

**恶意代码分析与防治技术课程实验报告**

**实验三：基本动态分析**

****

学 院 计网

专 业 信息安全

学 号 2111033

姓 名 艾明旭

班 级 信息安全一班

1. **实验目的**

**学会windows操作系统下的动态分析软件的使用和应用场景。**

**对基本的程序动态分析有一定的了解，**

**综合运用各种分析工具，分析Lab03里所有程序的基本信息，并推测其功能。**

**学习基本的动态分析的流程，体会通过动态分析了解一个程序的思路和方法。**

1. **实验原理**
   1. **实验环境**

Windows xp，windows8.1，VMWARE，Windows11 ,kali linux

* 1. **实验工具**

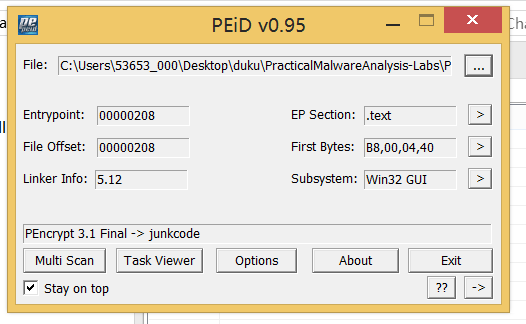
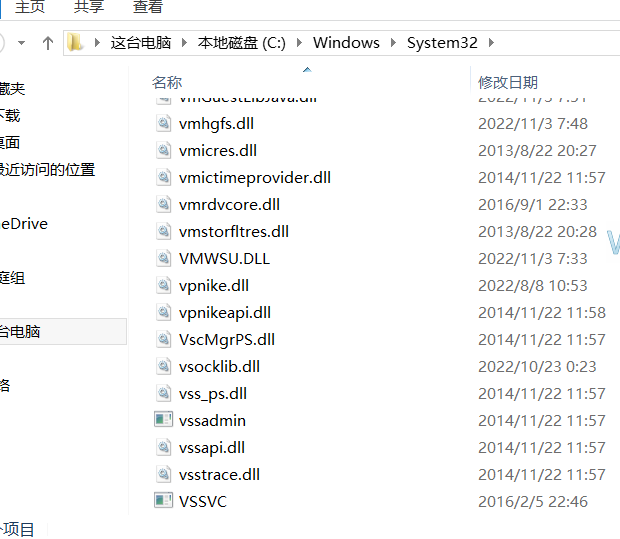
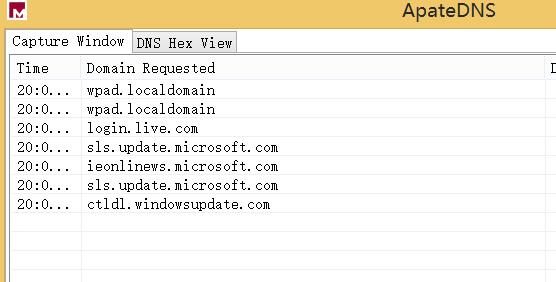
STRINGS, IDAPro,PEVIEW，YARA，upx,PEID,powershell，MD5deep,process Explorer,process monitor, ApateDNS,

* 1. **原理**

**动态分析可以针对一些需要打包或者在进行过模糊化处理之后可能在静态分析下没有办法分析查找其具有的问题，同时我们还不知道恶意代码能够产生的后果，可以对目标程序及逆行动态分析，了解恶意代码具有的问题和会造成的危害。**

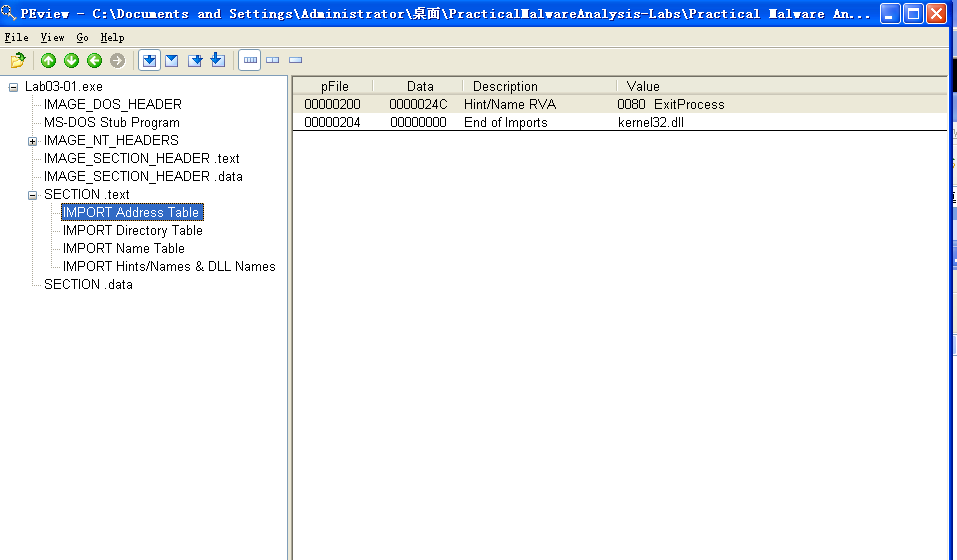
1. **实验过程**

**最开始我们选择了win8.1操作系统进行操作，首先是运行程序的时候，我们需要选择动态分析所有的问题，因此需要运行该程序，可是我们的Lab03-01.exe程序在运行的过程当中，首先会报错，之后我们再以管理员身份运行，会发现程序虽然可以运行，但是不仅在process explorer当中只会出现短暂的函数链接库的调用和跳动，之后我们再在process monitor当中尝试查找监视，发现在一定时间内并不能查找到完整程序的运行。在完成完整的程序分析过后，我尝试在ApateDNS和wireshark上查找该程序的痕迹，可以发现的是，该程序确实是在调用相关的动态链接库，尝试调用某些函数。但是我们可以发现其通过网络连接的网站并不是目标网站，同时函数执行的时间也非常短。最终的执行结果并不是预期的理想结果。从C:/windows/system32文件夹当中，我并不能查找到目标程序的复制程序，因此，我可以做出以下的判断，该程序在win8.1上的动态分析结果是失败的，我们并不能在较高版本的windows操作系统上成功运行分析改程序，需要我们到windows xp 系统中尝试进行分析。**

****

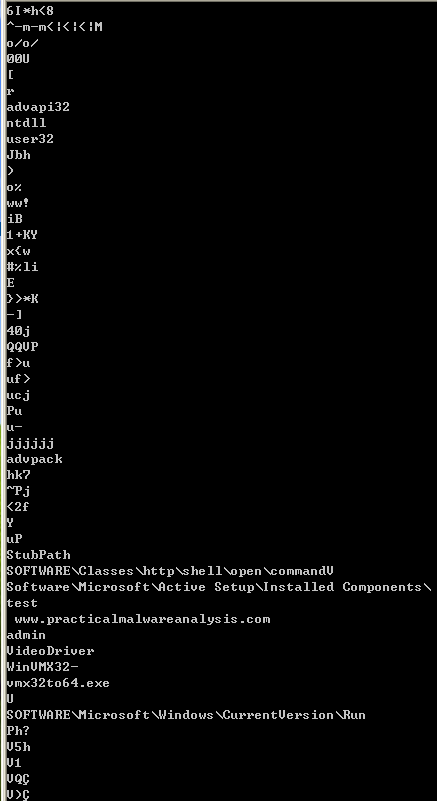
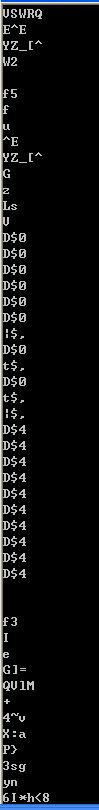
首先我们需要做的第一步工作就是利用windows xp进行静态分析（这一步在任何的windows操作系统当中都可以进行相关的尝试。我们首先是利用了MD5deep程序查看了相关程序的MD5码，MD5编码是密码学当中的一个重要的应用，不同应用程序，即使是一个微小的改动，MD5编码都会有很大的不同。然而对于相同的程序，MD5码却不可能不同，针对此功能，我们可以对相关应用程序的MD5编码进行检查，发现其有没有相同，相同则可以判断是同一个应用程序。

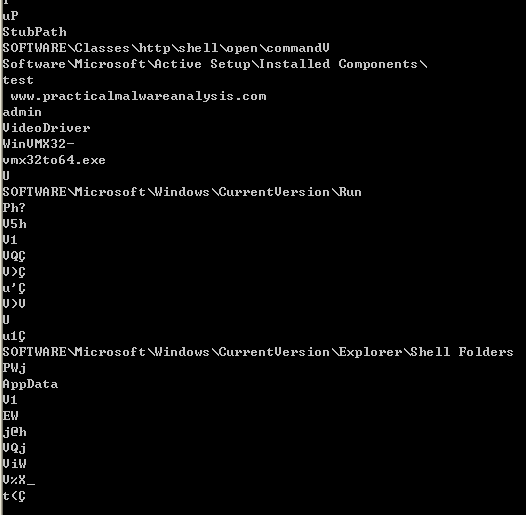
第一步我们首先针对文件的PEID进行检查，可以看到Lab03-01.exe已被加壳处理。



接下来我们分析文件的PE文件头。可以查看到相关的地址表和调用了哪些函数。使用PEview的分节IMAGE\_OPTIONAL\_HEADER.text/.data/.rsrc等查看

如果各节头表的虚拟内存大小＞原始内存大小，则该样本可能加了壳，但数据节头的虚拟内 存大小大于原始内存大小很正常

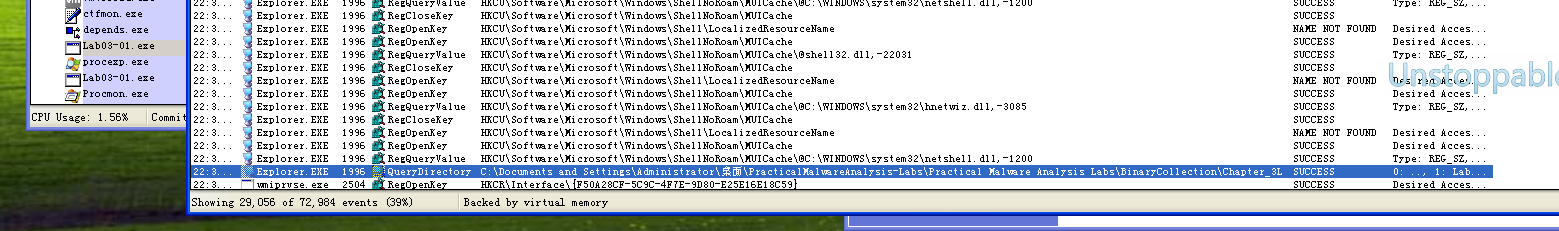


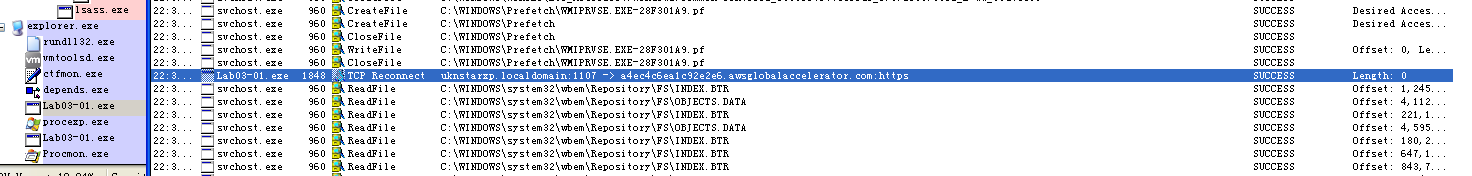
****

**之后我们对相关程序的字符串进行了分析，通过利用strings.exe应用程序，我们可以发现其中有很多类似打开文件或者网页的操作，需要我们着重注意。**

接下来我们打开process explorer ,process monitor，和regshot,ApateDNS，wireshark工具对相关的应用程序分析其是否具有变化。最基本的我们可以观察到的是出现了一个名为Lab03-01.exe的thread，并且其调用了很多的动态链接库。

其中的调用函数有许多我们认为有相当的危险性，需要我们去着重注意是否程序会因为其而固定对计算机系统造成危害。

****

****

****



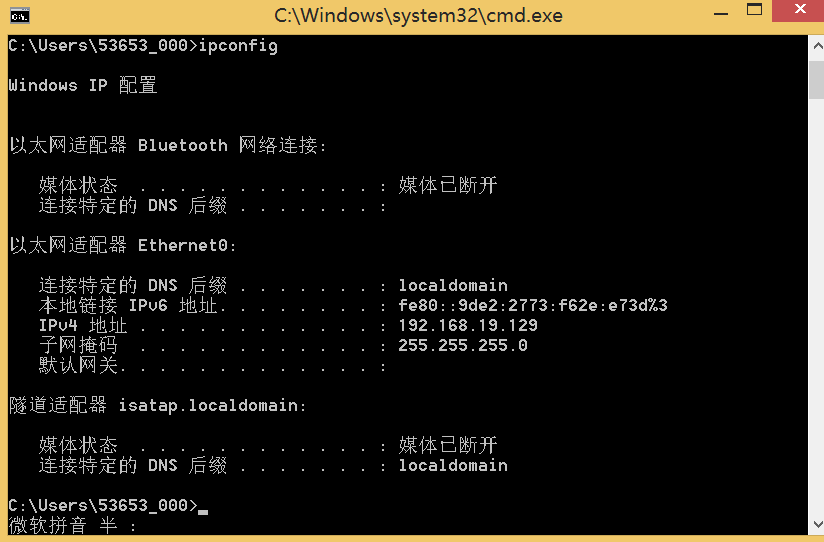
我们在最开始调用相关程序的时候发现并不能很好的运行，这里我们发现需要对其使用inetsim工具进行相关内部网络的配置。

****

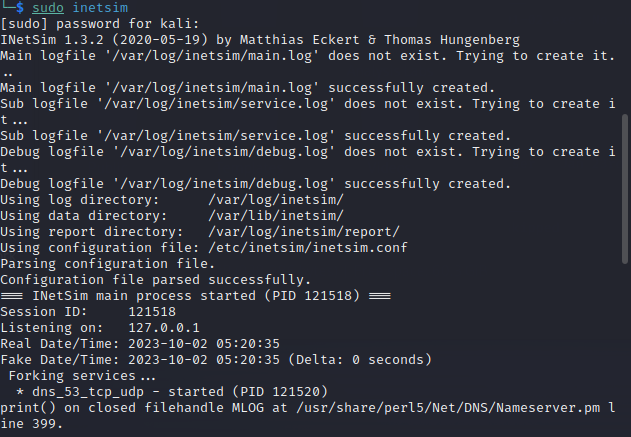
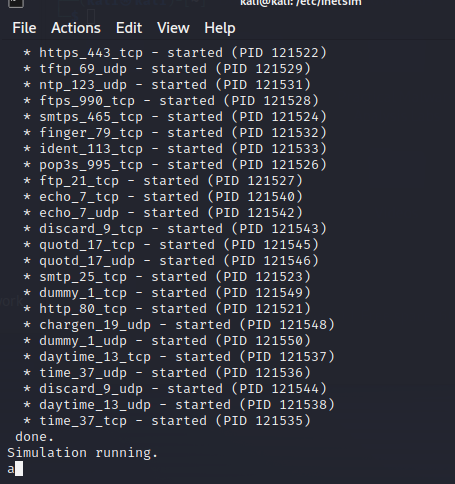
**我们需要在几台虚拟机当中配置一个内部网络，通过kali linux自带的inetsim功能模拟出相关的网络，从而保证网络不会被感染，也能使程序能够正常的运行。**

****

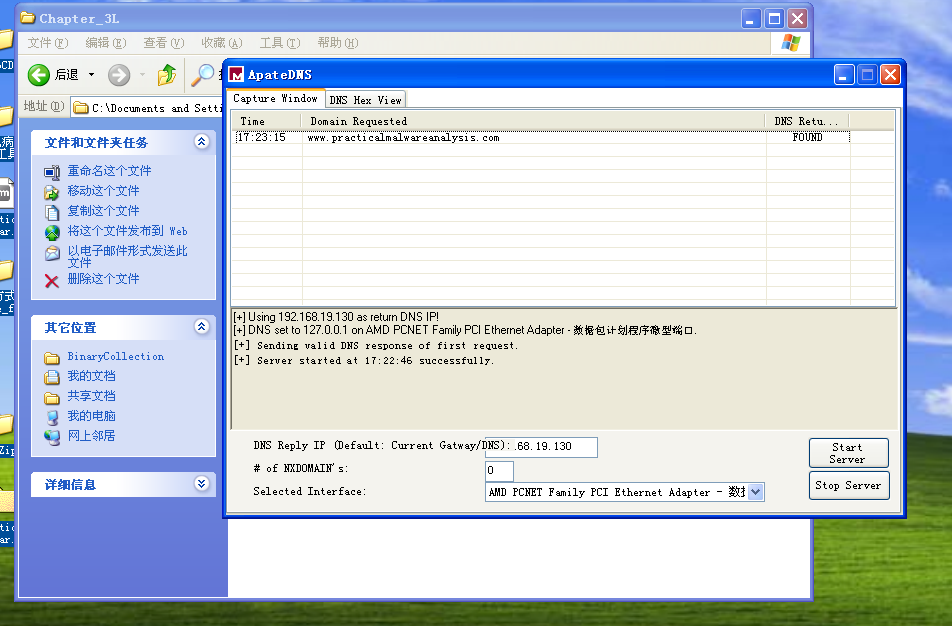
首先我们需要确定一个相关的内部仅主机模式连接的网络，配置其子网IP，并设置相关的DHCP，可以采用默认设置。

****

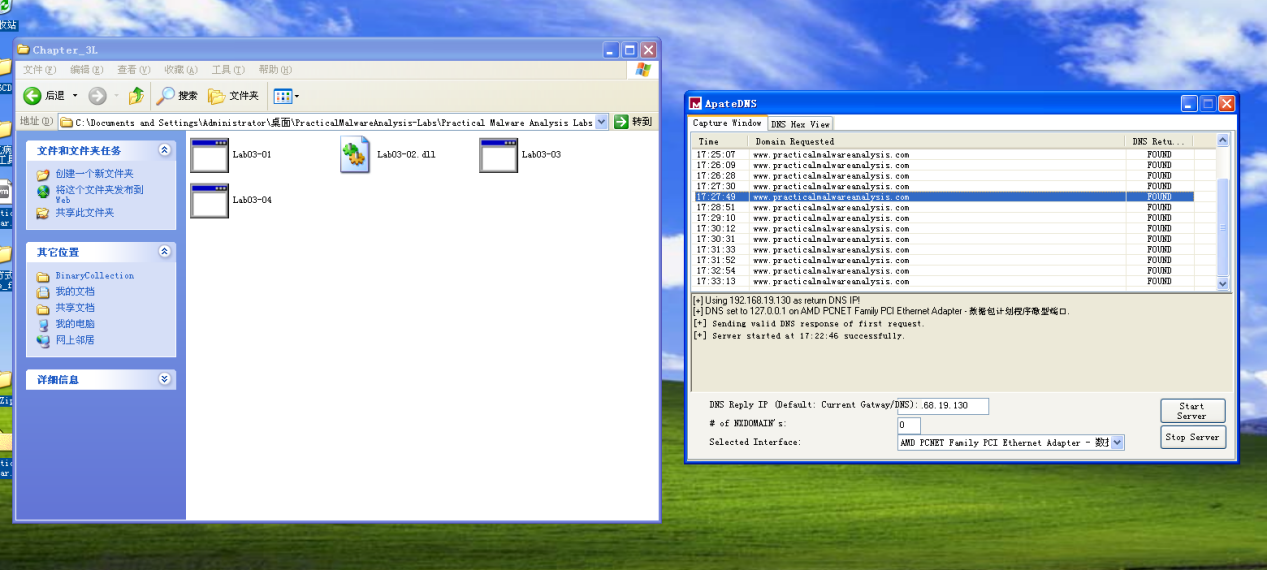
接下来我们就需要把所有的相关虚拟机接入我们的内部网络当中，如图所示，在相关的虚拟机当中输入IPconfig指令之后，可以发现他们的IP地址已经变成了我们设置的内部网络的地址，接下来我们就可以在内部网络当中运行该程序，同时wireshark和ApateDNS设置的监听接口也应该是我们设置的目的接口，对目的接口进行监听可以有效的监测出应用程序的网络请求。

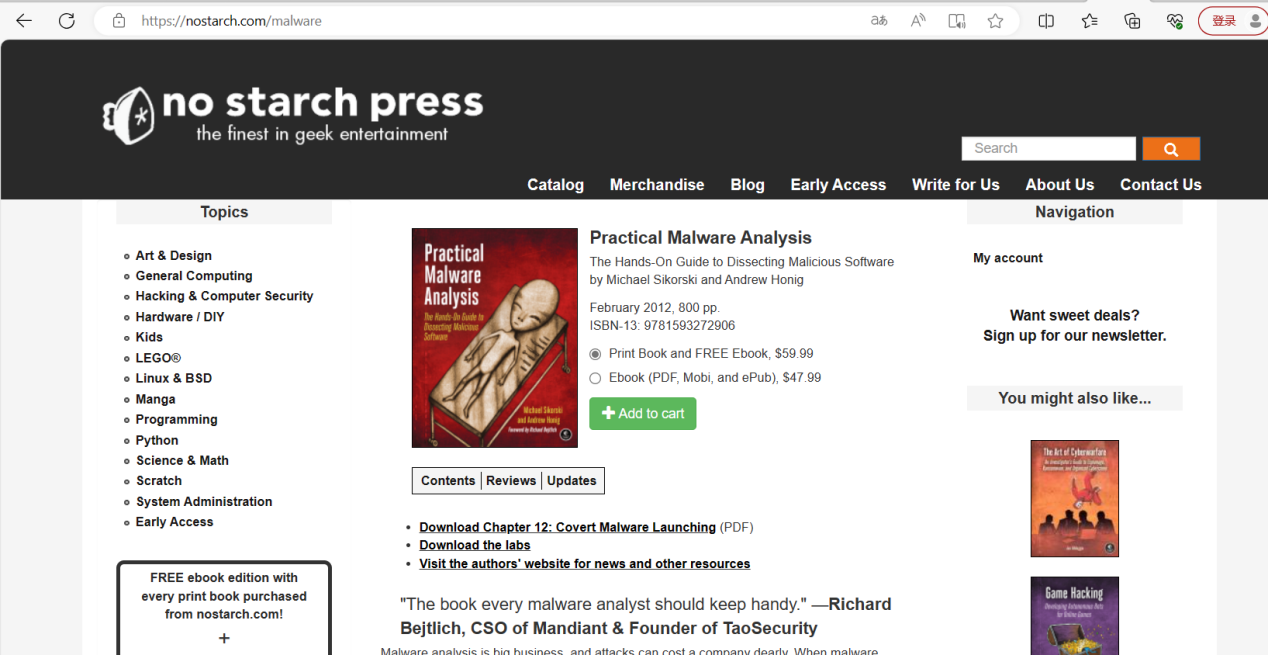
****

之后我们还需要将inetsim在kali linux 当中运行才算正式的成功运行，可以发现如图已经成功将inetsim运行起来。

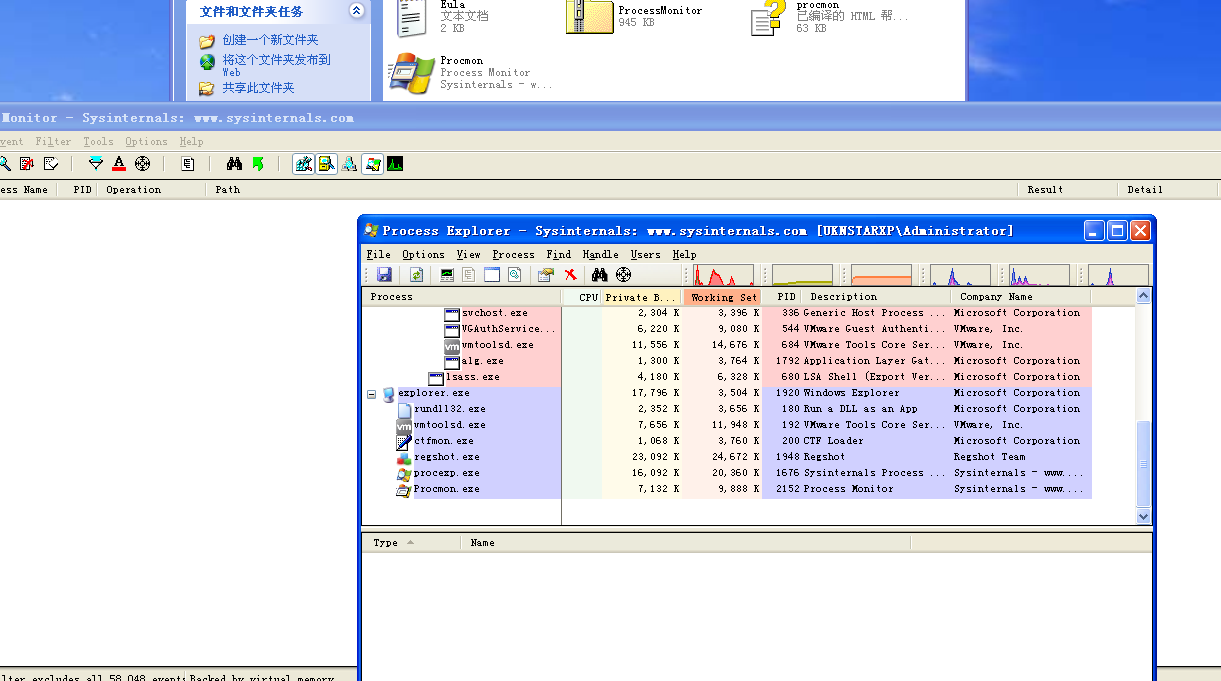
****

运行之后，我们可以在ApateDNS上发现目标代码在不断的访问一个网站。

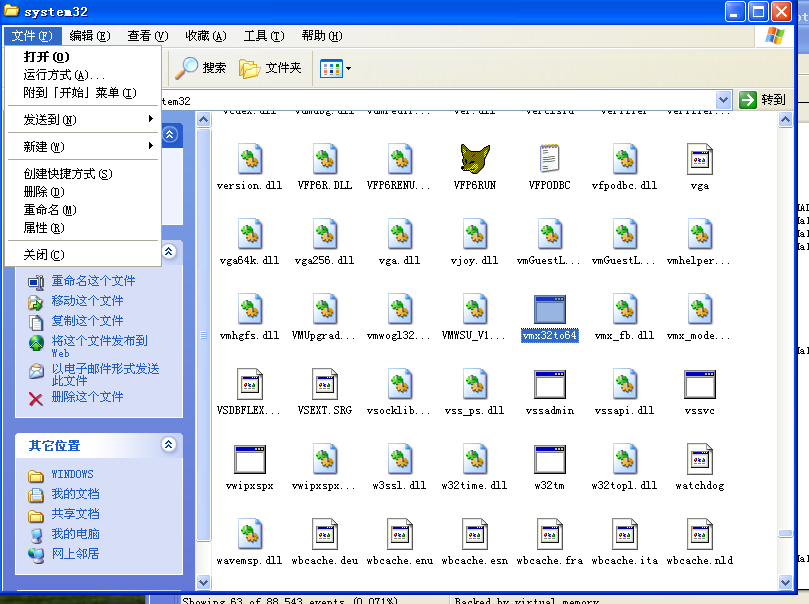
****

****

**我们可以在主机上查找运行该网站，（虚拟机由于在虚拟网络而非与主机互联的网络当中，因此没有连接互联网的功能）可以发现是本教材的官网。**

****

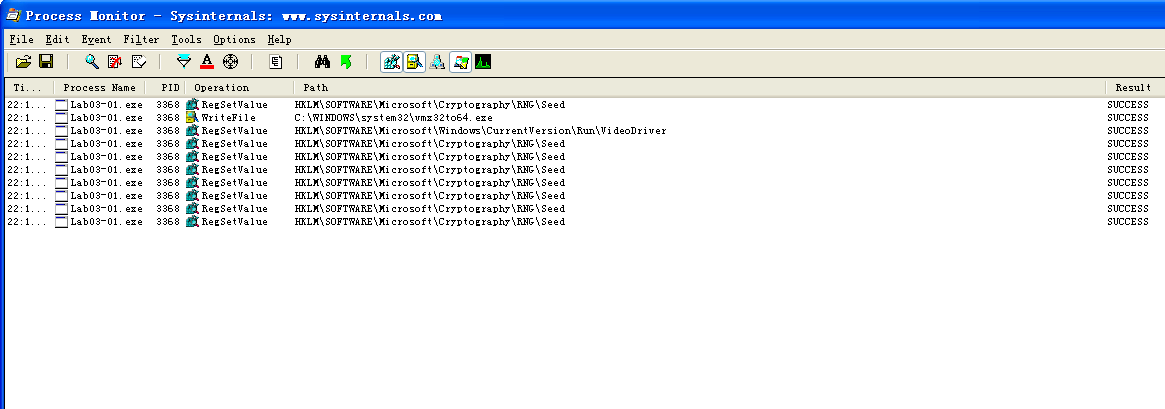
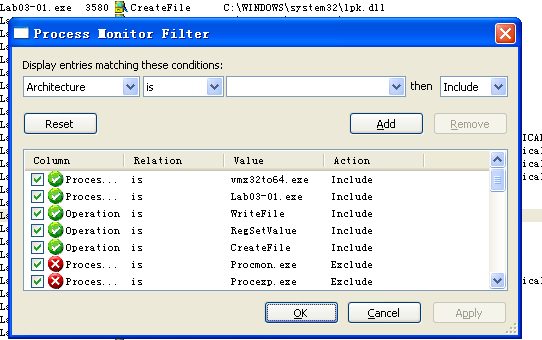
**我们继续在process explorer中寻找问题，在检查进程的时候，选择查找handles，可以发现恶意代码已经创建了一个名为winvmx32的互斥量，接下来我们又检查了它的DLL可以发现了一些具有联网功能的链接插件，可以验证我们上述猜想。**

****

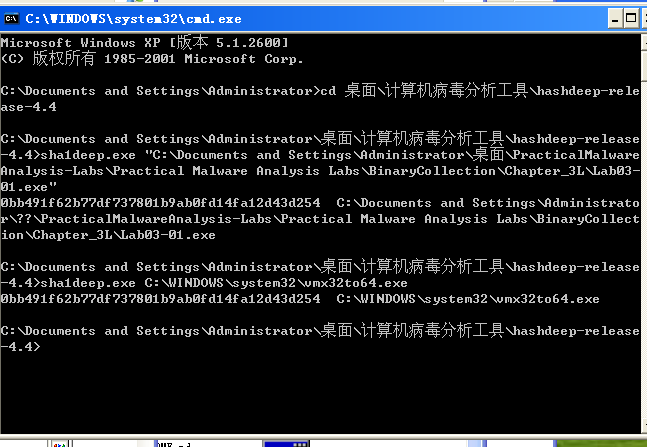
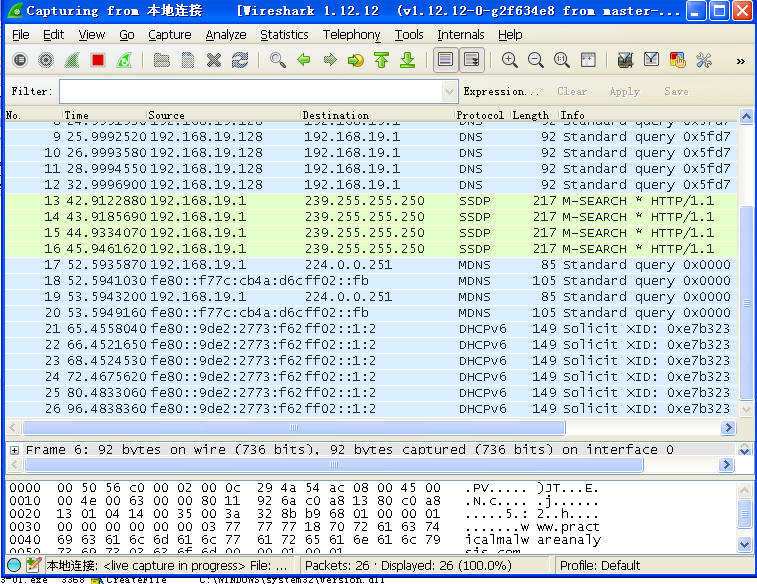
**之后我们打开regshot对运行应用程序前后的状态变化，可以发现应用程序进行了大量的修改，其中发现在自启动项VideoDriver中增加了键值43 3A 5C 57 49 4E 44 4F 57 53 5C 73 79 73 74 65 6D 33 32 5C 76 6D 78 33 32 74 6F 36 34 2E 65 78 65，将它换成字符为C:\WINDOWS\system32\vmx32to64.exe，说明VideoDriver自启动项就是指向system32目录下的vmx32to64.exe**

****

接下来我们分析process monitor当中可能存在的问题，我们把查找的文件设定为Lab03-01.exe把操作设定为writefile和regsetvalue，这样我们可以省略掉大量的无关操作，直接的寻找得到相关程序的关键操作，可以看到其候选进行了很多seed的更新，属于是无关操作，重点在于创建了seed和进行了vmx32to64这个文件的写文件操作，而这个写文件的操作使我们发现应用程序可能在系统当中创建了一个名为vmx32to64.exe的文件。

****

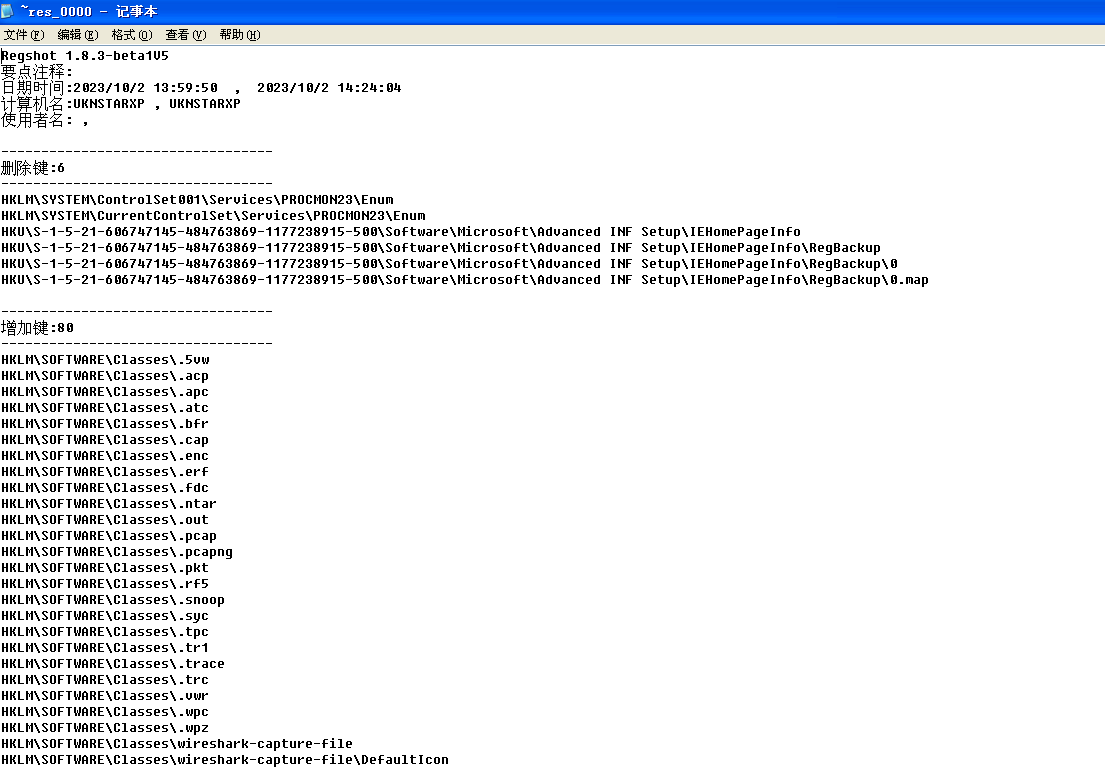
**之后我们在wireshark进行抓包，可以发现其在网络当中进行了大量的请求，确实是在链接网络。**

****

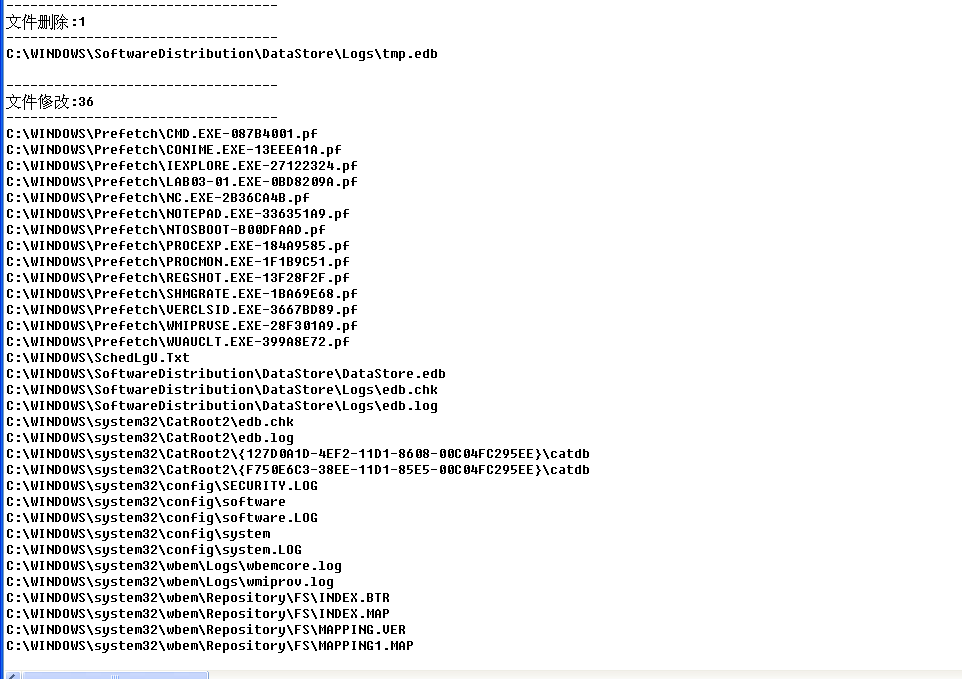
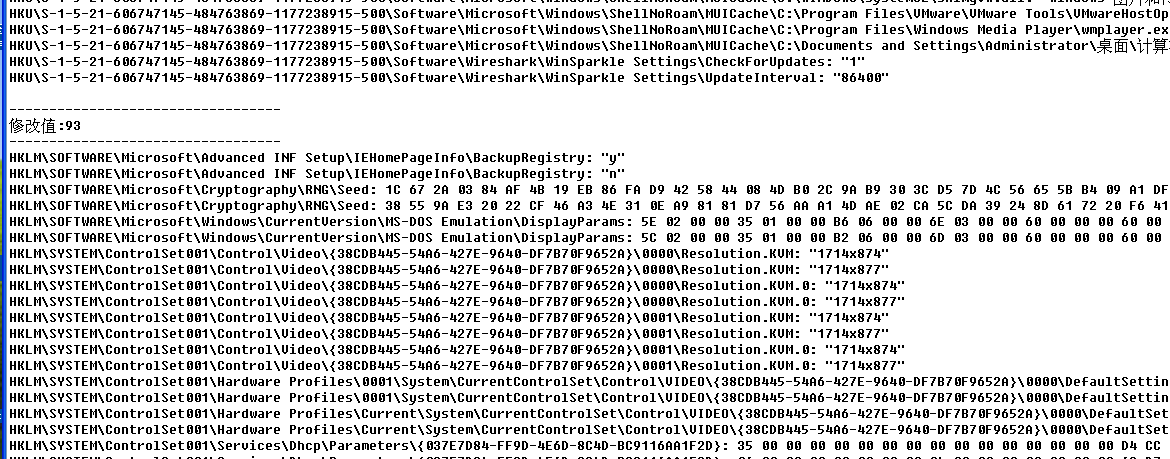
**之后我们对两个文件的MD5码进行了分析，可以发现两个文件的MD5码是相同的，可以说明我们的结论：Lab03-01.exe的功能将自己复制到system32目录下，并开机自启动**

**且Lab03-01.exe运行使访问www.practicalmalwareanalysis.com网址。**

**接下来是完整的regshot分析文件，其中有些干扰项，但是我们可以找到相关文件的增加和删除操作。**

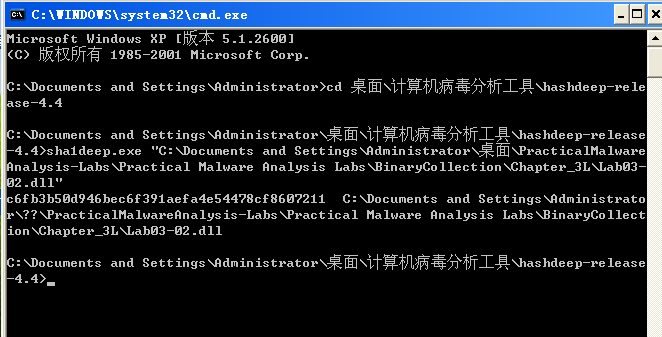
****

至于相关文件的删除，我们发现直接对其删除是不可能的，因为它一直处于开机启动且自动运行的状态，所以我们需要将网络链接调整回原来的状态，vmx32to64.exe就不能够成功的运行，我们就会发现它可以被删除掉，接下来就不会对我们后续的实验分析造成干扰。



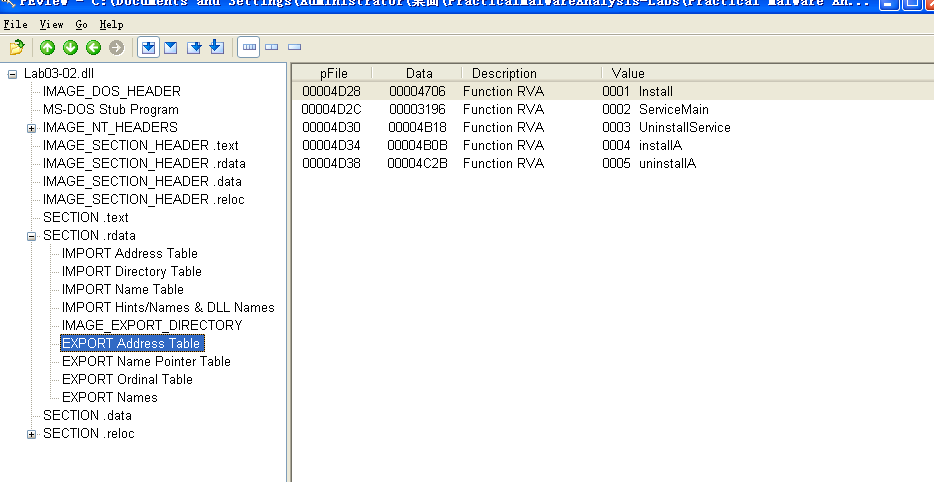
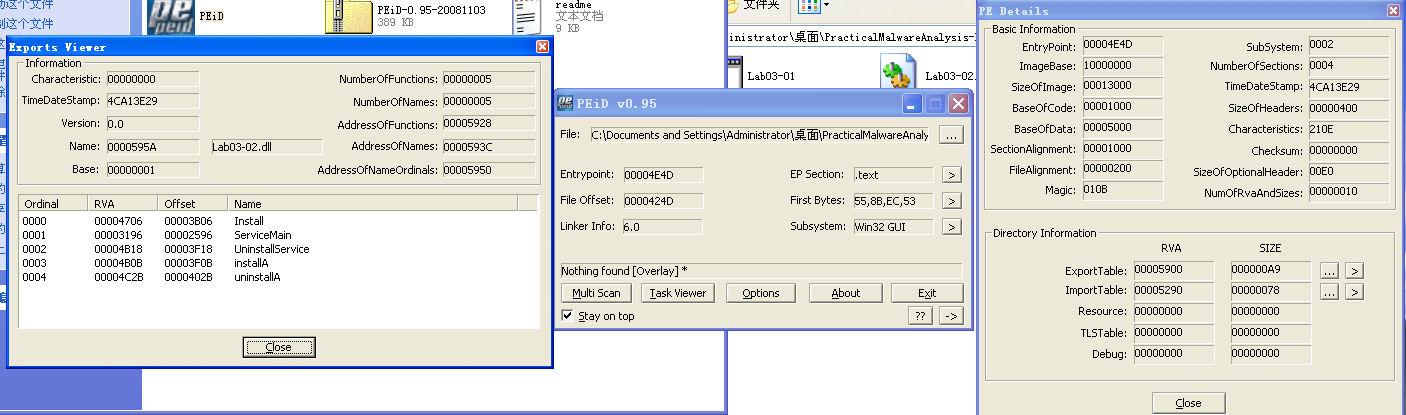
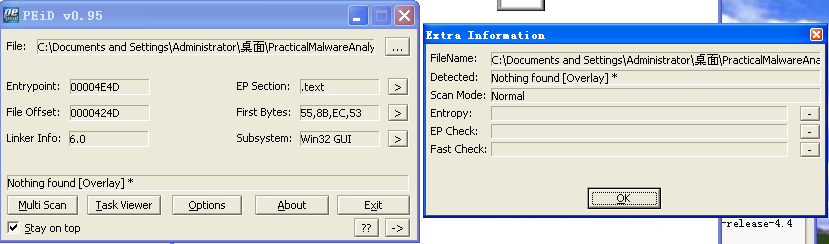
**Lab03-02.dll**

首先还是检查相关的MD5码，并记录相关的结果。



之后我们还是用PEID和PEview检查加壳

从中我们看到并未加壳处理，再观看它的导从中我们看到共5个导出函数，其中ServiceMain是和服务相关的函数，所以猜测Lab03-02.dll目的是下载某个服务。



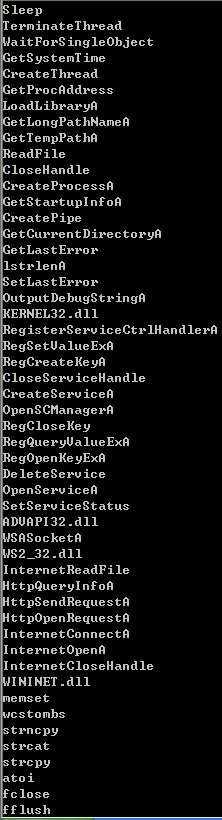
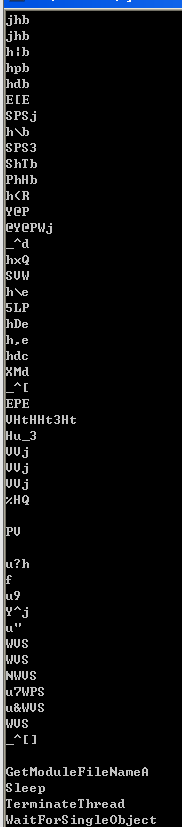
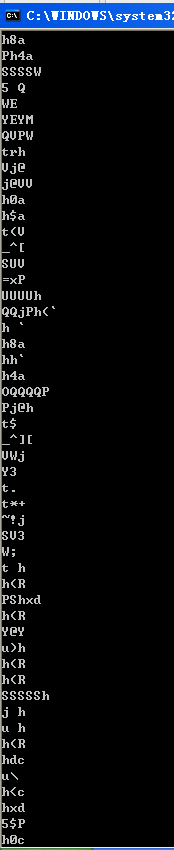
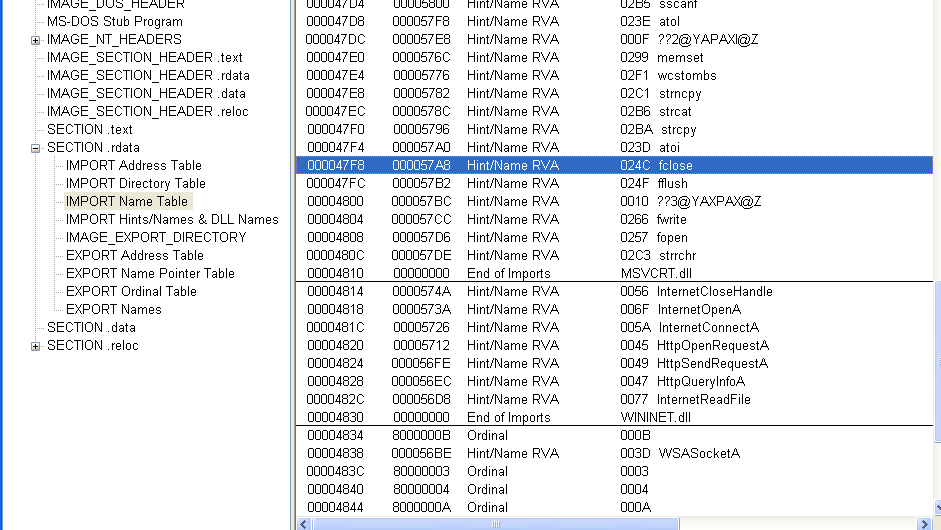
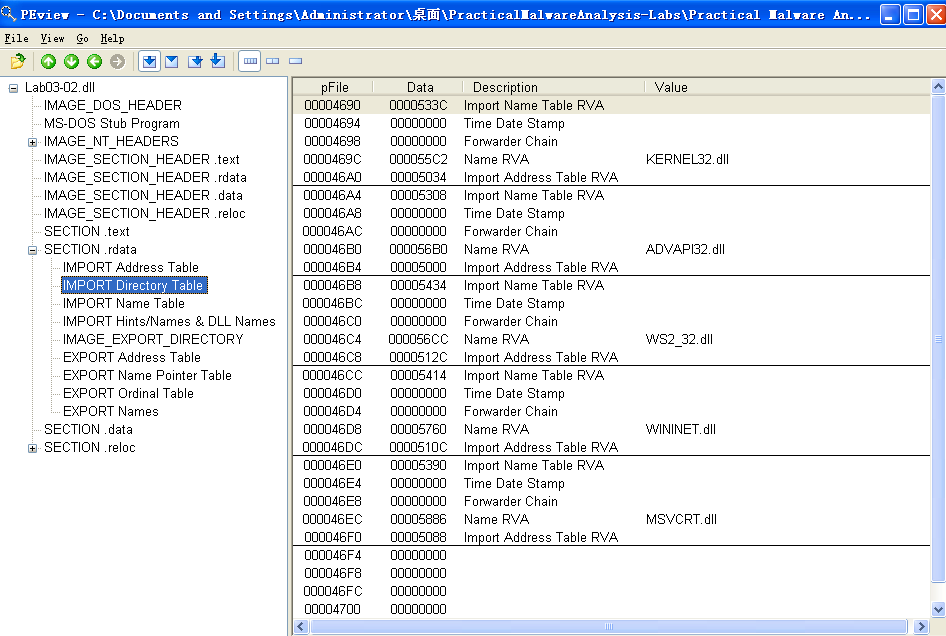
peview分析

首先它的导出函数和在peid中查看的相同

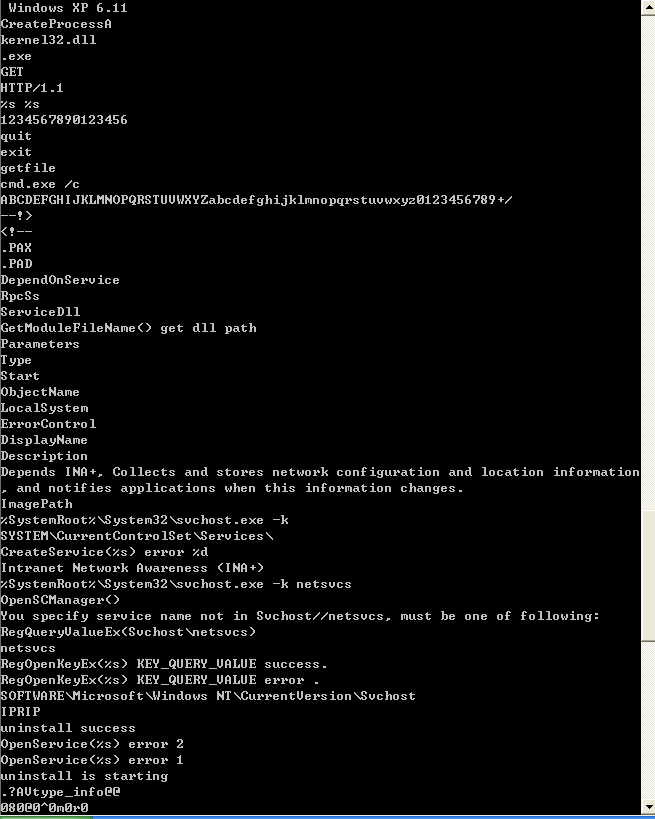
然后我们再来看一下它的导出表，发现了一些依赖导入函数动态链接库

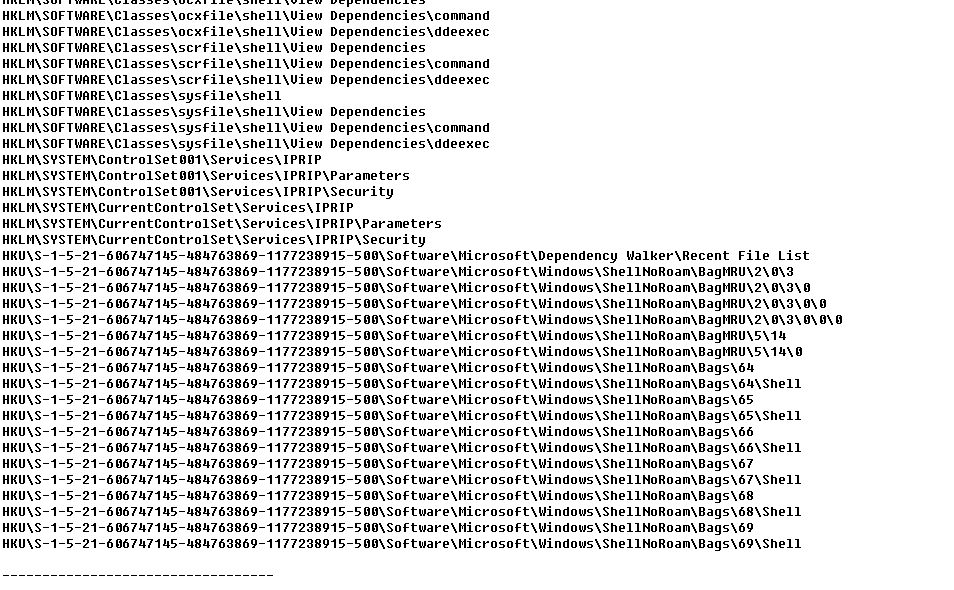
其中，通过观察ADVAPI32.dll依赖的导入函数，发现该恶意代码的功能实现中涉及了开启服务；开启注册表、注册表查询、创建注册表键、设置注册表键值、注册服务控制处理、设置服务状态等。这显然是为了实现ServiceMain的正常运行。

并且在它的导入表中发现了一些涉及网络访问功能的函数。



接下来我们在strings.exe工具当中尝试寻找相关的应用程序和字符串特殊表示，可以发现一些特殊的字符串试图引导程序调用一些函数进行特殊的操作。





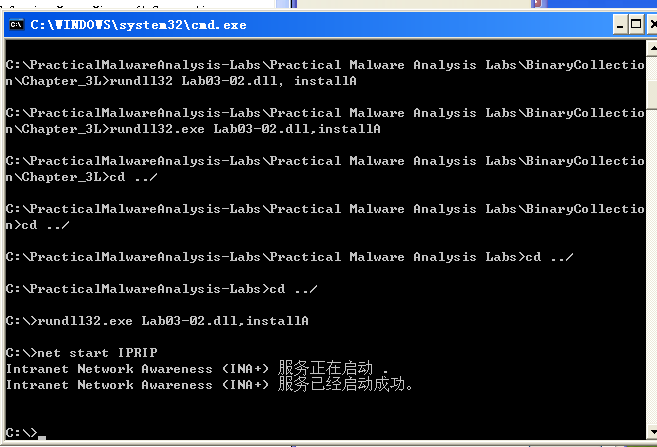
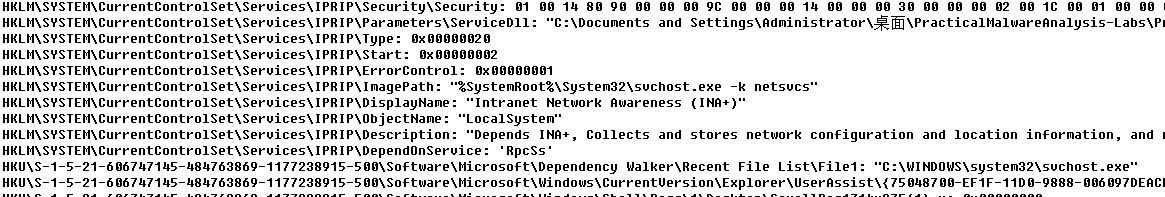
通过观察可打印字符串中发现，除了同上发现的一下导入导出函数外，还发现了practicalmalwareanalysis.com等一些域名信息以及一些注册表信息，此外还发现了svchost.exe的可执行程序。

通过以上信息可以得出总结，该程序可能通过访问 practicalmalwareanalysis.com网址，来下载某些程序，将自己下载为某个服务，而dll文件在Windows系统中一般不可以直接执行，所以必然存在某个可执行程序来运行Lab03-02.dll文件，而在可打印字符串中发现的svchost很有可能便是运行dll文件的exe文件。

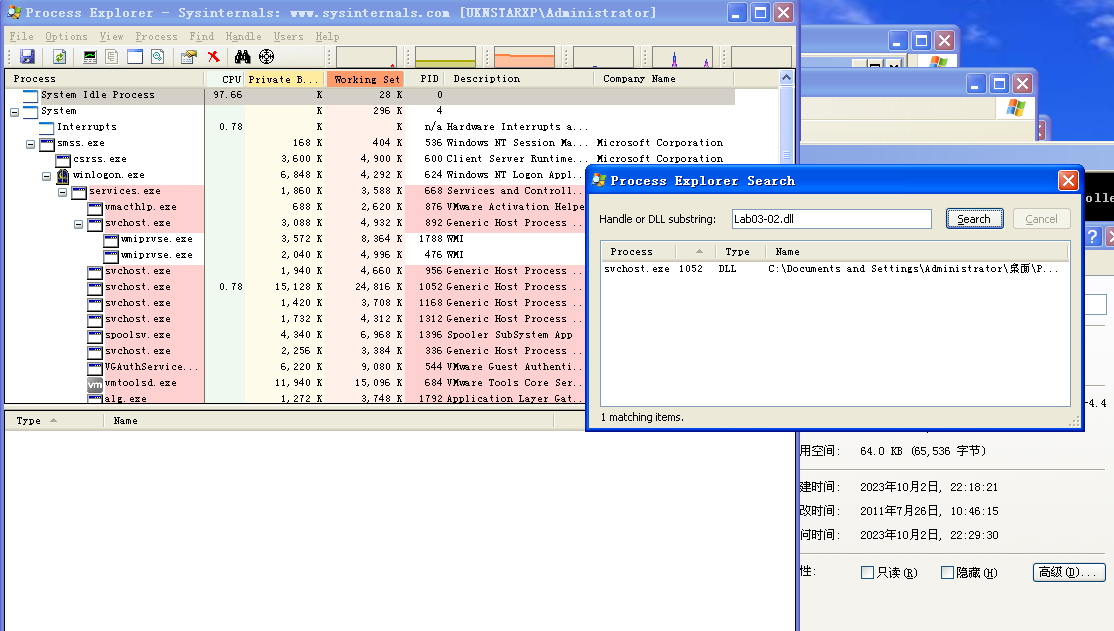
之后我们运行regshot，并且对相关的应用程序进行了比对，可以发现其中存在了一些问题。

在key adds中发现增加了一个IPRIP的服务，说明Lab03-02.dll将自己安装成一个IPRIP服务，印证了上述的猜测，并且dll需要一个可执行程序来执行它，在注册表中发现了svchost.exe执行程序，加重了是它运行的嫌疑。该恶意代码执行时显示的名字为“Intranet Network Awareness（NIA+）

而这个目的地址我们在字符串分析的时候也发现了存在相关的问题。



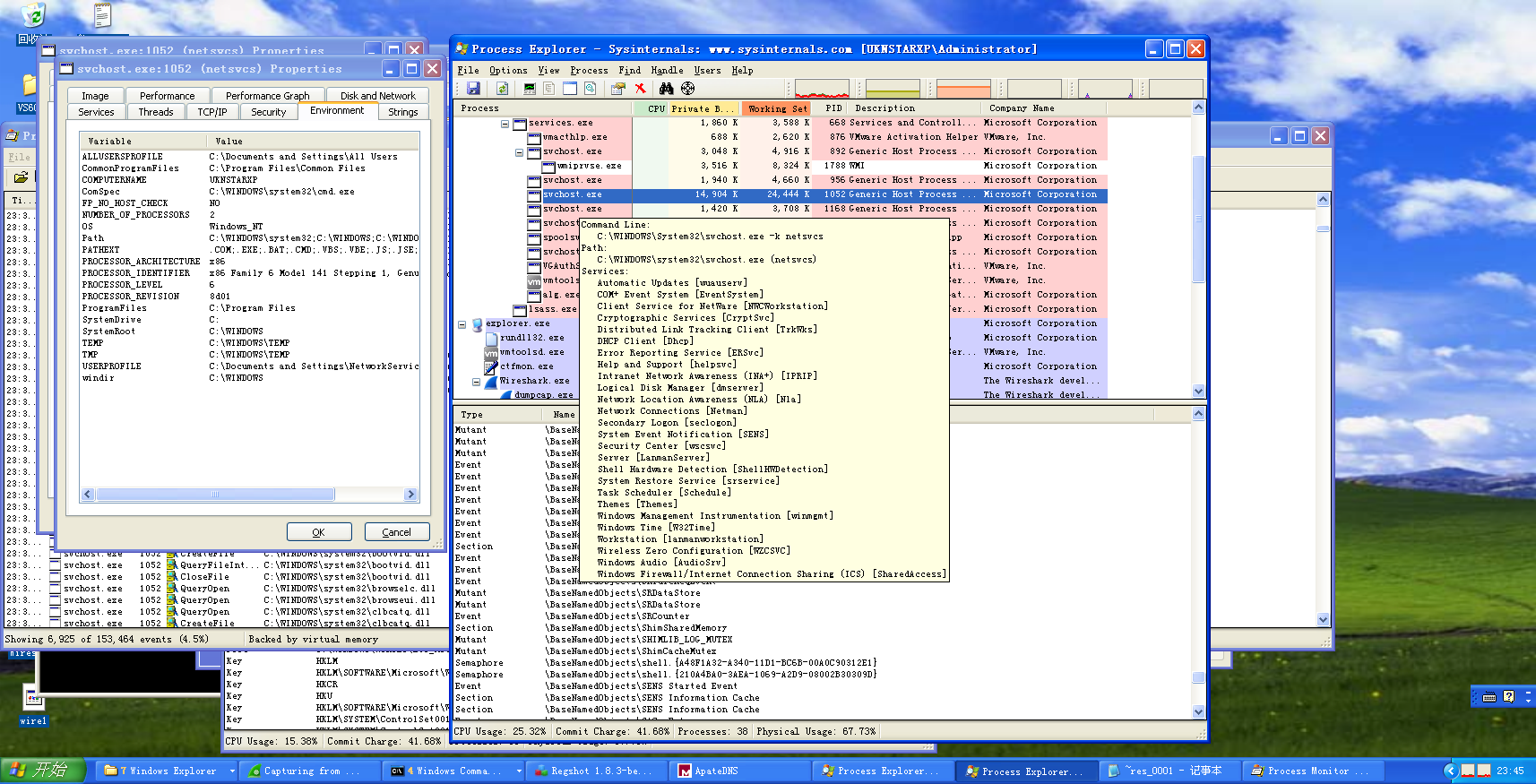
运行该程序需要我们利用rundll32.exe，我们把相关的应用程序启动之后，可以发现有一个INA+服务已经启动的标识，证明我们之前的猜想是正确的。



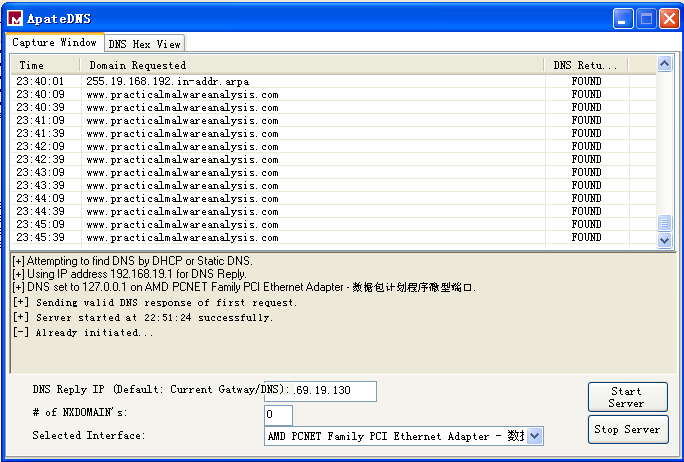
process explorer分析

开启IPRIP服务后ctrl+F搜索Lab03-02.dll后，发现Lab03-02.dll确实附着在svchost.exe上运行，其pid为1052（不同电脑可能会不同），因此在这里就可以确定Lab03-02.dll就是通过svchost.exe执行的.

之后找到1052程序，确实在它的里面找到了Lab03-02.dll



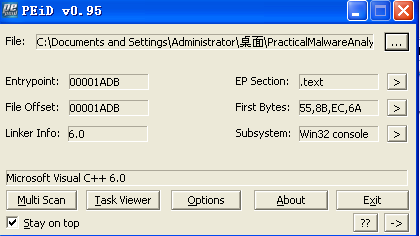
在相关的文件当中，我们可以发现其具有特殊的动态链接库，并且动态链接库有一定的问题，它在尝试性的调用Lab03-02.dll。



接下来我们继续分析ApateDNS，启动IPRIP服务后发现Lab03-02.dll访问了practicalmalwareanalysis.com网址。

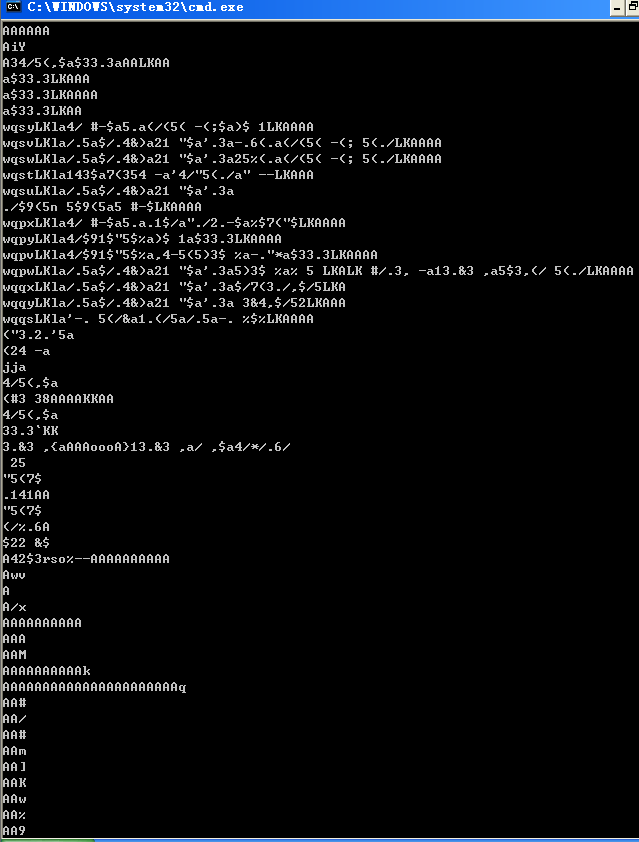
之后的操作过程当中我还在kali linux当中尝试对netcat进行监听，但是我并没有发现有用的结果，可能是监听的端口并不正确，与书上示例采用的端口并不相同。

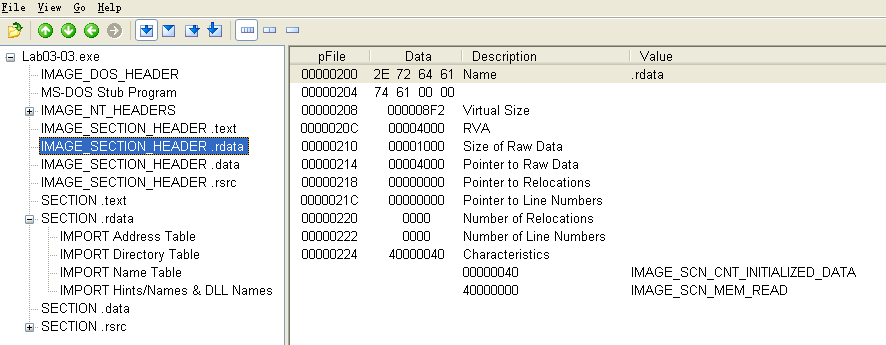
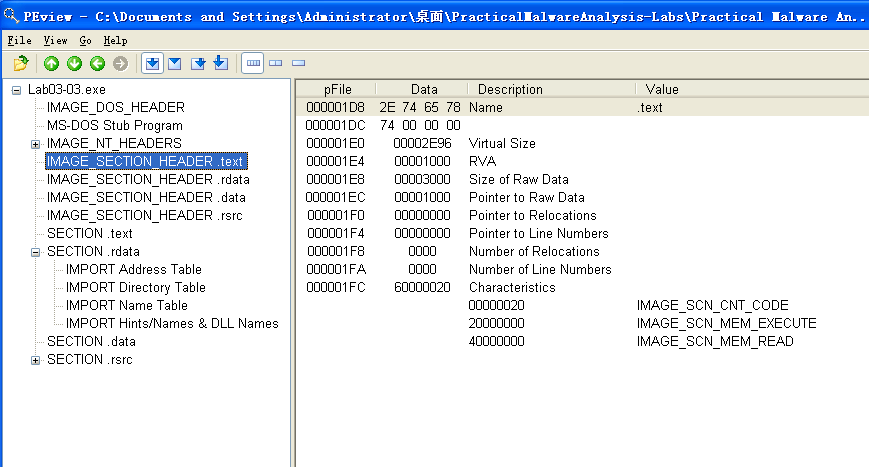
Lab03-03.exe

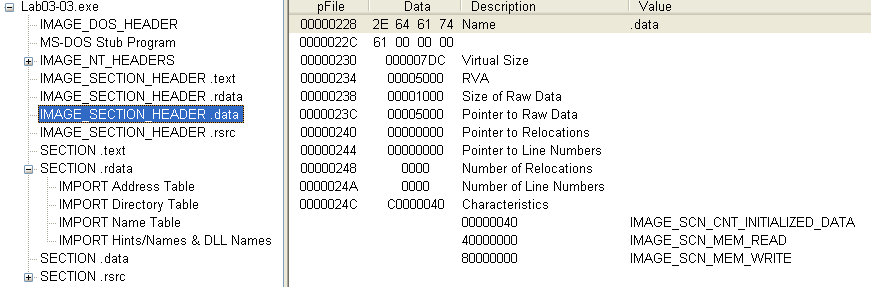


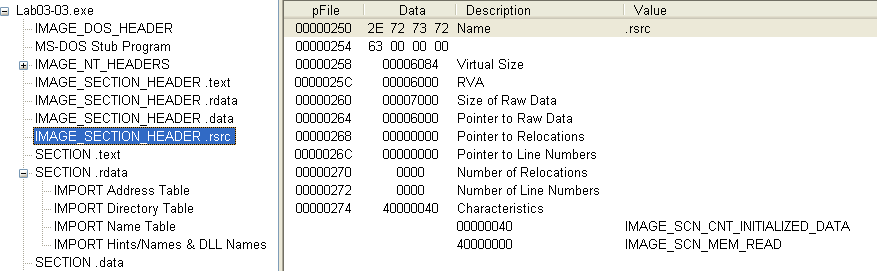
PEID的显示当中，我们发现的是该程序并没有出现加壳，用strings查看可打印字符串时，没有发现任何有用的信息，只是一些乱码。说明我们的程序可能采用了一些特殊的功能，隐藏了自己的许多恶意功能不被发现，需要我们去做进一步分析才能得出一个比较好的结果。

因此我们接下来选择查看PEview当中是否会存在该程序有怎样的隐藏功能需要我们通过静态分析就能够挖掘发现。将它拖到peview中查看，发现各节头表的虚拟内存大小均小于原始内存大小，因此也没有发现，使用了非常规的混淆技术。



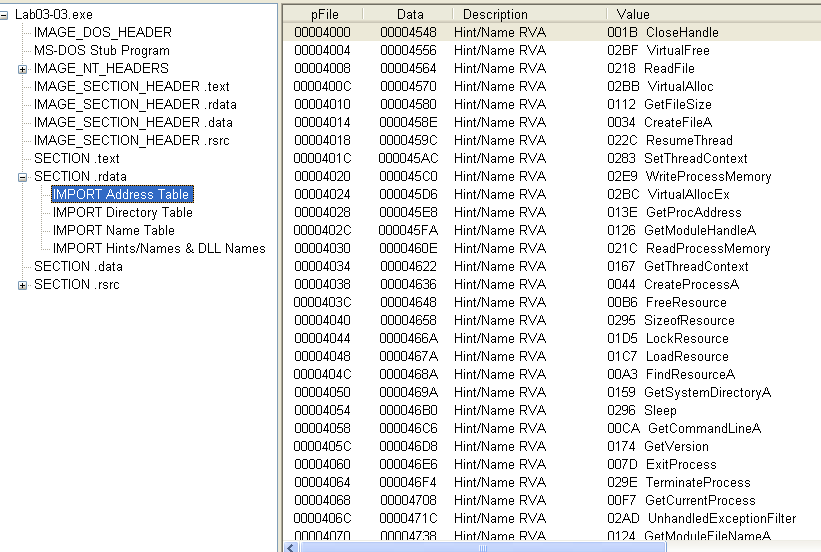


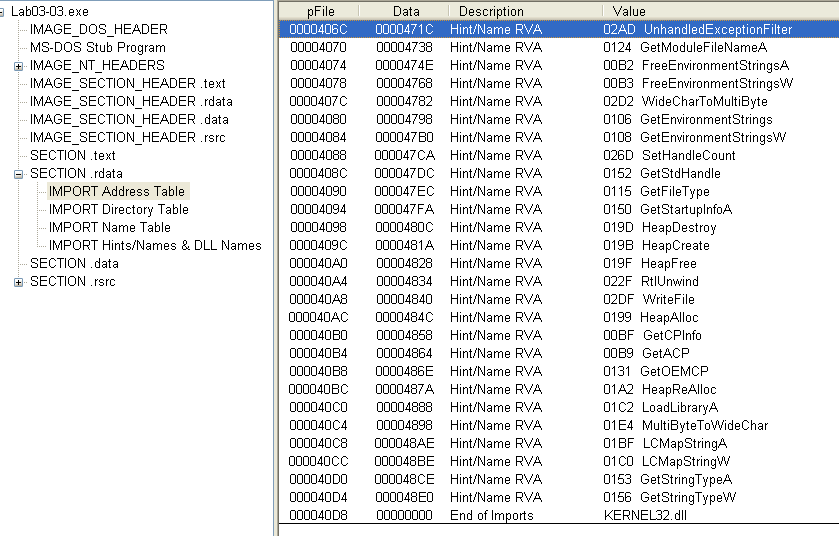




然后我们再次查看它的导入函数列表，发现了许多与文件和进程相关操作的函数。例如：get procaddress,Load libraryA，close handle, read file, get filesize, resume thread, write process memory, getthreadcontext,createprocessA，getFileType,WriteFile等函数

所以猜测Lab03-03.exe是对内存中的某些文件进行创建修改删除等恶意操作。

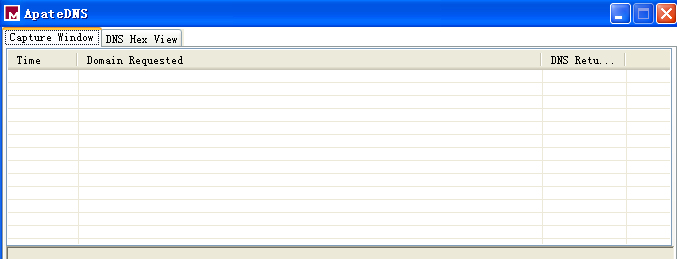


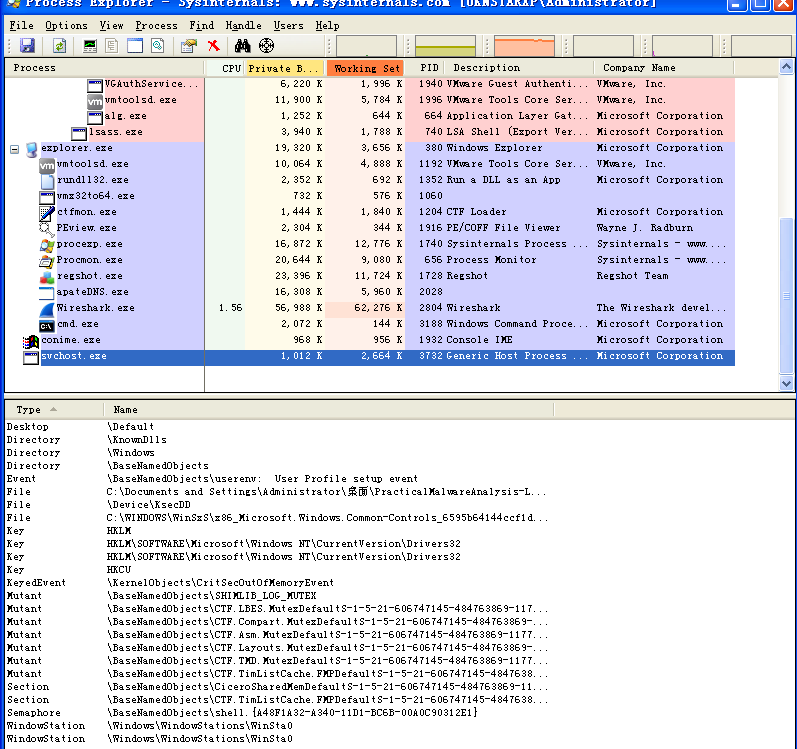
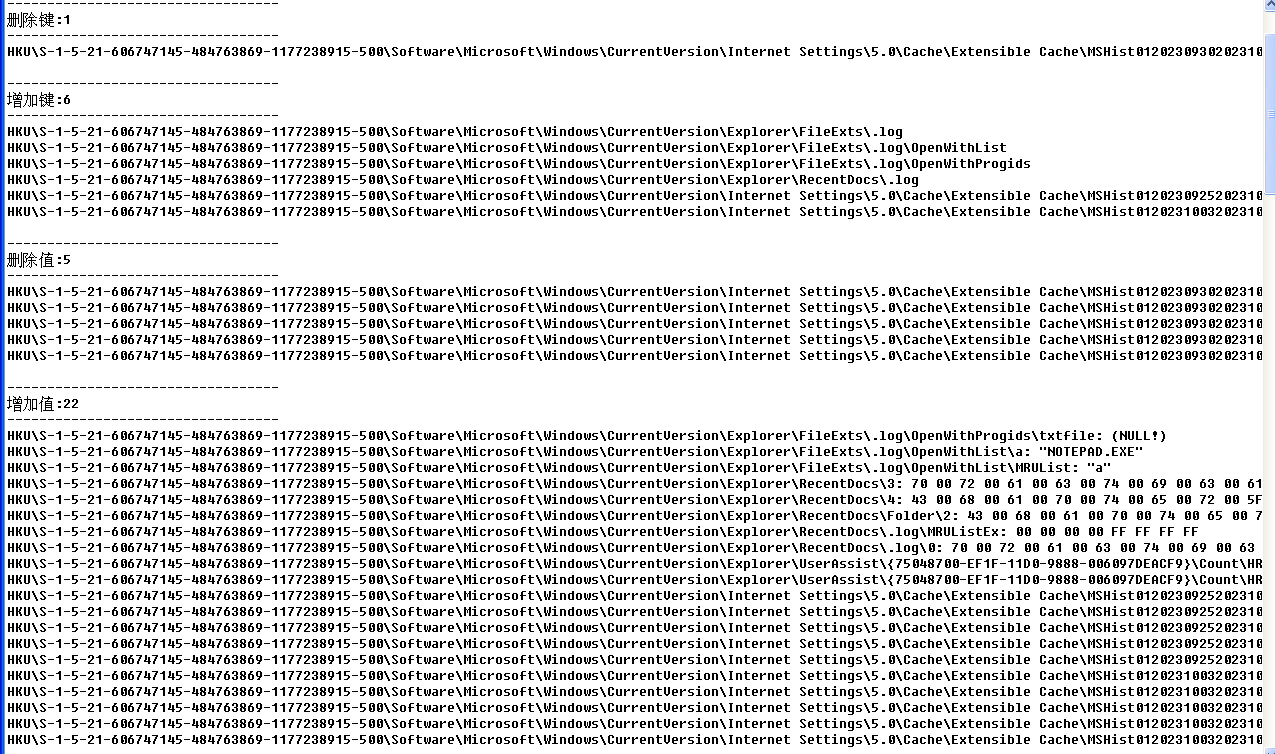


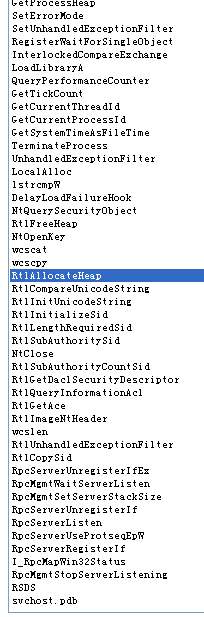
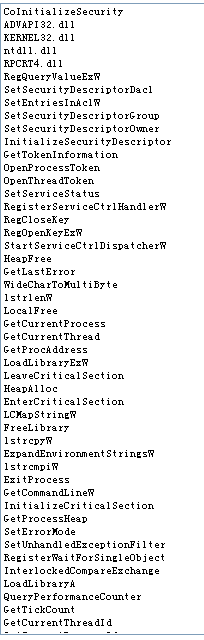
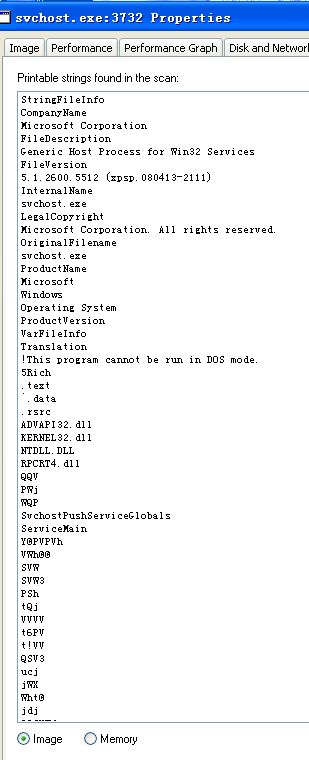
在程序运行前我们先分别打开Process Explorer，Process Monitor以及ApateDNS虚拟网络配置。

注册表并没有提供有用的信息，该程序并没有对注册表做出明显的修改。

ApateDNS也并没有捕获到明显的网络特征，只是在程序执行1分钟后进行了APR请求。







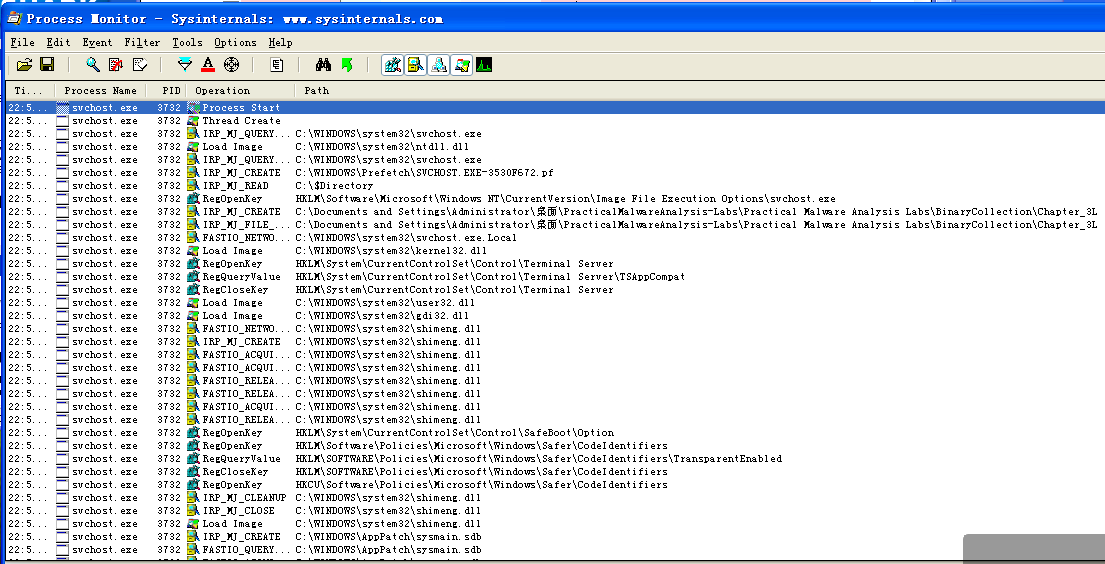
发现在执行Lab03-03.exe后，其自动结束进程并留下了一个svchost.exe子程序，并且发现每运行一下Lab03-03.exe都会留下一个“孤儿"程序svchost.exe，我们双击这个孤儿进程svchost.exe，点击strings分别查看它磁盘映像和内存映像的字符串列表。

发现它内存映像的字符串列表明显和磁盘映像的字符串列表不一样，说明对内存中的svchost做出了修改，并且下滑后发现一个practicalmalwareanalysis.log文件以及[SHIFT],[ENTER]等与键盘输入相关的字符，所以可以猜测他是一个击键记录器。

同时还含有gettickscount等字符串的出现，让我们可以猜测它上述的猜想更有一定的可能。

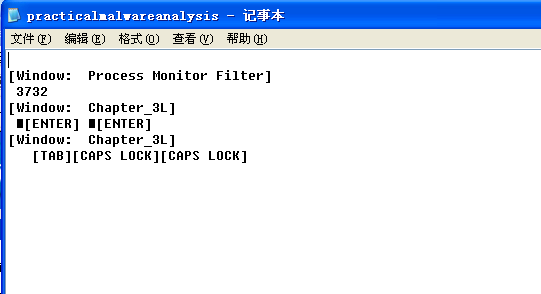
为了证明我们的猜测，我们打开一个记事本，随便输入一些信息，查看是否对键盘操作进行了记录。

在Monitor中对孤儿进程svchost(3732)进行过滤,



正如猜测的一样，svchost正在创建和写入一个名为practicalmalwareanalysis.log的文件，而这个文件就在Lab03-03.exe的同目录下。

我们打开practicalmalwareanalysis.log文件，发现里面记录的就是我们刚刚的操作。



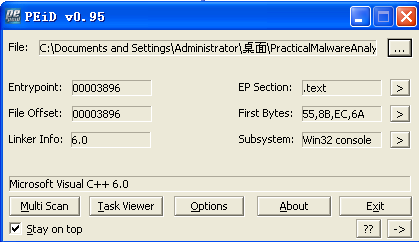
结论

Lab03-03.exe对svchost.exe执行了进程体会，来启动一个击键记录器，将所记录的内容存放在同目录下的practicalmalwareanalysis.log文件中。

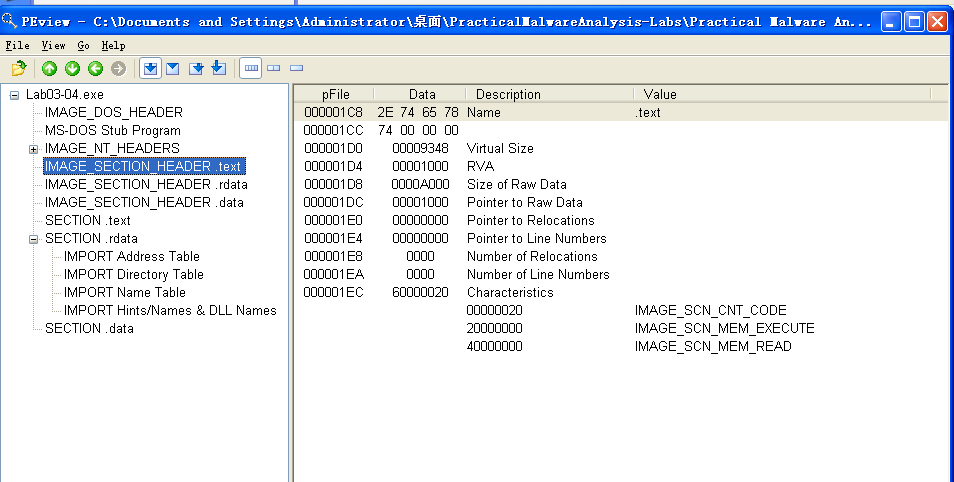
Lab03-04.exe

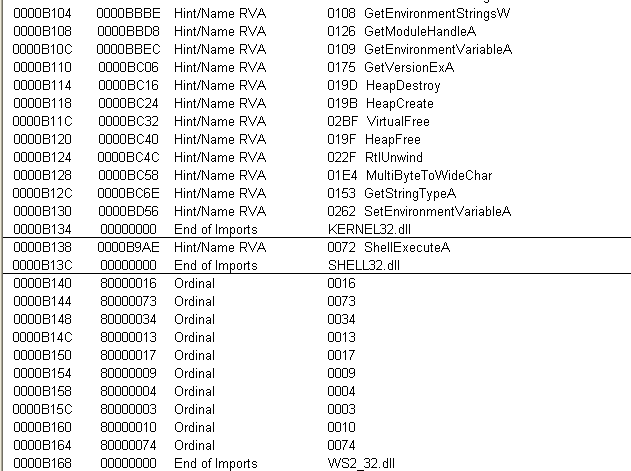
首先我们对程序进行基本的静态分析。

第一步，利用PEID检查是否存在加壳，可以明显的发现，该应用程序并不存在加壳的现象，因此可以直接分析字符串。



但是visual size与raw data的大小并不是相同的。



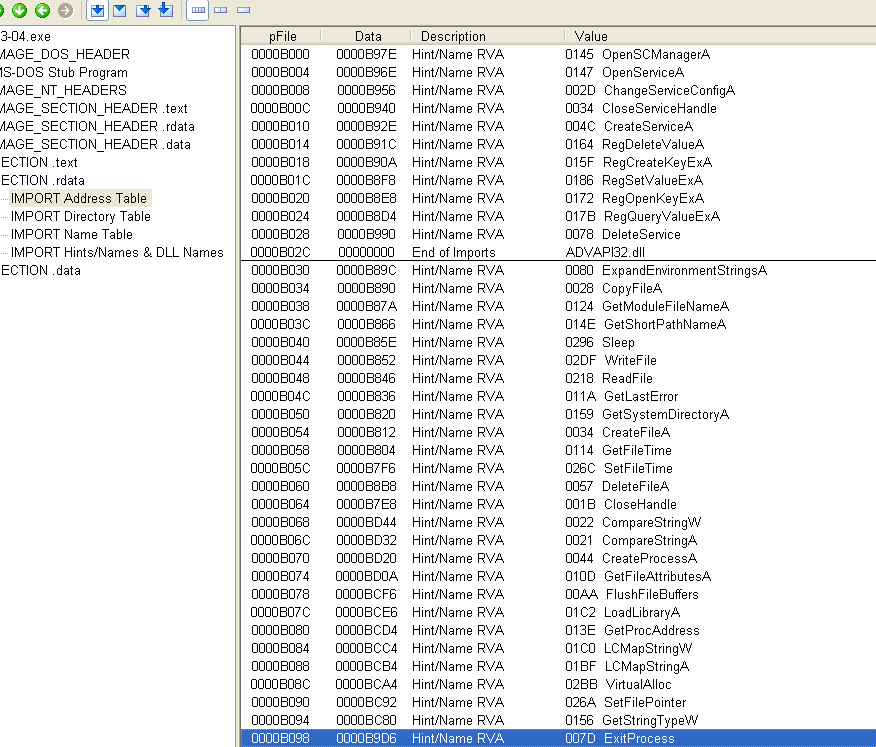


在PEVIEW当中我们还可以发现导入表具有很多复杂的操作，其中每个动态链接库都能有createfile以及参数值更新等等复杂的操作。

可以看到导入表中具有复制文件、读写文件功能。

还有创造文件、创建进程等功能。

还有个getSystemDirectoryA函数用来获取系统目录。

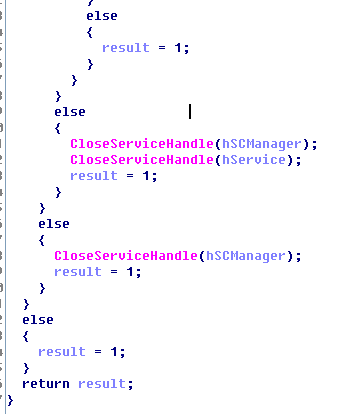


通过对字符串的分析，我们可以发现exe文件当中的字符串除了有一定的乱码存在之外，包含大量的有关函数的信息，使用了超大量的函数完成相关操作，说明可能程序与逆行的时候需要大量的函数变化存在。

另一种角度来说，我们可以发现，程序运用了大量的堆栈操作，可能有许多堆栈的创建以及压栈出栈等等复杂的操作。

程序最后还有上传和下载目标网址的操作，可能会包含对目标网址当中的文件进行上传，下载，以及getfile等等读取文件等等复杂的操作。





可以在IDA当中简单对目的文件进行分析，很明显就能发现文件当中，具有大量，判断函数，最终返回的有closeservice的函数，说明该函数涉及到对应用程序的关闭。

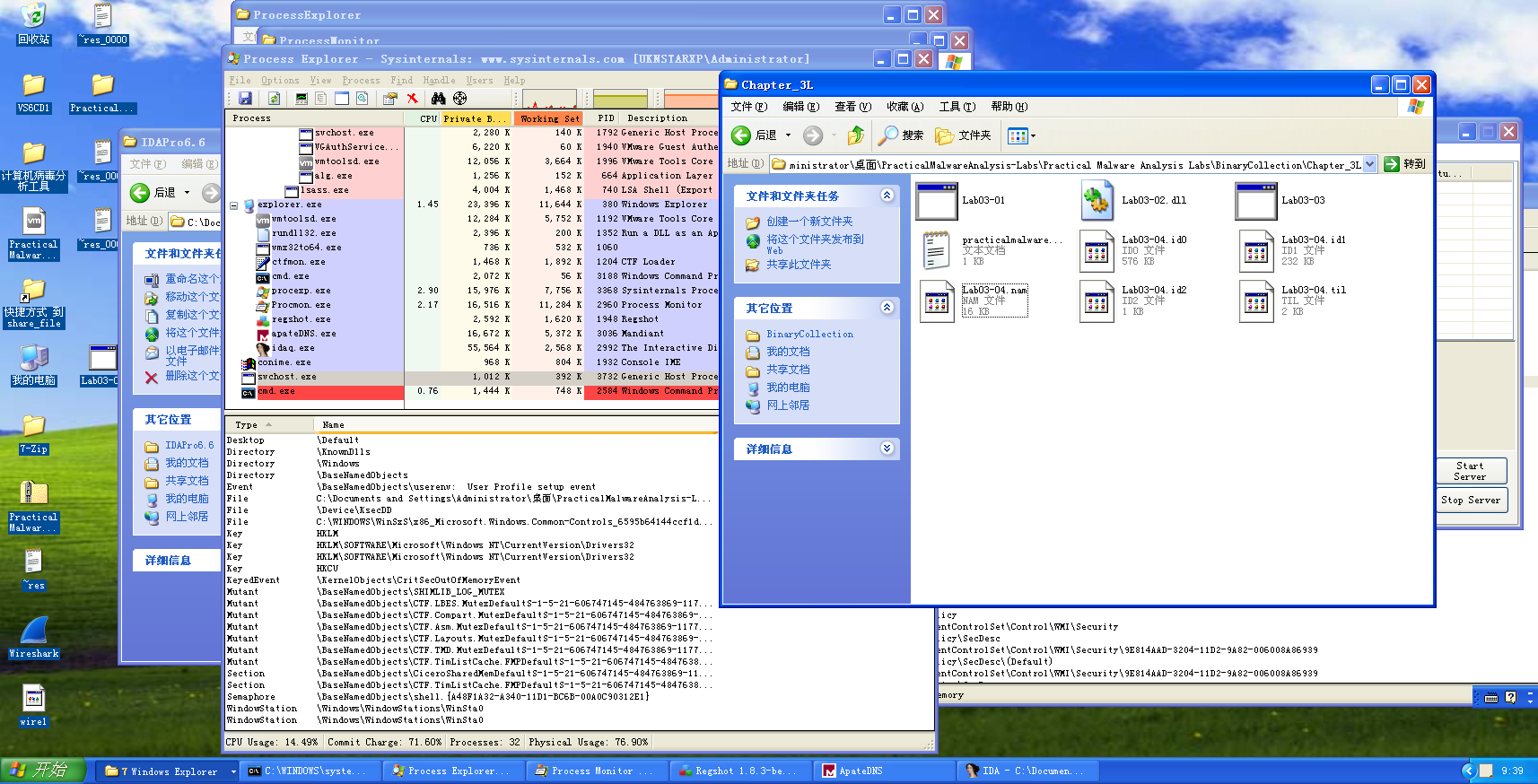


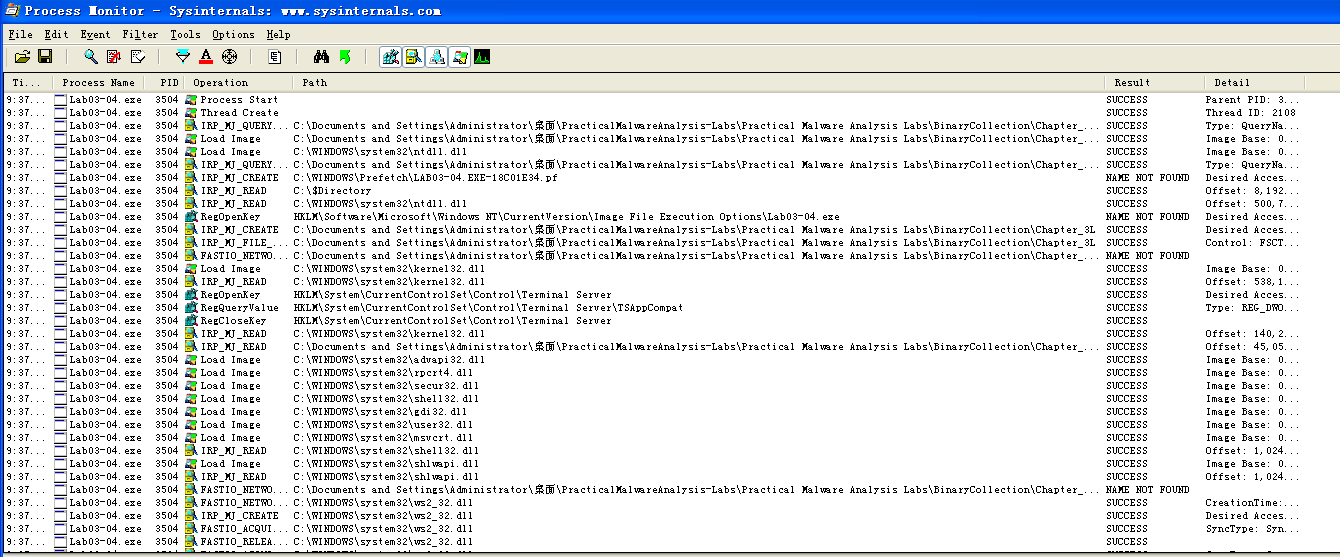
接下来我们尝试运行该应用程序，会发现该程序闪一下之后就会将自身自动删除。



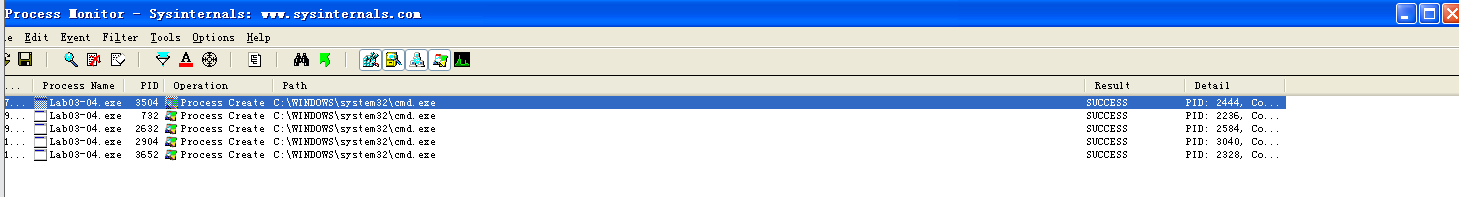
我们可以看到的是，程序在process explorer当中迅速启动了cmd，之后迅速将自身删除。

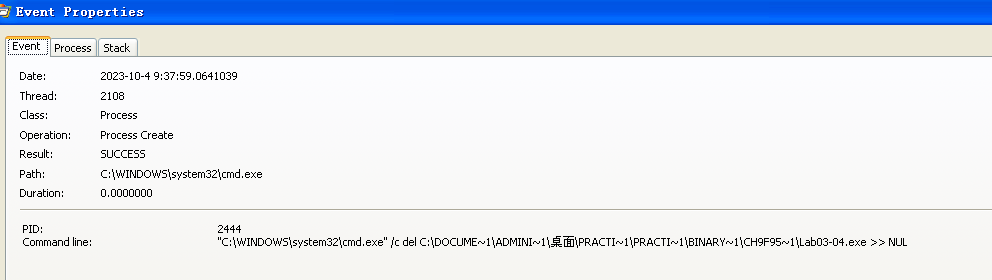
进程自删除造成了动态分析无法有效实施，无法判断其运行状态。





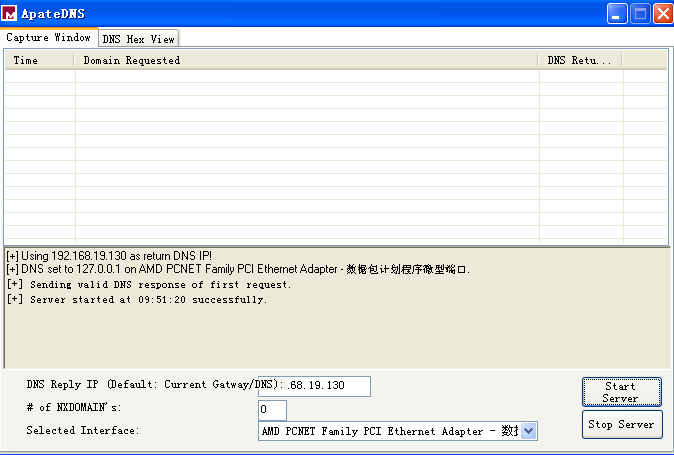
运行process monitor进行分析，发现程序运行痕迹。



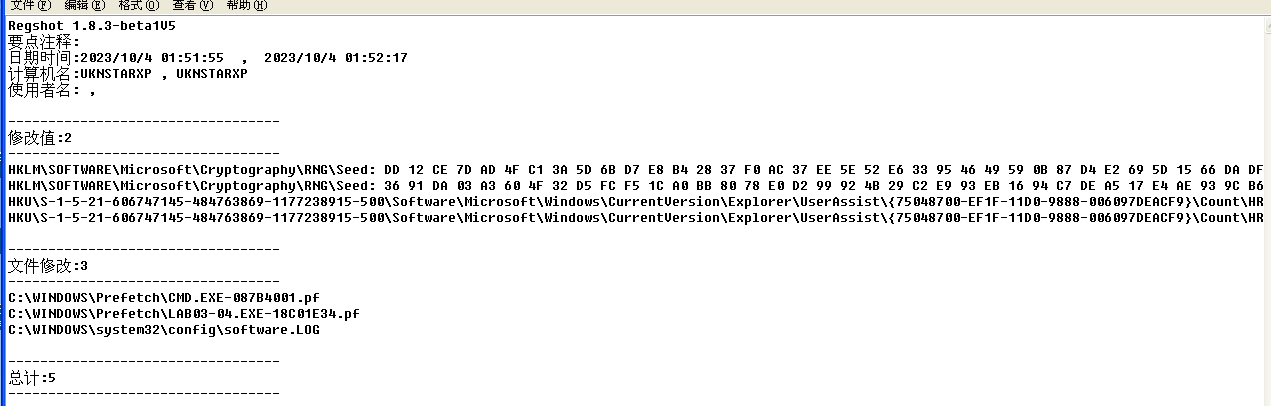


接下来我们发现存在许多Process Create的操作，这些操作打开了cmd。

其中的命令我们发现是"C:\WINDOWS\system32\cmd.exe" /c del C:\DOCUME~1\ADMINI~1\桌面\PRACTI~1\PRACTI~1\BINARY~1\CH9F95~1\Lab03-04.exe >> NUL



ApateDNS当中提取不到任何有效的操作。



regshot当中我们可以看到相关打开cmd的操作，但是并不能发现其他更有价值的信息。

最后我们使用如下的yara规则扫描文件夹（注意要在64位操作系统上，这里选择win8.1)

rule Lab3

{

meta:

description = "rules for Lab3"

date = "202x/xx/xx"

strings:

$a = "vmx32to64" wide ascii

$b = "serve.html" wide ascii

$c = "http://www.malwareanalysisbook.com" wide ascii

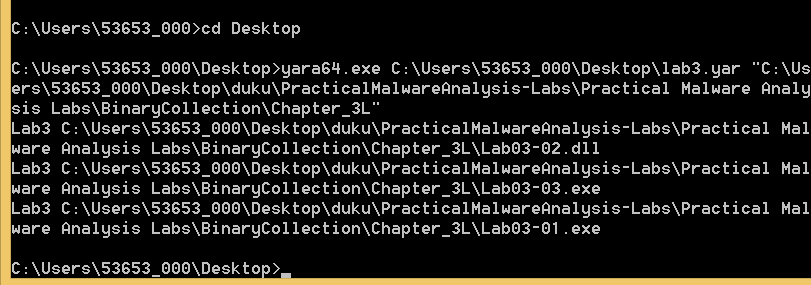
$d = "svchost" wide ascii

$e = "practicalmalwareanalysis.log" wide ascii

condition:

any of them

}



1. **实验结论及心得体会**

**通过本次实验，我初步领会到了基本的动态分析的方法，步骤，以及大致的使用操作工具，我还熟悉了大量的工具的特殊用法，对于process monitor process explorer等等的工具有很好的熟悉掌握，我们接下来还将继续学习使用这些工具进行更多的学习。**

**我还重新熟悉掌握了基本静态分析软件的更多用法，对PE文件头对于加壳和查壳的方法有了更多的掌握。**

**除了之前使用过IDA进行程序分析之外，我还学会了利用各种静态和动态分析软件查找程序运行的蛛丝马迹，之后在其中寻找特殊的值得注意的问题，汇总得出最终的结论。**

**我还学会了在虚拟机之间配置内部网络，通过配置内部网络，可以很方便的将kali虚拟机当中的一些功能在windows上成功使用，并且在网络连接通讯的过程当中，我们还可以很好对程序对网络的请求进行监听，得知目的软件的联网功能以及通过wireshark抓包查找是否出现了一定的问题。**

**最后我还通过regshot得知了程序变化前后会修改注册表等当中的一些文件的键以及增删等等其他操作，通过regshot可以有效的监听这些程序的运行。**