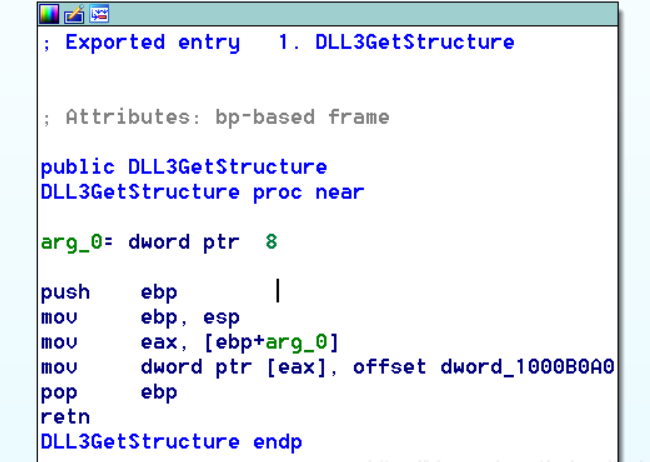
### 当Lab09-03.exe调用WriteFile函数时，它写入的文件名是什么？





文件名是“temp.txt”。

### 当Lab09-03.cxe使用NetScheduleobAdd创建一个job时，从哪里获取第二个参数的数据？



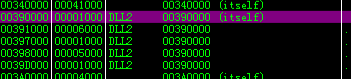


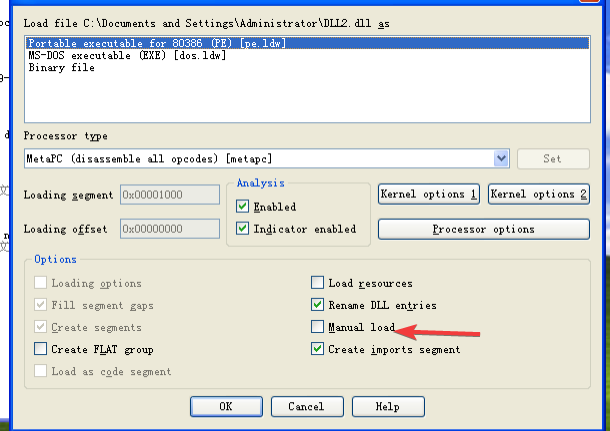
Lab09-03.exe 从DLL3GetStructure中获取NetScheduleJobAdd调用的缓冲区，它动态地解析获得第二个参数的数据。

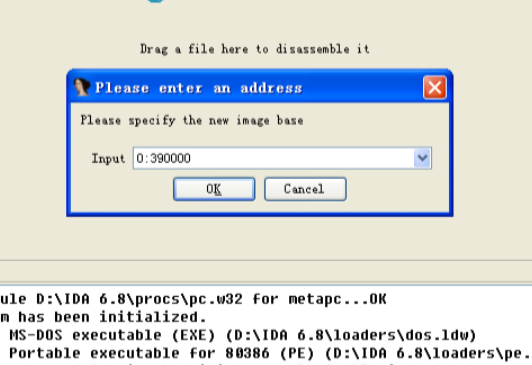
### 在运行或调试Lab09-03.exe时，你会看到Lab09-03.exe打印出三块神秘数据。DLL1的神秘数据DLL2的神秘数据，DLL3的神秘数据分别是什么？

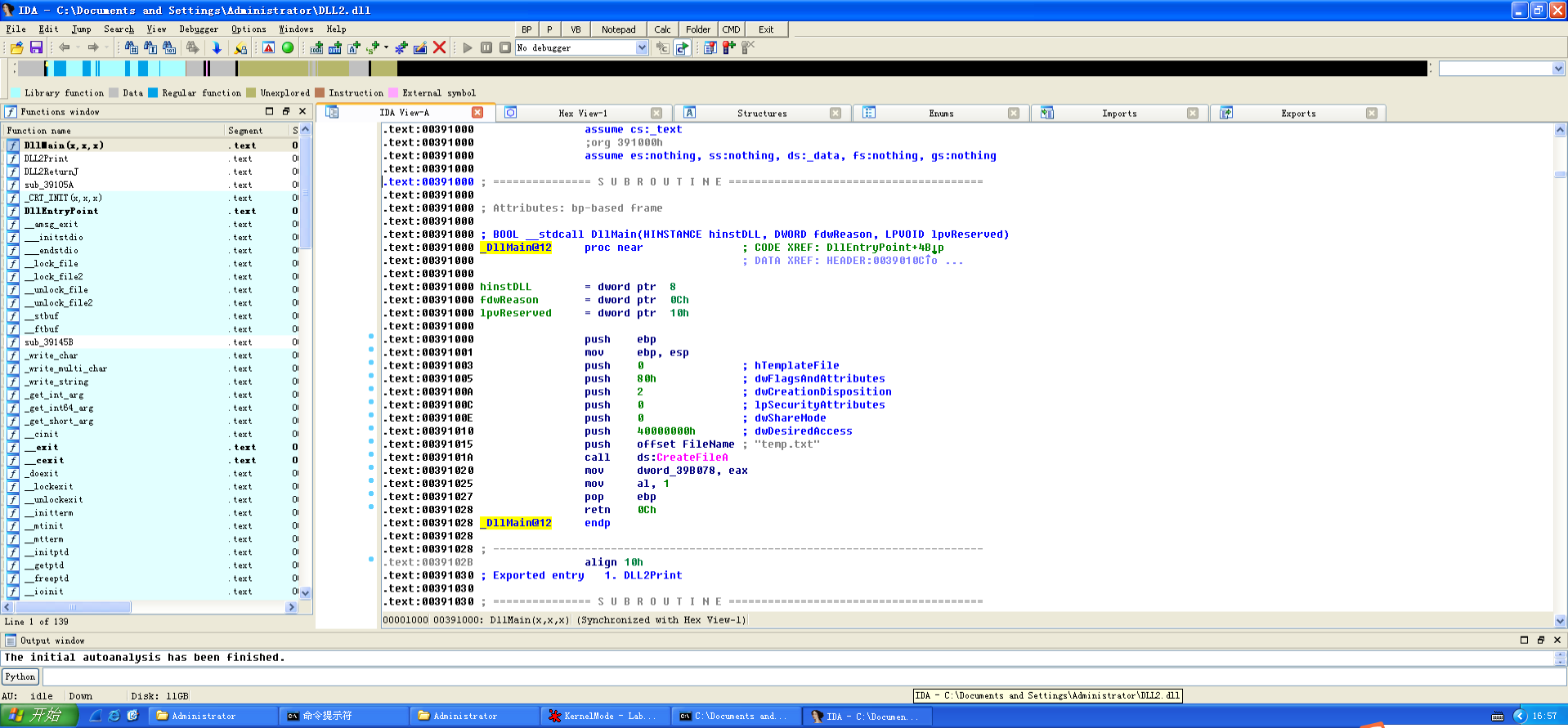
DLL1的神秘数据是进程ID，DLL2的神秘数据是打开temp.txt文件的句柄，DLL3的神秘数据是字符串“ping www.malwareanalysisbook.com”。

### 如何将DLL2.d加载到IDAPro中，使得它与OlyDbg使用的加地址匹配









## yara

import "pe"

rule UrlRequest {

strings:

$http = "http"

$GET = "GET" nocase

$com = /[a-zA-Z0-9\_]\*.com/

condition:

$http or $GET or $com

}

rule cmd {

strings:

$name = "cmd" nocase

condition:

$name

}

rule EXE {

strings:

$exe = /[a-zA-Z0-9\_]\*.exe/

condition:

$exe

}

rule Regedit{

strings:

$system = "system32"

$software = "SOFTWARE"

condition:

$system or $software

}

rule DLL {

strings:

$dll = "DLL"

condition:

$dll

}

rule SOCKET {

strings:

$name = "Socket"

condition:

$name

}

1. **实验结论及心得体会**

**对分析windows恶意程序有了更多的经验。**

**更熟悉恶意分析工具的使用。**