恶意代码分析与防治技术实验报告

实验三

网络空间安全学院

信息安全专业

2211985 李佳璐

一、实验目的

1. 使用动态分析技术完成对给定病毒样本的分析；
2. 熟悉虚拟机、Yara等的操作。

二、实验原理

为了保护本机免受恶意代码攻击，本次实验主体在虚拟机上完成，以下为相关环境：

1. 已关闭病毒防护的Windows11
2. 在VMware上部署的Windows XP虚拟机

三、实验过程

Lab03-01.exe

问题：

使用动态分析基础技术来分析在Lab03-01.exe文件中发现的恶意代码。

1. 找出这个恶意代码的导入函数与字符串列表？

   这个恶意代码看起来是被加过壳的。使用Dependency Walker工具可见唯一的导入函数是ExitProcess，虽然出现字符串大多是明文，而未被混淆过。

2. 这个恶意代码在主机上的感染迹象特征是什么？

   使用Process Explorer可见该恶意代码创建了一个名为`WinVMX32`的互斥量，并复制自身到`C:\Windows\System32\vmx32to64.exe`，并安装自己到系统自启动项中，通过创建注册表键值`HKLMSOFTWAREMicrosoft\Windows\CurrentVersion\Run\VideoDriver`，并将其设置为复制副本的位置。

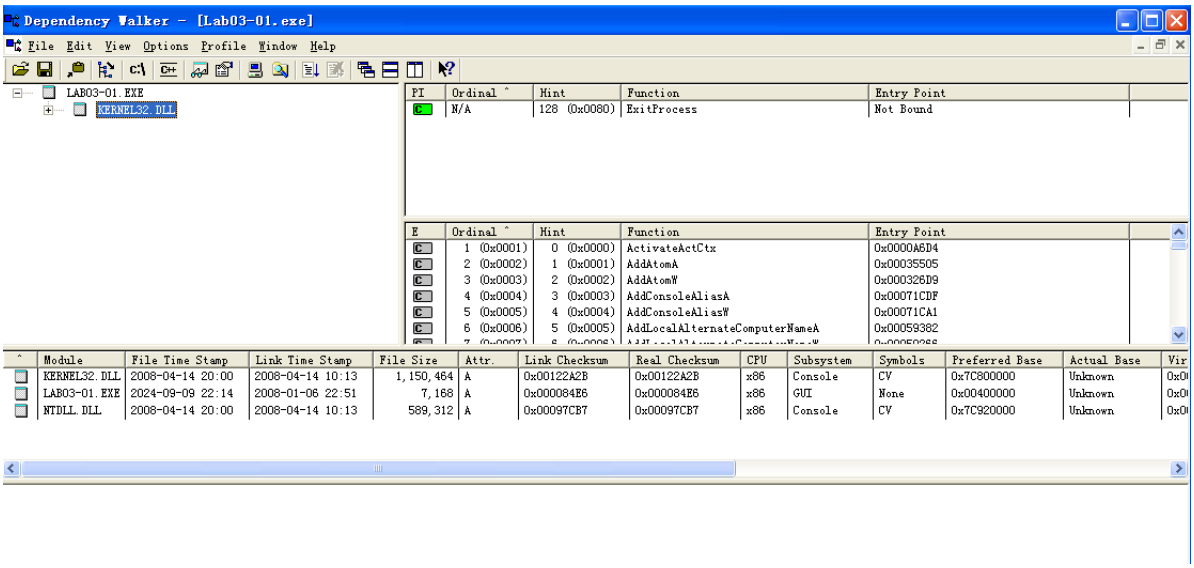
3. 这个恶意代码是否存在一些有用的网络特征码？如果存在，它们是什么？

   进一步使用Wireshark等工具，发现恶意代码在进行`www.practicalmalwareanalysis.com`的域名解析后，持续地广播大小为 256字节的数据包，其中包含看似随机的二进制数据。

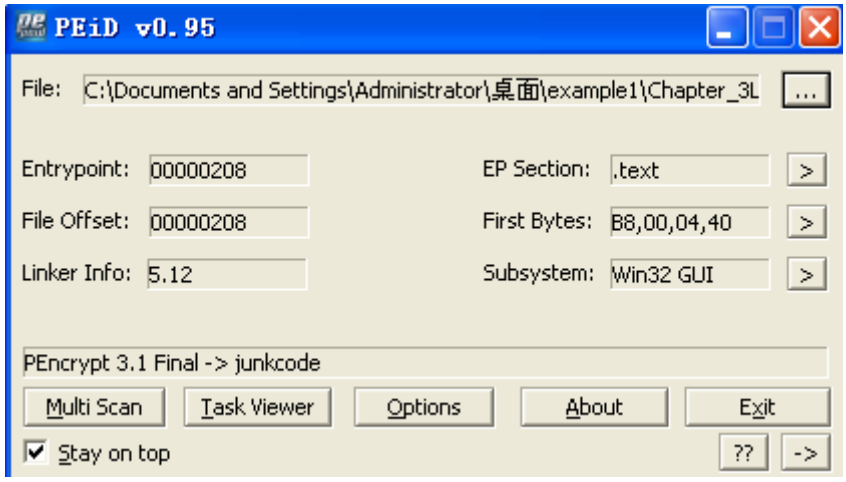
具体解析过程如下：

- **静态分析**

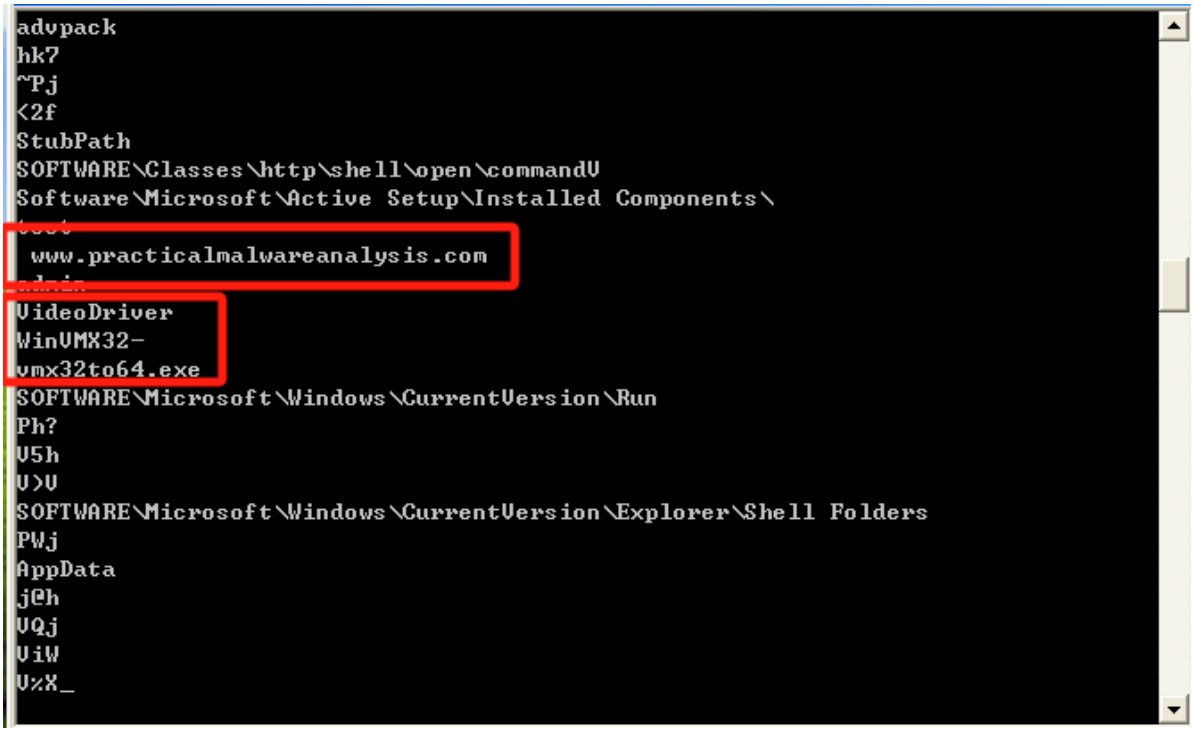
使用Dependency Walker打开文件，查看其导入函数。可以看到，仅有一个ExitProcess。而单凭这一个函数无法使这个程序正常运行，怀疑该文件出现了加壳。



使用PEiD打开，检测加壳信息。如图所示，可见为`PEncrypt 3.1 Final -> junkcode`加壳



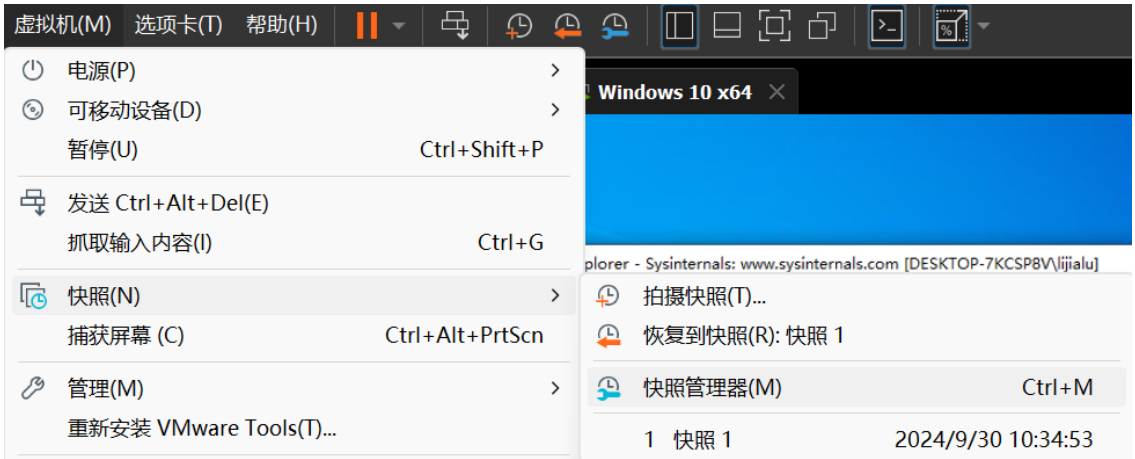
使用string试图查看字符串，因为加壳，预期是看不到明文字符串的。但发现了可能暗示恶意代码功能的字符串，比如注册表位置、域名、WinVMX32、VideoDriver、vmx32to64.exe。



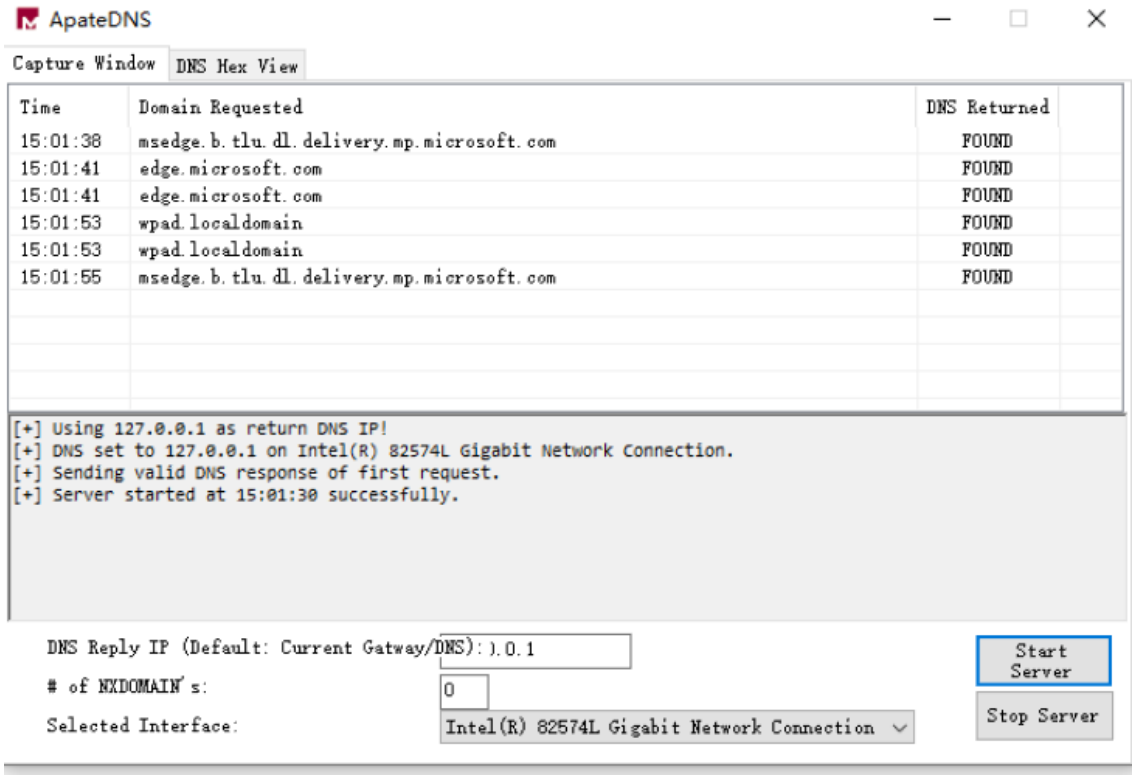
- **动态分析**

在进行动态分析时，需对虚拟机做如下处理：

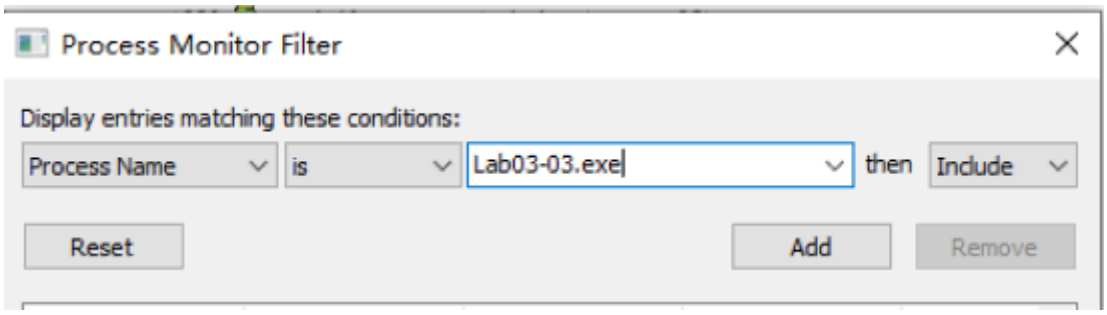
1. 对VMware进行快照，便于恢复到运行前的状态



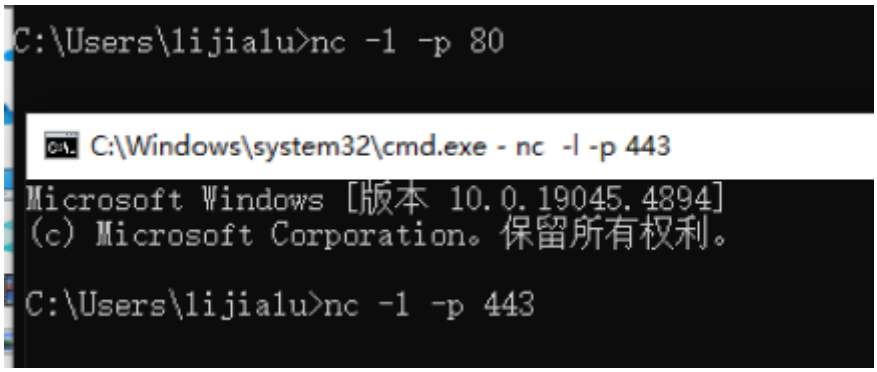
2. 启动ApateDNS，将DNS Reply IP设置为127.0.0.1



3. 启动Process Monitor，并按照实验要求设置过滤条件



4. 启动Process Explorer
5. 启动netcat：nc-l -p XXX，监听端口80和443



6. 启动wireShark抓取数据包

运行恶意代码后，在Process Explorer中检查进程。

单击Lab03-01.exe进程项目和选择`View → Lower Pane View → Handles`。在这个视图中，可以看到，恶意代码已经创建了一个名为WinVMX32的互斥量。

Lab03-01.exe	816 K	2,260 K	1956	
conime.exe	972 K	3,208 K	5496 Console IME	Microsoft Corporation

Type	Name
Desktop	\Default
Directory	\KnownDlls
Directory	\Windows
Directory	\BaseNamedObjects
File	C:\Documents and Settings\Administrator\桌面\恶意代码\上机实验样本\Chapte...
File	\Device\KsecDD
File	\Device\Tcp
File	\Device\Tcp
File	\Device\Ip
File	\Device\Ip
File	\Device\Ip
File	\Device\Tcp
Key	HKLM
Key	HKLM\SYSTEM\ControlSet001\Services\WinSock2\Parameters\Protocol_Catalog9
Key	HKLM\SYSTEM\ControlSet001\Services\WinSock2\Parameters\NameSpace_Catalog5
Key	HKLM\SYSTEM\ControlSet001\Services\Tcpip\Linkage
Key	HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters
Key	HKLM\SYSTEM\ControlSet001\Services\NetBT\Parameters\Interfaces
Key	HKLM\SYSTEM\ControlSet001\Services\NetBT\Parameters
[illegible]	[illegible]oryEvent
Mutant	\BaseNamedObjects\WinVMX32
[illegible]	[illegible]
Thread	Lab03-01.exe(1956): 6012
WindowStation	\Windows\WindowStations\WinSta0
WindowStation	\Windows\WindowStations\WinSta0

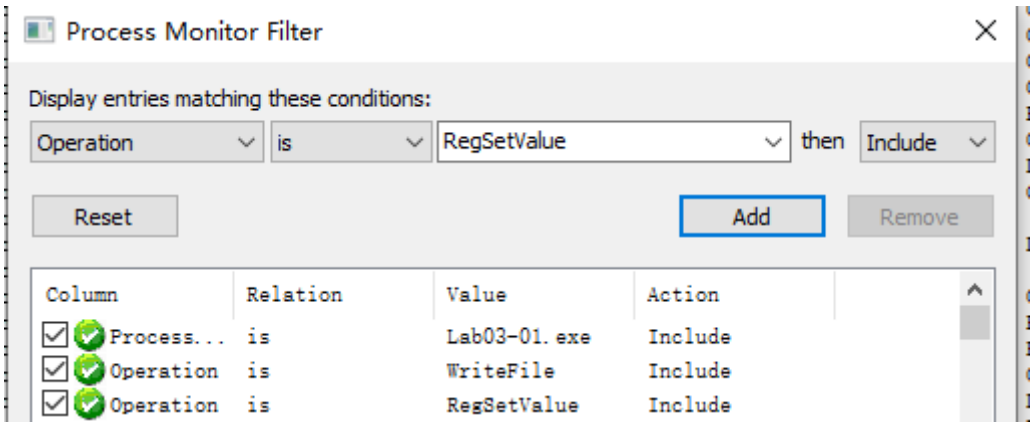
另外选择`View → Lower Pane View → DLL`,看到恶意代码动态装载的DLL文件，如ws2_ 32.dll和wshtcpip.dll，这意味着它具有联网功能。

Lab03-01.exe	816 K	2,260 K	1956	
conime.exe	972 K	3,208 K	5496 Console IME	Microsoft Corporation

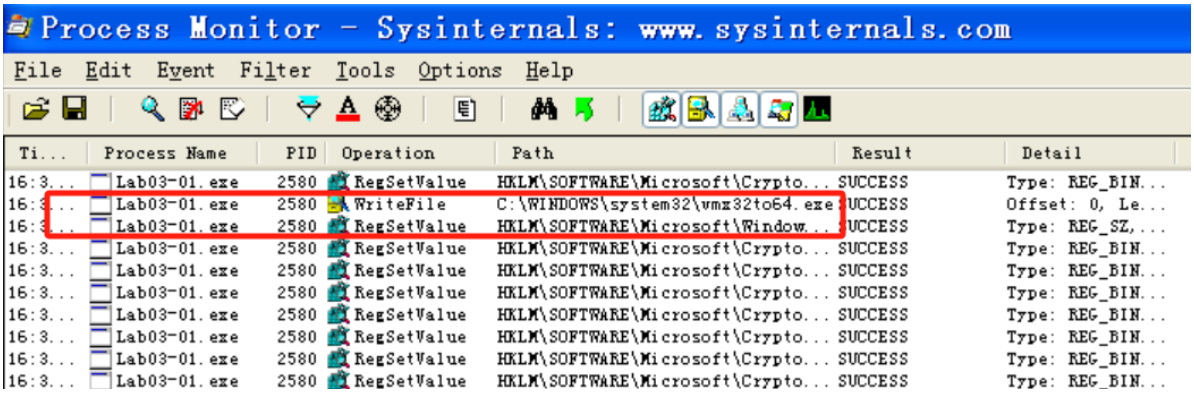
Name	Description	Company Name	Path
imm32.dll	Windows XP IMM32 API Clien...	Microsoft Corporation	C:\WINDOWS\system32\imm32.dll
iphlpapi.dll	IP Helper API	Microsoft Corporation	C:\WINDOWS\system32\iphlpapi.dll
kernel32.dll	Windows NT BASE API Client...	Microsoft Corporation	C:\WINDOWS\system32\kernel32.dll
Lab03-01.exe			C:\Documents and Settings\Administrator\桌面\...
locale.nls			C:\WINDOWS\system32\locale.nls
lpk.dll	Language Pack	Microsoft Corporation	C:\WINDOWS\system32\lpk.dll
msvcrt.dll	Windows NT CRT DLL	Microsoft Corporation	C:\WINDOWS\system32\msvcrt.dll
mswsock.dll	Microsoft Windows Sockets ...	Microsoft Corporation	C:\WINDOWS\system32\mswsock.dll
ntdll.dll	NT Layer DLL	Microsoft Corporation	C:\WINDOWS\system32\ntdll.dll
ole32.dll	Microsoft OLE for Windows	Microsoft Corporation	C:\WINDOWS\system32\ole32.dll
rasadhlp.dll	Remote Access AutoDial Helper	Microsoft Corporation	C:\WINDOWS\system32\rasadhlp.dll
rpcrt4.dll	Remote Procedure Call Runtime	Microsoft Corporation	C:\WINDOWS\system32\rpcrt4.dll
secur32.dll	Security Support Provider ...	Microsoft Corporation	C:\WINDOWS\system32\secur32.dll
setupapi.dll	Windows Setup API	Microsoft Corporation	C:\WINDOWS\system32\setupapi.dll
sortkey.nls			C:\WINDOWS\system32\sortkey.nls
sorttbls.nls			C:\WINDOWS\system32\sorttbls.nls
unicode.nls			C:\WINDOWS\system32\unicode.nls
user32.dll	Windows XP USER API Client...	Microsoft Corporation	C:\WINDOWS\system32\user32.dll
usp10.dll	Uniscribe Unicode script p...	Microsoft Corporation	C:\WINDOWS\system32\usp10.dll
version.dll	Version Checking and File ...	Microsoft Corporation	C:\WINDOWS\system32\version.dll
winrnr.dll	LDAP RnR Provider DLL	Microsoft Corporation	C:\WINDOWS\system32\winrnr.dll
[illegible]	Win32 LDAP API DLL	Microsoft Corporation	C:\WINDOWS\system32\wldap32.dll
ws2_32.dll	Windows Socket 2.0 32-Bit DLL	Microsoft Corporation	C:\WINDOWS\system32\ws2_32.dll
[illegible]	Windows Socket 2.0 Helper ...	Microsoft Corporation	C:\WINDOWS\system32\ws2help.dll
[illegible]	Windows Sockets Helper DLL	Microsoft Corporation	C:\WINDOWS\system32\wshbth.dll
wshtcpip.dll	Windows Sockets Helper DLL	Microsoft Corporation	C:\WINDOWS\system32\wshtcpip.dll

使用procmon工具寻找额外的信息。

设置三个过滤器:一个是对进程名称的过滤(显示Lab03-01.exe对系统所做的)，和两个操作上的过滤。包含了RegSetValue和writeFile，来查看恶意代码对文件系统和注册表的修改操作。

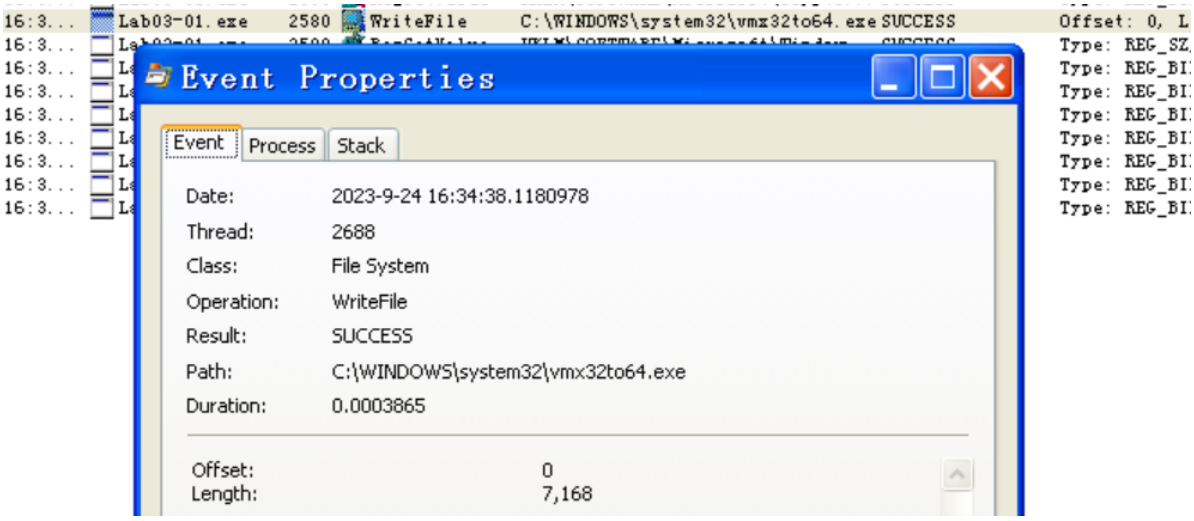


设置完过滤器之后，单击Apply按钮，便可看到过滤后的结果。显示的条目从数千减少到只有10条。只有一个WriteFile操作的条目，并有9个关于RegSetValue操作的条目。

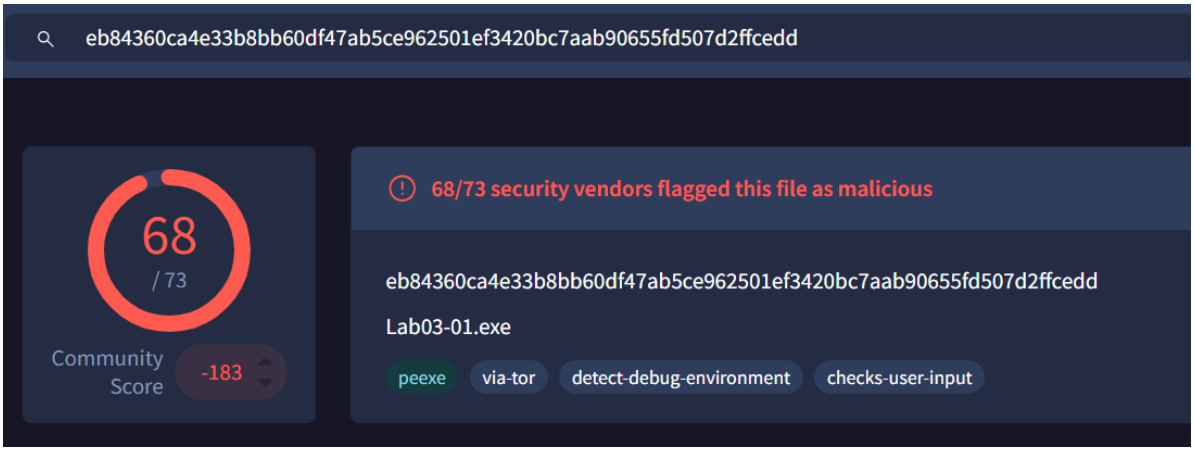


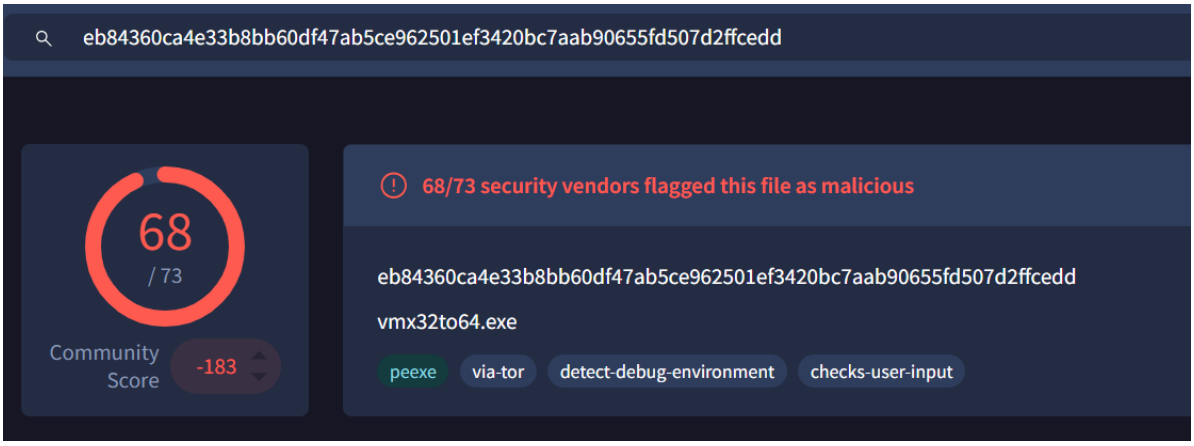
除红框两项，其余项`HKLM\SOFTWARE\Microsoft\cryptography\RNG\Seed`键值上的`RegSetValue`操作是典型的噪声，因为随机数发生器的种子会有软件在注册表中不停地更新。

其余其中一项是writeFile操作。双击该条目，显示记录会告诉我们，恶意代码往`C:\WINDOWS\System32\vmx32to64.exe`中写了7168字节。这恰好是`Lab03-01.exe`文件的大小。



打开Windows资源管理器浏览到这个位置，使用virustotal查看对比`Lab03-01.exe`和`vmx32to64.exe`：

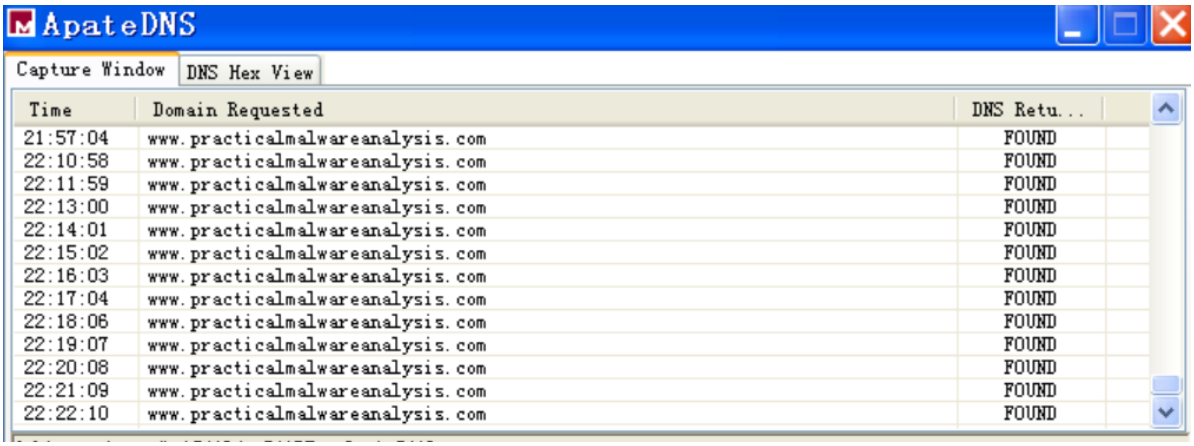




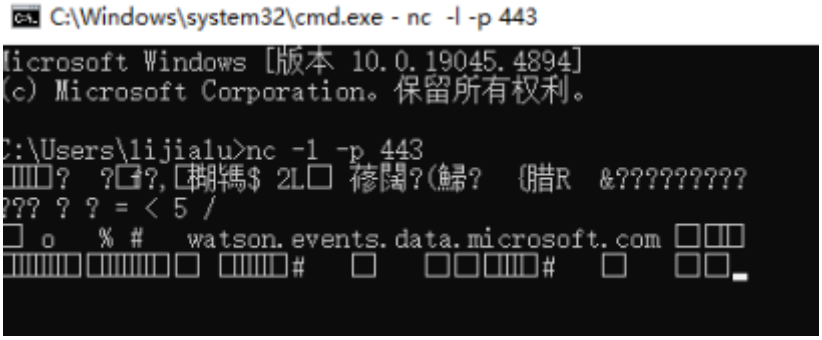
可以看到这个新创建的文件有着与Lab03-01.exe相同的MD5哈希值，两者分析结果没有差别。可以看出该恶意代码已经复制本身到这个文件系统位置上。这是一个非常有用的感染主机迹象特征，因为它使用了一个硬编码的文件名。

另外一项RegSetValue可见文件还将数据写入注册表项，命名为VideoDriver。分析可知，该文件将vmx32to64.exe写入到了开机启动的项目中。

检查ApateDNS，来查看恶意代码是否执行了DNS请求。通过检查输出，我们看到了一个对`www.practicalmalwareanalysis.com`域名的请求，这匹配到了字符串列表中的显示。

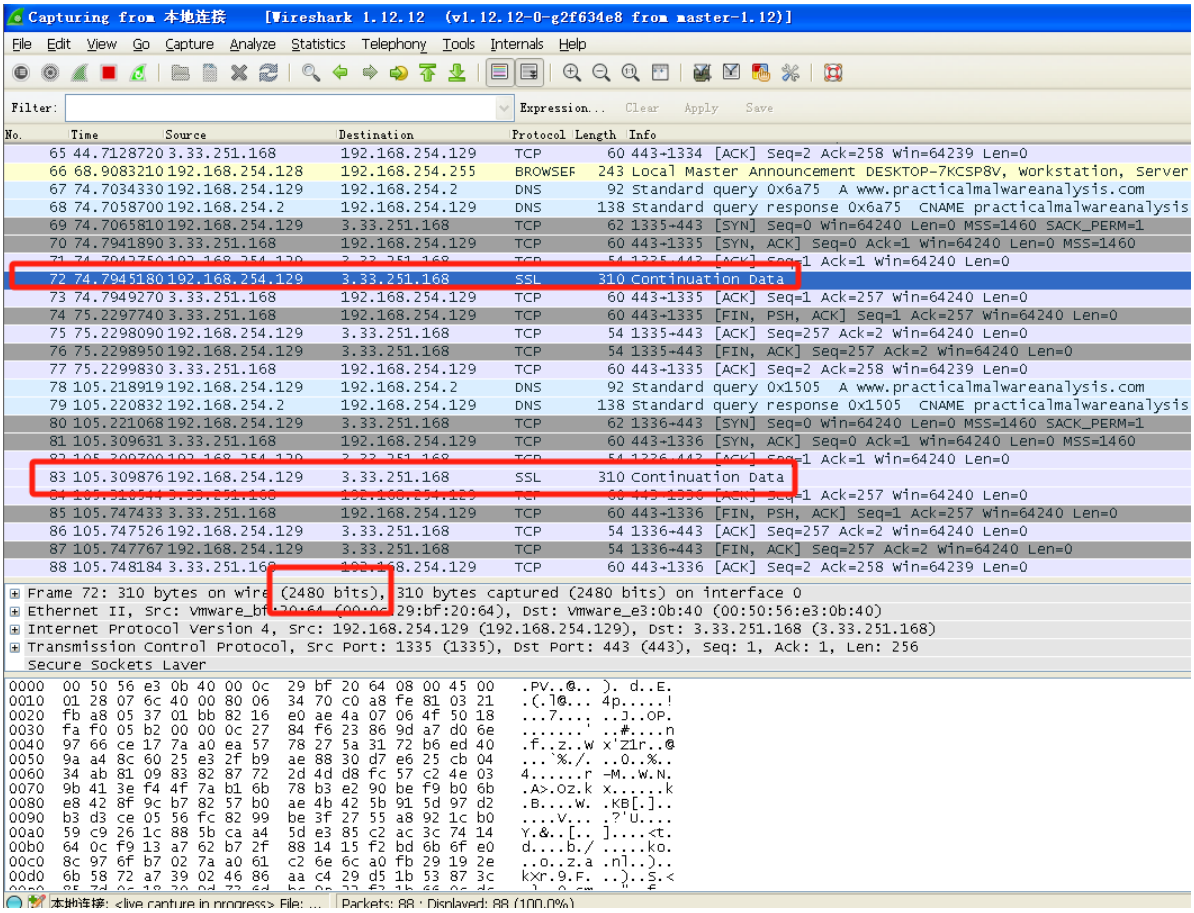


检查Netcat的结果来完成网络分析过程



恶意代码看起来在443端口上发出了一些信号（正在使用Netcat监听着80和443端口），多次运行这个测试，每次的数据看起来都是随机的。

对Wireshark结果的后续分析可见，这些信号数据包是固定长度的(256字节)，看起来包含的是一些随机数据，而与普通在443端口上跑的SSL协议没有关联。



Lab03-02.dll

问题：

使用动态分析基础技术来分析在Lab03-02.dll文件中发现的恶意代码

1. 你怎样才能让这个恶意代码自行安装？

   利用 `rundll32.exe` 工具，使用命令 `rundll32.exe Lab03-02.dll,installA`，运行恶意代码导出的 `installA` 函数，便可将恶意代码安装为一个服务。

2. 在安装之后，你如何让这个恶意代码运行起来？

   使用net命令执行 `net start IPRIP`，便可启动恶意代码安装的服务。

3. 你怎么能找到这个恶意代码是在哪个进程下运行的？

   使用ProcessExplorer的 `Find Dll` 功能来搜索 Lab03-02.dll，可以看到该恶意代码是在svchost.exe进程下运行的。

4. 你可以在 procmon 工具中设置什么样的过滤器，才能收集这个恶意代码的信息？

   在procmon工具中，可以使用上题中在Process Explorer 发现的 PID 进行过滤。

5. 这个恶意代码在主机上的感染迹象特征是什么？

   默认情况下，恶意代码将安装为IPRIP服务，显示的服务名称为 `IntranetNetworkAwareness(INA+)`，描述为 *“Depends INA+, Collects and stores network configuration and location information, and notifies applications when this information changes ”* 它将自身持久地安装在注册表中 `HKLM\SYSTEM CurrentControlSet Services IPRIP\Parameters\ServiceDll:%CurrentDirectory%Labe3-02.dll`。如果将Lab03-02.dll 重命名为其他文件名，如 `maware.dll`，然后这个恶意代码就会把 `malware.dll` 写入到注册表项中，而不是使用名称 `Lab03-02.dll`。

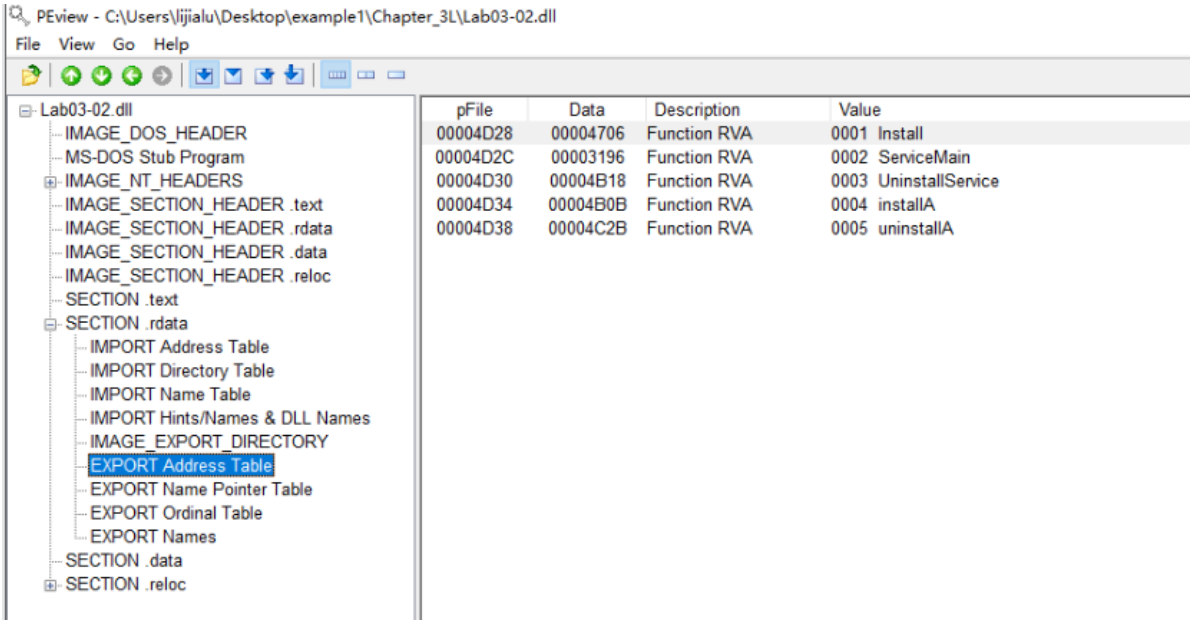
6. 这个恶意代码是否存在一些有用的网络特征码？

   恶意代码申请解析域名 `practicalmalwareanabysis.com`，然后通过80端口连接到这台主机，使用的协议是HTTP协议。它在做一个GET请求 `serve.html`，使用的用户代理为 `dell-cdce31a6e0 windows XP6.11`。

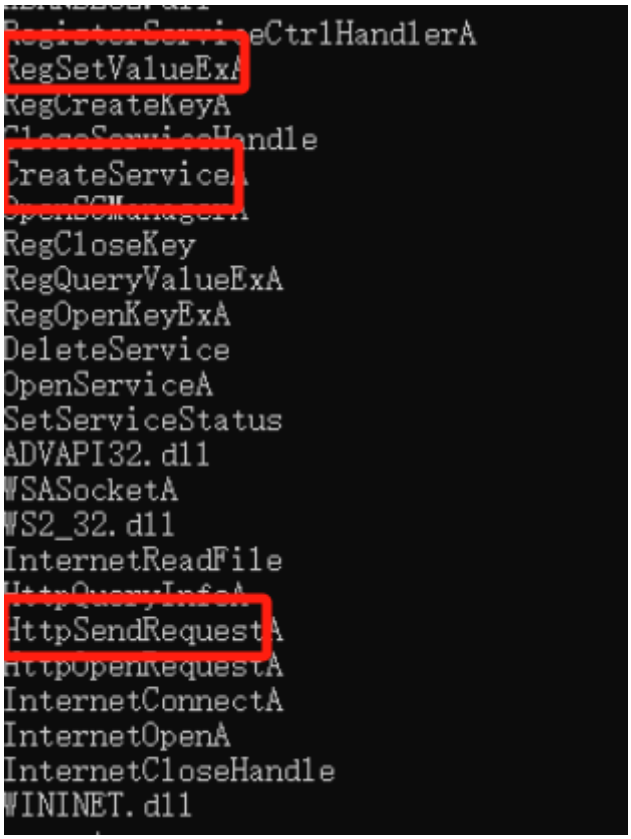
具体解析过程如下：

- **静态分析**

使用PEview工具查看这个样本的PE文件结构，可见这个DLL有五个导出函数。从导出函数ServiceMain表明，这个恶意代码需要安装成一个服务，使其能够正常运行。



使用strings工具查看字符串，其中包含恶意代码的导入函数，这里包括了一些服务操作函数，比如 `CreateService`，以及一些注册表操作函数，如函数 `RegSetValueEx`。导入的网络操作函数，如 `HttpSendRequest` 等，表明恶意代码使用了HTTP。



此外，还可以看到一些字符串，包括注册表位置、域名、一些独特的字符串如`IPRIP`和`serve.html`，以及一些编码的字符串。

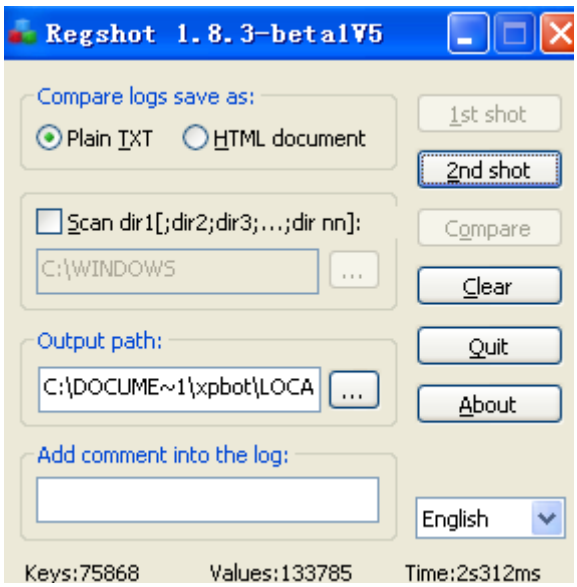
```
C:\Windows\system32\cmd.exe
Parameters
Type
Start
ObjectName
LocalSystem
ErrorControl
DisplayName
Description
Depends INA+, Collects and stores network configuration and location information, and notifies
formation changes.
ImagePath
%SystemRoot%\System32\svchost.exe -l
SYSTEM\CurrentControlSet\Services\
CreateService(%s) error %d
Intranet Network Awareness (INA+)
%SystemRoot%\System32\svchost.exe -k netsvcs
OpenSCManager()
You specify service name not in Svchost//netsvcs, must be one of following:
RegQueryValueEx(Svchost\netsvcs)
netsvcs
RegOpenKeyEx(%s) KEY_QUERY_VALUE success.
RegOpenKeyEx(%s) KEY_QUERY_VALUE error .
SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost
IPRIP
uninstall success
OpenService(%s) error 2
OpenService(%s) error 1
uninstall is starting
.?AVtype_info@@
080@0`0m0r0
```

- **动态分析**

将恶意代码安装为服务

静态分析基础技术的结果可知这个恶意代码需要使用导出函数installA将自身注册为一个服务。

故使用这个函数来试图安装恶意代码，在做安装恶意代码之前，需使用Regshot工具，来对注册表做个基线快照。



并且使用ProcessExplorer来监控在系统上运行的进程。

当我们设置好Regshot和Process Explorer之后，我们使用rundll32.exe来安装恶意代码，使用如下命令:

```
C:\Documents and Settings\Administrator\桌面\恶意代码\上机实验样本\Chapter_3L>ru
ndll32.exe Lab03-02.dll,installA
```

确保rundll32.exe已经终止后，使用Regshot工具做第二个快照，输出的Regshot结果如下

```
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Enum\0: "Root\LEGACY_IPRIP\0000"
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Enum\Count: 0x00000001
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Enum\NextInstance: 0x00000001
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Security\Security: 01 00 14 80 90 00 00 00 9C 00 00 00 14 00 00 00 30 00 00 00 02 00 1C 00 01 00 00 00 02 80 14 00 FF 01 0F 00 01 01 00 00 00 00 00 01 00
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Parameters\ServiceDll: "C:\Documents and Settings\Administrator\桌面\恶意代码\上机实验样本\Chapter_3L\Lab03-02.dll"
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Type: 0x00000020
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Start: 0x00000002
HKLM\SYSTEM\ControlSet001\Services\IPRIP\ImagePath: "%SystemRoot%\System32\svchost.exe -k netsvcs"
HKLM\SYSTEM\ControlSet001\Services\IPRIP\DisplayName: "Intranet Network Awareness (INA+)"
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Description: "Depends INA+, Collects and stores network configuration and location information, and notifies applications when this information changes."
HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_IPRIP\0000\Control\*NewlyCreated*: 0x00000000
HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_IPRIP\0000\Control\ActiveService: "IPRIP"
HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_IPRIP\0000\Service: "IPRIP"
HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_IPRIP\0000\Legacy: 0x00000001
HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_IPRIP\0000\ConfigFlags: 0x00000000
HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_IPRIP\0000\Class: "LegacyDriver"
```

1. 首先显示了恶意代码将自身安装为IPRIP服务，由于这个恶意代码是个DLL文件，它依赖于一个可执行文件来执行它。
2. 另外看到ImagePath被设置为svchos.exe，这意味着，这个恶意代码将会在一个svchost.exe进程中启动。
3. 另外信息包括DisplayName和Description，可以作为识别这个恶意服务的独特指纹特征。

已知上述信息，再次检查strings工具显示的字符串列表

```
%SystemRoot%\System32\svchost.exe -k netsvcs
You specify service name not in Svchost//netsvcs, must be one of following:
RegQueryValueEx(Svchost\netsvcs)
netsvcs
RegOpenKeyEx(%s) KEY_QUERY_VALUE success.
SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost
IPRIP
uninstall success
OpenService(%s) error 2
OpenService(%s) error 1
uninstall is starting
.?AVtype_info@@
080@0^0m0r0
```

仔细地检查字符串列表，我们可以看到`SOFTWARE\Microsoft\windows NT\CurrentVersion\SvcHost`和一个消息`You specify service name not in Svchost//netsvcsmustbe one of following`。

检查`Svchost//\Svchost\netsvcs`注册表项可以看到其他可能被使用的服务名称，比如`6to4 AppMgmt`。运行`Lab03-02.dll`，`installA6to4`将安装恶意代码为6to4服务，而不再是前面所列出的IPRIP服务。

以上即可作为该恶意代码在主机上的感染迹象特征。

运行恶意代码

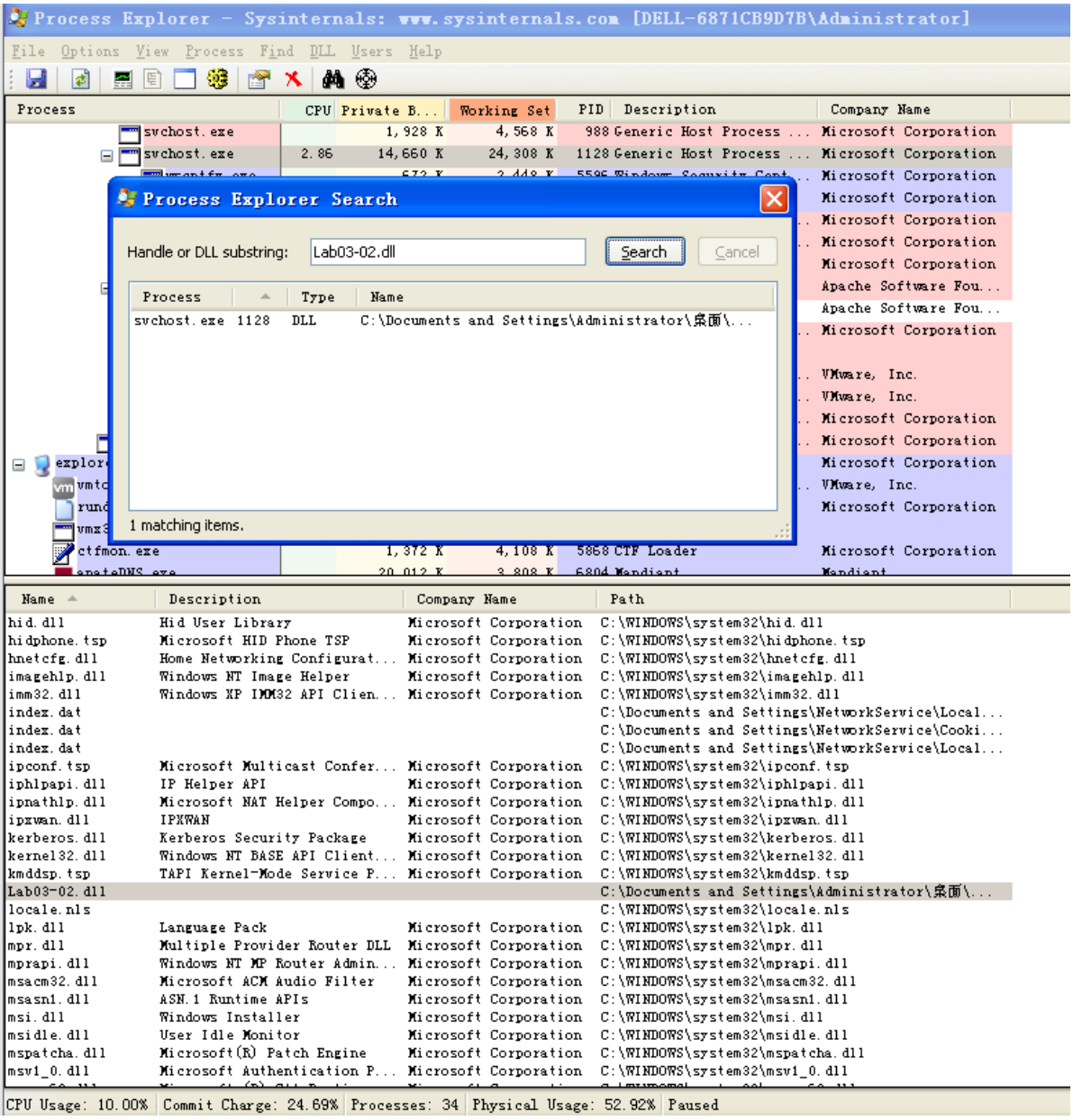
在将恶意代码安装为服务之后，就可以启动运行它了。在运行之前，首先需要配置动态分析基础技术工具。

- 运行procmon工具(并清除所有事件)，启动ProcessExplorer，以及配置一个虚拟网络，包括ApateDNS、监听端口80的Netcat。

这个恶意代码安装为IPRIP服务，可以使用Windows中的net命令来启动

```
C:\Documents and Settings\Administrator\桌面\恶意代码\上机实验样本\Chapter_3L>net start IPRIP
Intranet Network Awareness (INA+) 服务正在启动 .
Intranet Network Awareness (INA+) 服务已经启动成功。
```

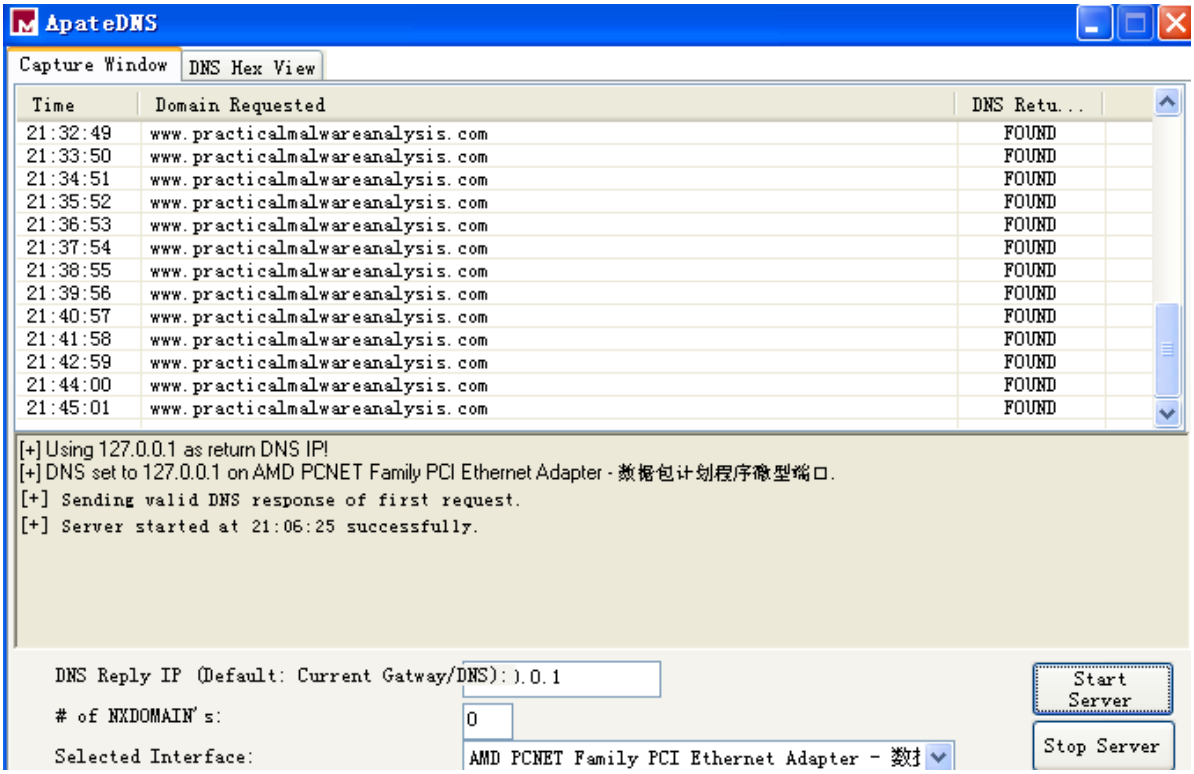
打开ProcessExplorer中，尝试寻找出恶意代码运行的进程，输入Lab03-02.dll进行Search。可以看到`Lab03-02.dll`是由 PID 为 1128 的`svchost.exe`进程加载的。



在procmon工具中，即可以使用在Process Explorer 发现的 PID：1128 进行过滤，收集这个恶意代码的信息。

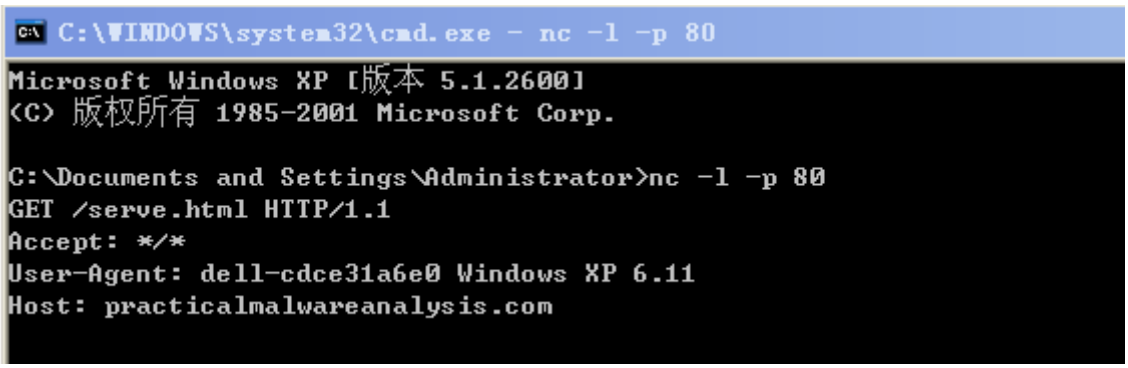
网络分析工具

检查ApateDNS，查看到恶意代码执行了DNS请求



输出结果显示了一个向`practicalmalwareanalysis.com`的DNS请求，这也匹配到了静态分析得到的字符串列表。

最后分析Netcat的结果（在先前字符串列表中可以看到有HTTP，所以使用Netcat监听80端口）



可见恶意代码执行了一个通过80端口的HTTP `GET`请求。多次运行这个测试，数据显示结果对每次运行都是一致的。

从这个数据可以创建出一组网络特征码。恶意代码始终发出一个对`serve.html`的GET请求，可以使用这个GET请求作为网络特征。

恶意代码也使用User-Agent为`dell-cdce31a6e0 windows XP6.11`.`dell-cdce31a6e0`是恶意代码分析虚拟机的名称，第二部分的User-Agent(windowsXP6.11)可以用来作为网络特征。

Lab03-03.exe

问题：

在一个安全的环境中执行Lab03-03.exe文件中发现的恶意代码，同时使用基础的动态行为分析工具监视它的行为。

1. 当你使用 Process Explorer 工具进行监视时，你注意到了什么？

   恶意代码文件运行时创建了一个子进程，接着结束掉自己的进程，留下了之前创建的子进程独立运行，执行了对svchost.exe 文件的替换。

2. 你可以找出任何的内存修改行为吗？

对比内存映像与磁盘映像中的svchost.exe，显示它们并不是一样的。内存映像拥有如 `practicalmalwareanalysis.log` 和 `[ENTER]` 这样的字符串，而磁盘镜像中却没有。

3. 这个恶意代码在主机上的感染迹象特征是什么？

   这个恶意代码创建了一个 `practicalmalwareanalysis.log` 日志文件

4. 这个恶意代码的目的是什么？

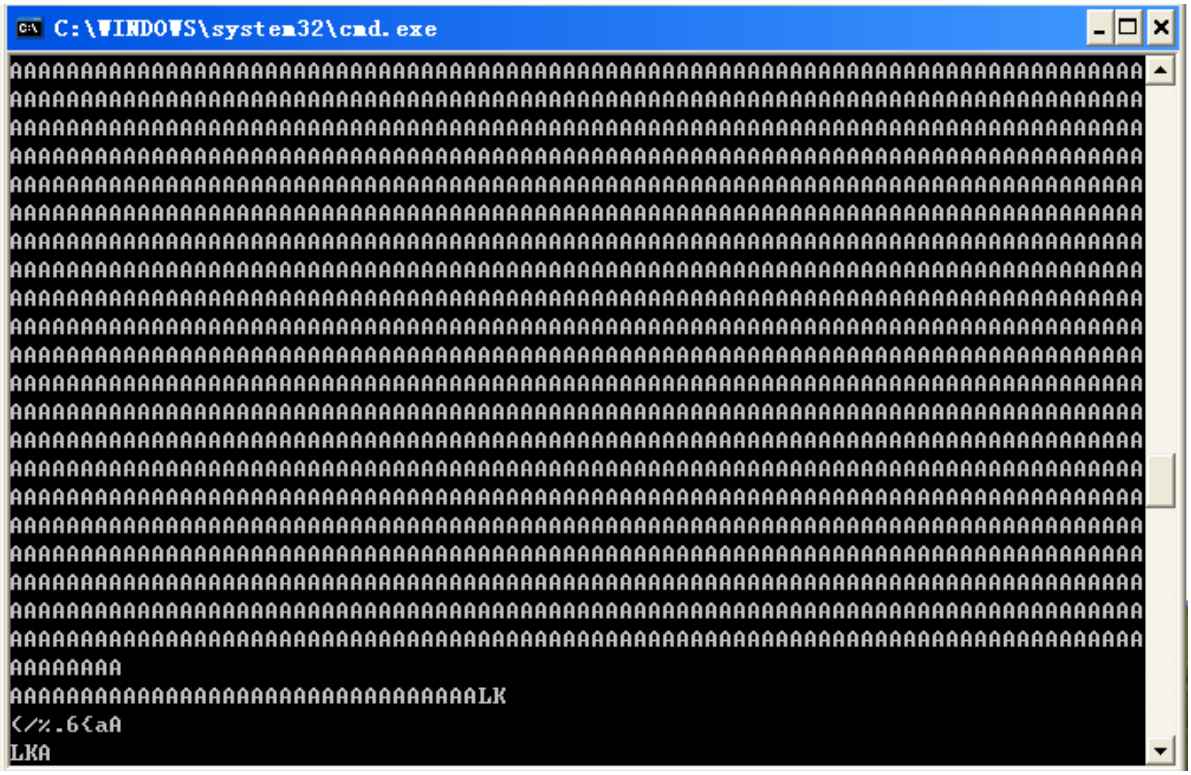
   这个程序在 `svchost.exe` 进程上执行了进程替换，来启动一个击键记录器，将键盘键入的信息记录在了 `practicalmalwareanalysis.log` 文件中。

具体解析过程如下：

- **静态分析**

使用PEiD工具查看该恶意代码，可以看到该恶意代码没有加壳迹象。

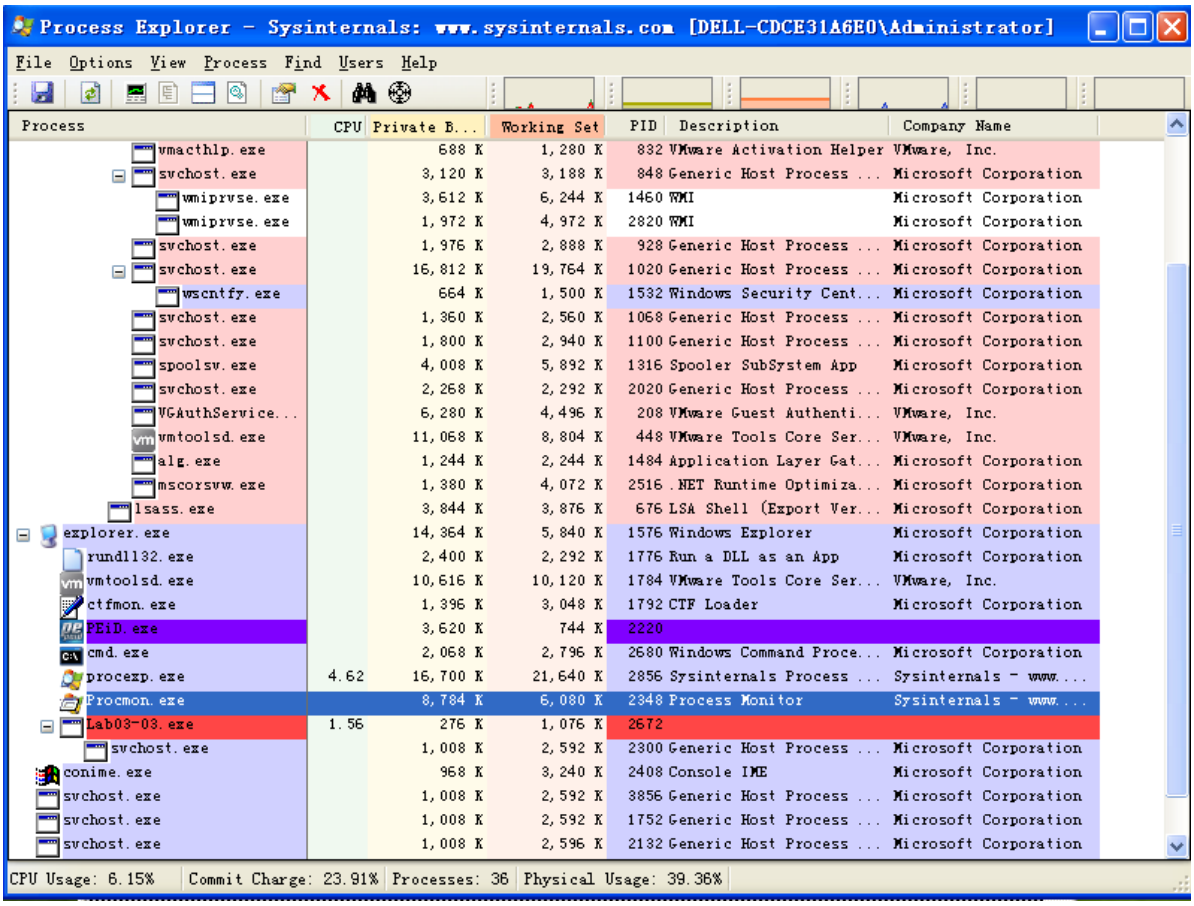
使用strings工具查看该恶意代码的字符串列表



可以发现，该文件中含有大量的字母A，这可以作为yara检测的一个规则。

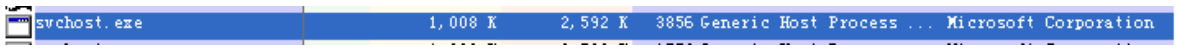
- **动态分析**

启动Process Explorer和Process Monitor监测工具进行动态检测。运行`Lab03-03.exe`。

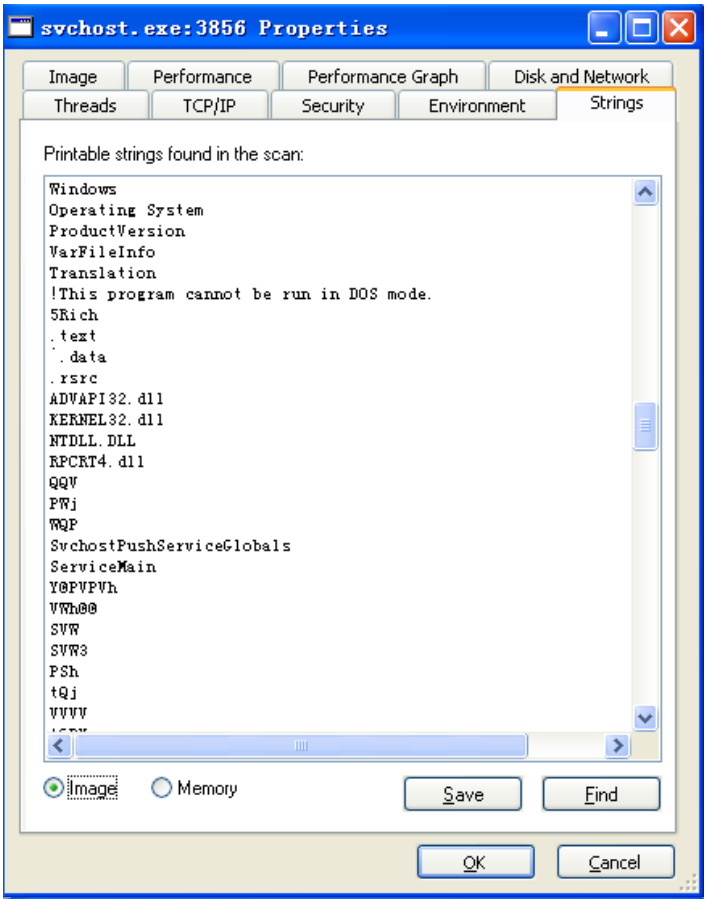


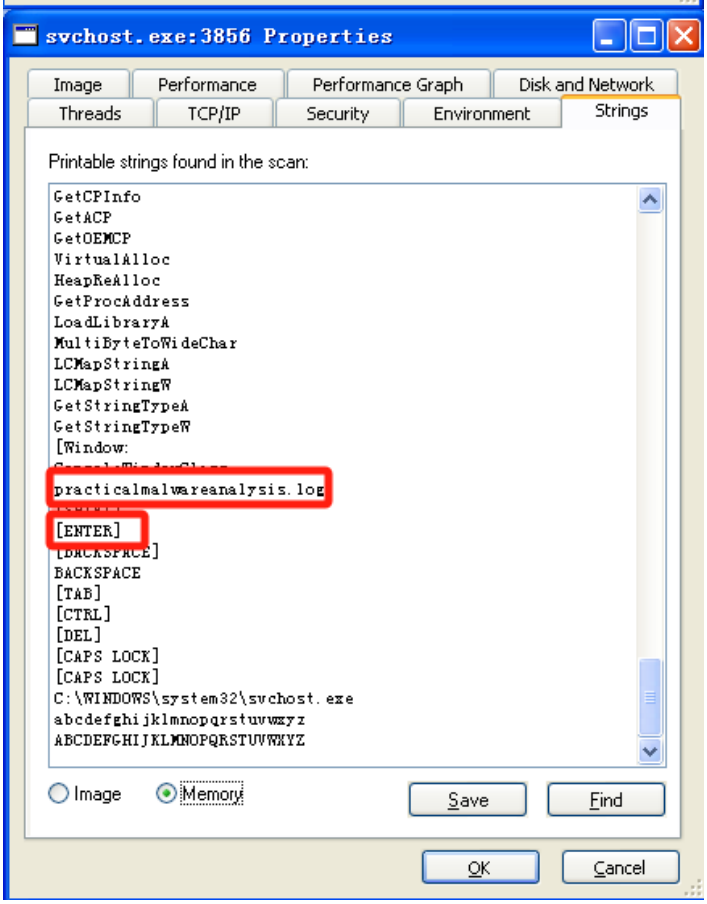
观察Process Explorer，可以看到，该文件除创建自己的进程外，还创建了一个子进程`svchost.exe`，创建后`Lab03-03.exe`自行退出，仅剩子进程`svchost.exe`独立运行。

查看`svchost.exe`进程的属性



这个进程看起来像是一个PID为3856的合法`svchost.exe`进程，但这个`svchost.exe`是很可疑的，因为`svchost.exe`通常是`services.exe`的子进程。



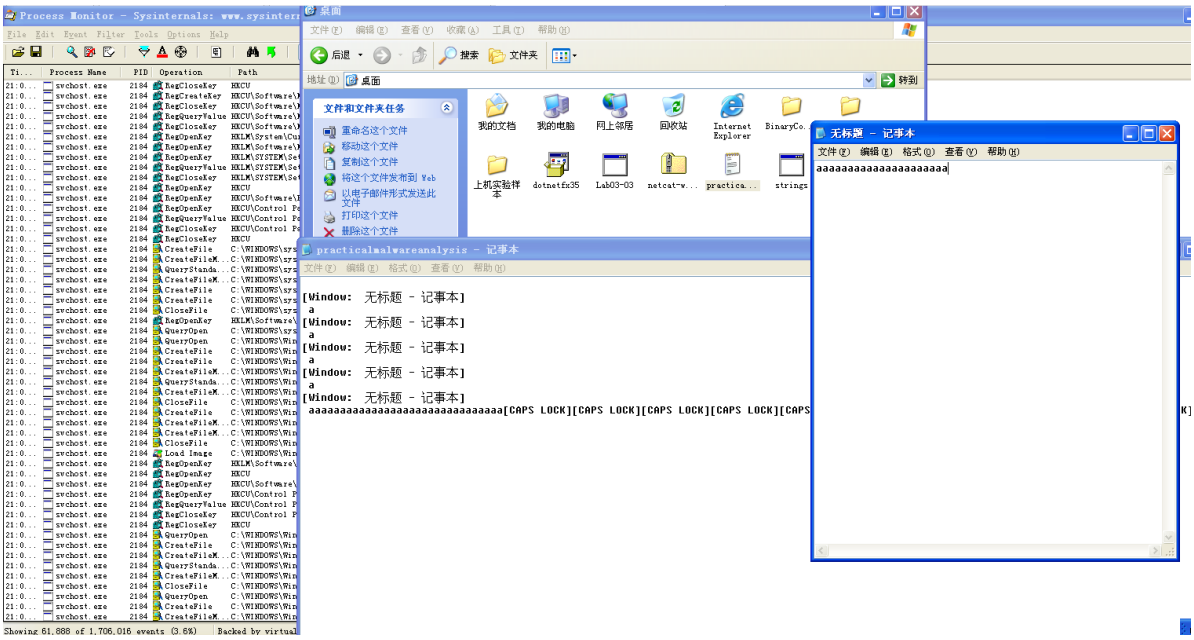


从这个相同的属性页面中，选择Strings查看在磁盘镜像中和内存镜像中可执行文件的字符串列表。通过在Image和Memory单选按钮之间切换，可以看出两者镜像中的显著差异。

右侧图中内存镜像中的字符串列表里包含了`practicalmalwareanalysis.log`和`[ENTER]`，而它们都不会在磁盘镜像中一个典型的svchos.exe文件中出现，如左侧图中所示。
`practicalmalwareanalysis.log`字符串的存在，再加上出现了`[ENTER]`和`[CAPS LOCK]`这样的字符串，表明这个程序很可能是一个击键记录器。

打开记事本程序，然后键入一段简短信息，来看看恶意代码是否在进行击键记录。

先在procmon工具中创建一个过滤器，只显示从svchost.exe进程的事件。`svchost.exe`的`CreateFile`和`WriteFile`事件正在写一个名为`practicalmalwareanalysis.log`的文件。



同时也可以看到文件创建的路径，按照该路径去打开，可以看到`practicalmalwareanalysis.log`文件。打开该日志文件，可以看到刚才写在记事本中的内容，这也与先前在内存镜像中看到的字符串一致。

综上可知这个恶意代码是一个击键记录器，使用svchost.exe进程替换技术。

Lab03-04.exe

问题：

使用基础的动态行为分析工具来分析在Lab03-04.exe文件中发现的恶意代码。

1. 当你运行这个文件时，会发生什么呢？

   当运行这个恶意代码时，程序会立即将自身删除掉，根据ProcessExplorer的进程变化，只看到了一个cmd进程，后迅速消失。

2. 是什么原因造成动态分析无法有效实施？

   根据上题中看到的cmd进程，怀疑该恶意代码可能需要提供一个命令行参数，或者这个程序的某个部件缺失了。

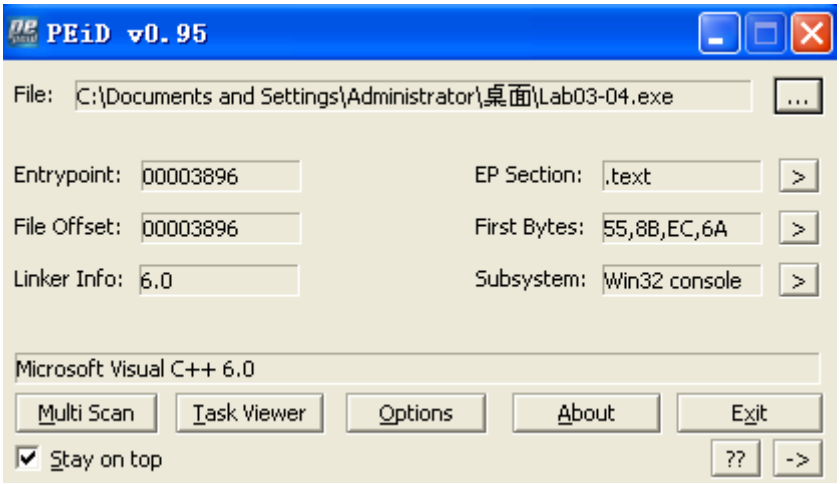
3. 是否有其他方式来运行这个程序？

   在尝试使用命令行运行恶意代码时，使用在字符串列表中发现的一些命令行参数(-in、-re、-cc)，但都无法成功运行，程序还是会删除自身。需要更深入的分析。

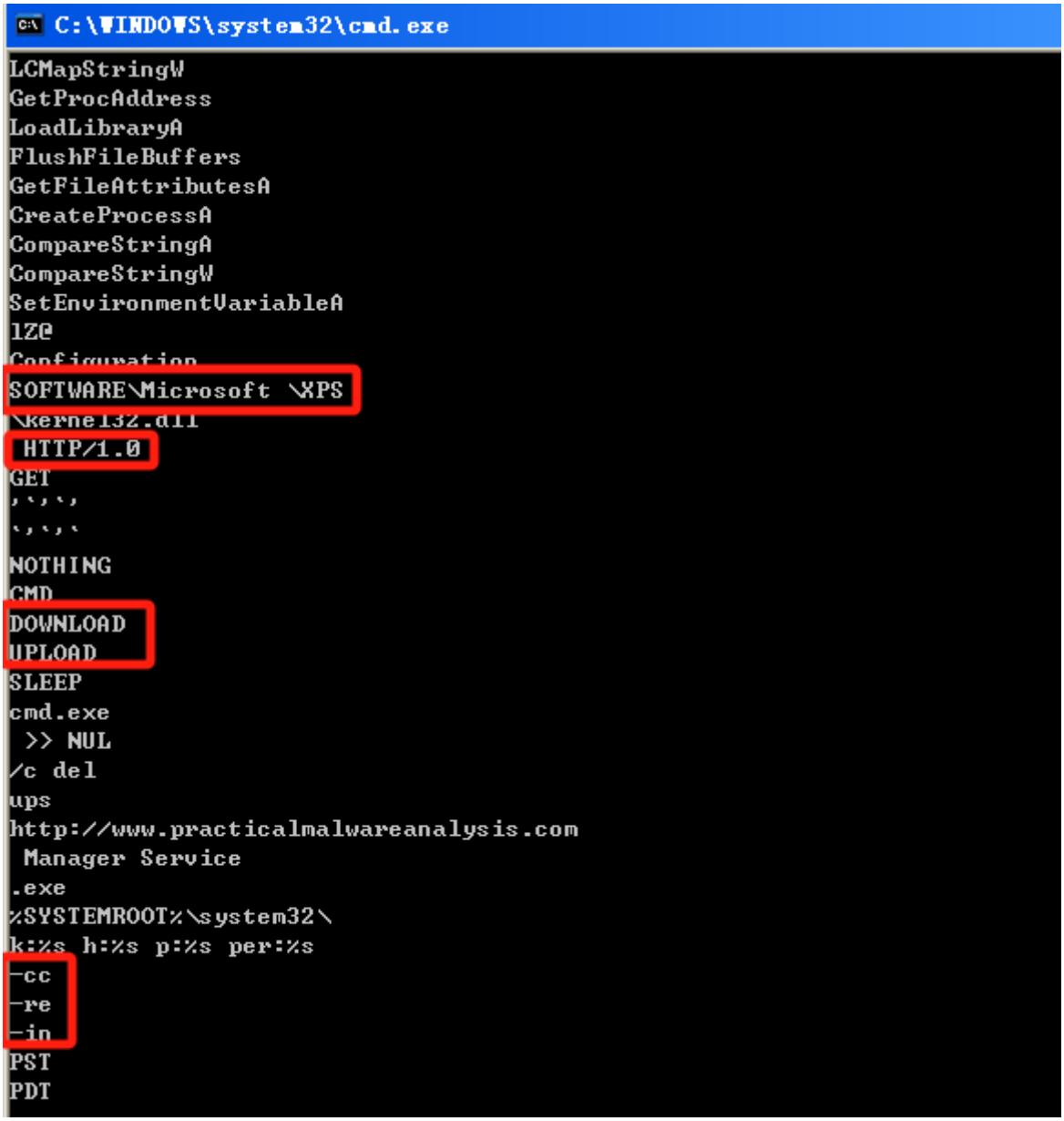
具体解析过程如下：

- **静态分析**

使用PEiD工具打开恶意代码，查看到该文件未加壳。



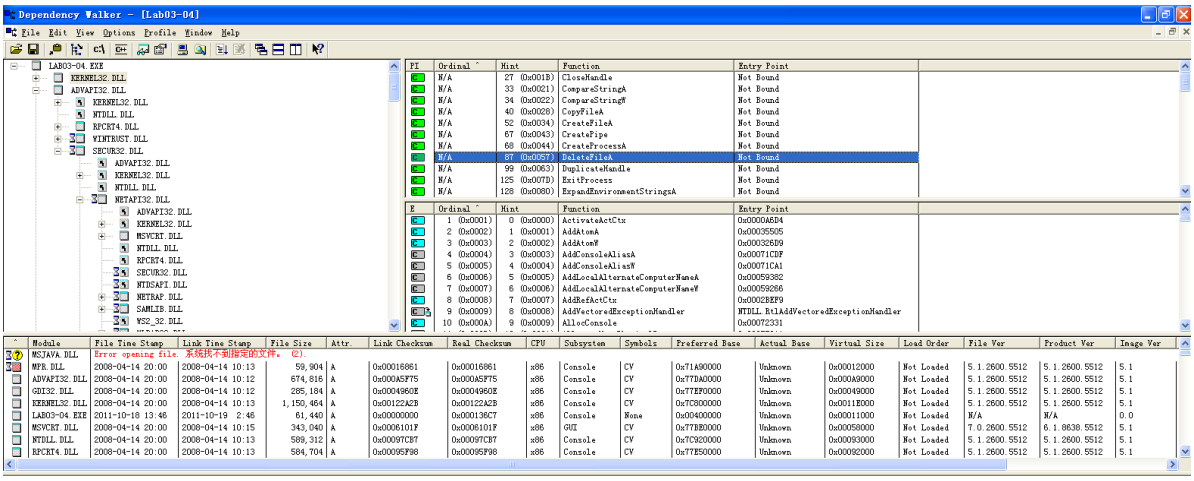
使用strings工具查看该恶意代码的字符串列表



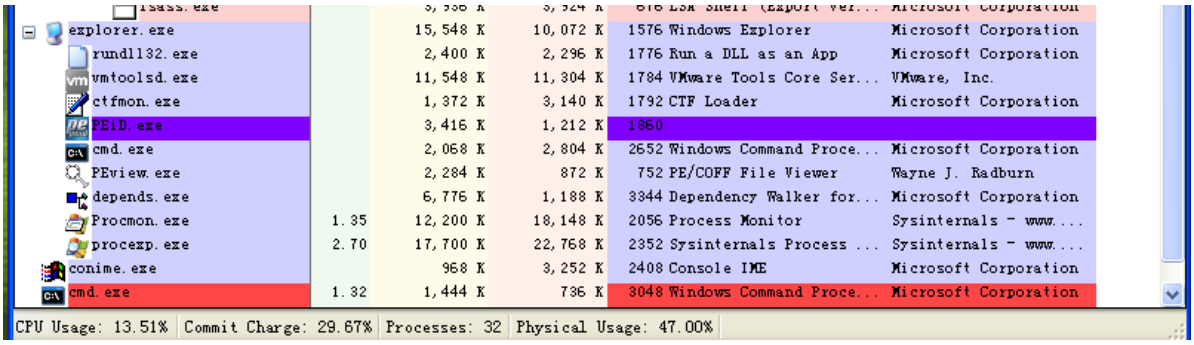
可见域名、注册表位置(如字符串`SOFTWARE\Microsoft \XPS`像`DOWNLOAD`、`UPLOAD`这样的命令字符串，以及`HTTP/1.0`字符串等。这些表明恶意代码可能是一个HTTP后门程序。字符串`-cc`、`-re`、`-in`应该是一些命令行参数(例如`-in`可能是install的缩写)。

- **动态分析**

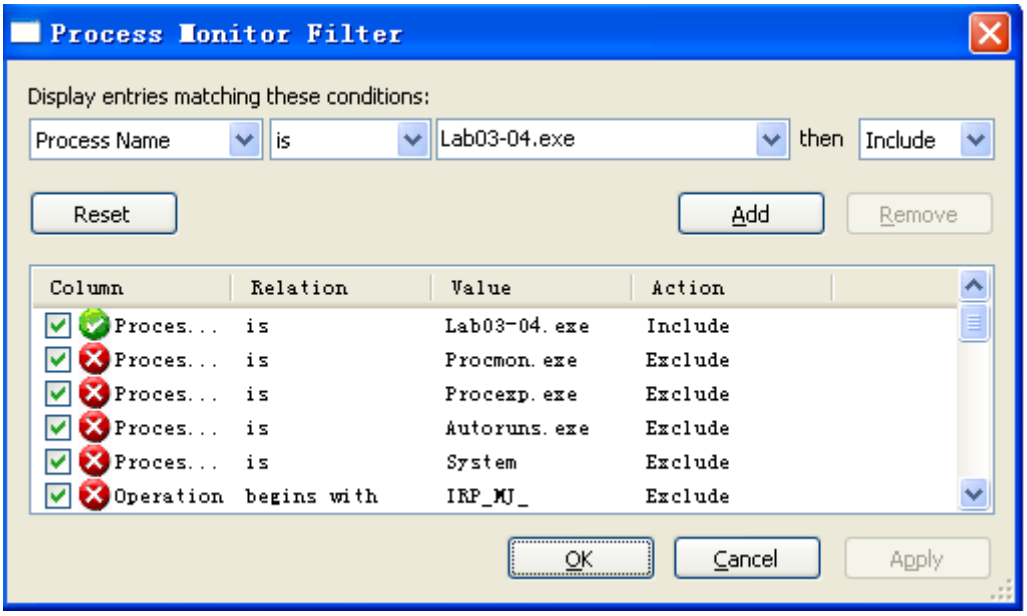
使用DependencyWalker打开该文件看到这个恶意代码导入了一些联网功能函数、服务操作函数和注册表操作函数，如DeleteFile和ShellExcute等可疑函数。



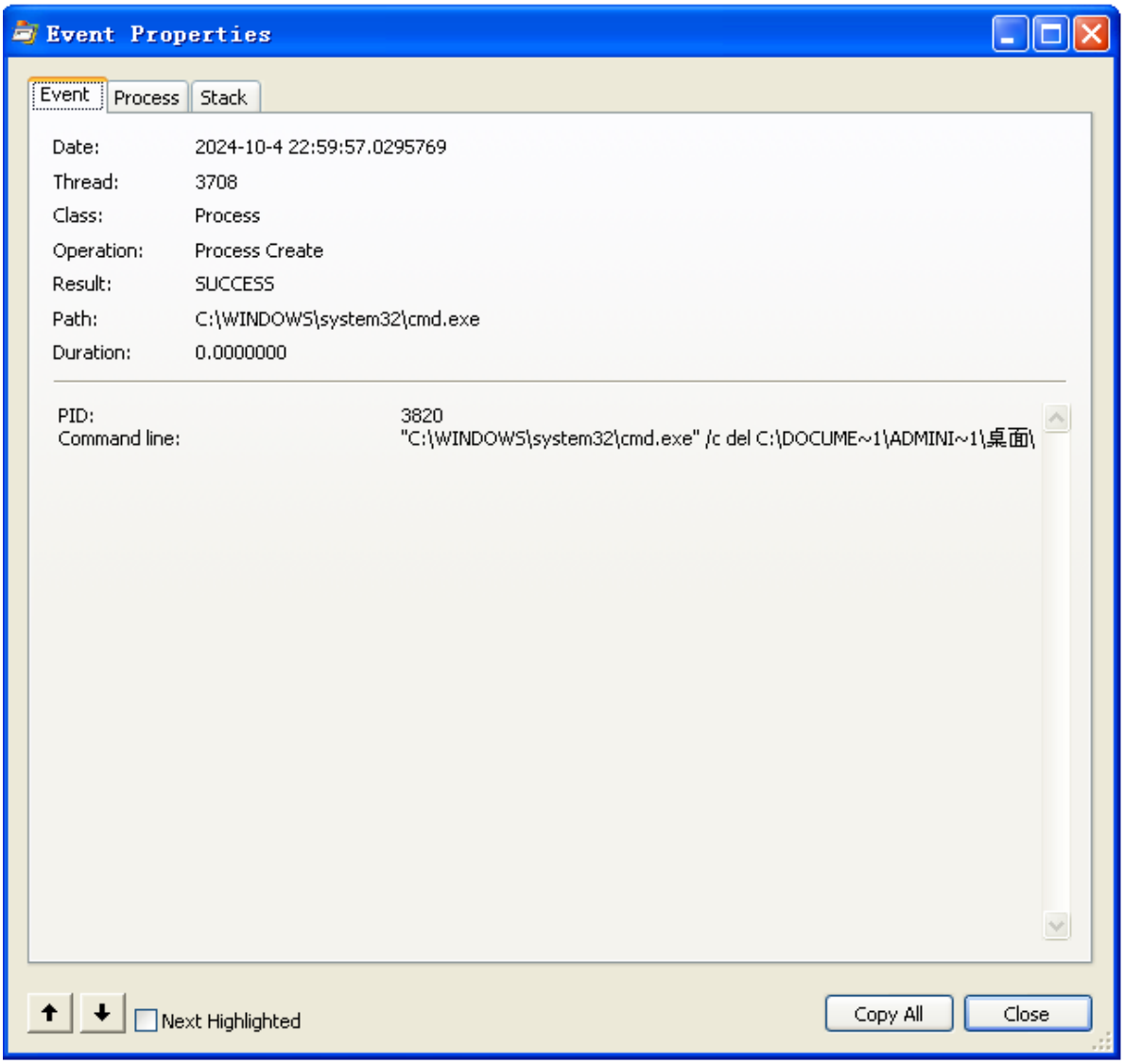
运行恶意代码之前，运行procmon工具并清除所有事件，启动ProcessExplorer，并设置一个虚拟网络。当运行`Lab03-04.exe`，只出现一个cmd进程，可以看到几秒后，文件消失了，打开回收站可以看到刚才打开的文件，该恶意代码似乎立即删除了自身，在Process Explorer中没有观察到特殊行为。



在Process Monitor中添加过滤规则`Process Name is Lab03-04.exe`，可以看到与该文件有关的操作。经过分析可以看到，恶意代码执行了与文件或是注册表相关的操作。



没有获得如WriteFile、RegSetValue之类的事件，但是通过深入挖掘后，发现了一个ProcessCreate的条目。查看该条目属性可以看到，恶意代码是通过使用`"C:\WINDOWS\system32\cmd.exe"/c del Z:\Lab03-04.exe>>NUL`来从系统中删去自身的。



四、Yara规则

基于上述分析，借助yargen工具，编写得到如下yara规则：

```
rule Lab03_01 {
   meta:
      description = "Lab03-01.exe"
   strings:
      $s1 = "vmx32to64.exe" fullword ascii
      $s2 = "SOFTWARE\\Classes\\http\\shell\\open\\commandV" fullword ascii
      $s3 = " www.practicalmalwareanalysis.com" fullword ascii
      $s4 = "CONNECT %s:%i HTTP/1.0" fullword ascii
      $s5 = "advpack" fullword ascii
      $s6 = "VideoDriver" fullword ascii
      $s7 = "AppData" fullword ascii /* Goodware String - occured 74 times */
      $s8 = "6I*h<8" fullword ascii /* Goodware String - occured 1 times */
      $s9 = "StubPath" fullword ascii /* Goodware String - occured 1 times */
      $s10 = "WinVMX32-" fullword ascii
      $s11 = "Software\\Microsoft\\Active Setup\\Installed Components\\"
fullword ascii /* Goodware String - occured 4 times */
      $s12 = "^-m-m<|<|<|M" fullword ascii
   condition:
      uint16(0) == 0x5a4d and filesize < 20KB and 8 of them
```

```
}

rule Lab03_02 {
   meta:
      description = "Lab03-02.dll"
   strings:
      $x1 = "%SystemRoot%\\System32\\svchost.exe -k " fullword ascii
      $x2 = "cmd.exe /c " fullword ascii
      $s3 = "RegOpenKeyEx(%s) KEY_QUERY_VALUE error ." fullword ascii
      $s4 = "Lab03-02.dll" fullword ascii
      $s5 = "practicalmalwareanalysis.com" fullword ascii
      $s6 = "RegOpenKeyEx(%s) KEY_QUERY_VALUE success." fullword ascii
      $s7 = "GetModuleFileName() get dll path" fullword ascii
      $s8 = "dW5zdXBwb3JO" fullword ascii /* base64 encoded string 'unsupport'
*/
      $s9 = "Y29ubmVjdA==" fullword ascii /* base64 encoded string 'connect' */
      $s10 = "OpenService(%s) error 2" fullword ascii
      $s11 = "OpenService(%s) error 1" fullword ascii
      $s12 = "CreateService(%s) error %d" fullword ascii
      $s13 = "You specify service name not in Svchost//netsvcs, must be one of
following:" fullword ascii
      $s14 = "RegQueryValueEx(Svchost\\netsvcs)" fullword ascii
      $s15 = "netsvcs" fullword ascii
      $s16 = "serve.html" fullword ascii
      $s17 = "DependOnService" fullword ascii
      $s18 = ":$:2:K:U:\\:l:" fullword ascii
      $s19 = "uninstall is starting" fullword ascii
      $s20 = "uninstall success" fullword ascii
   condition:
      uint16(0) == 0x5a4d and filesize < 70KB and 1 of ($x*) and 4 of them
}

rule Lab03_03 {
   meta:
      description = "Lab03-03.exe"
   strings:
      $s1 = "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA" ascii
      $s2 =
"AAAAAAAAAAAAAAAAAAAA
```

```
        $s12 =
"AAAA
```

1. 实验过程中遇到的部分问题

（1）windowsXP中的apateDNS不能正常运行，会显示应用程序正常初始化失败

需要另外下载`.NET framework`，在更高级的windows操作系统中是可以自动查询缺失文件进行加载，而在WindowsXP中需要自主下载安装包，同时还需注意下载版本兼容问题。

（2）在Win10上进行恶意代码分析时，发现其有些分析结果与xp操作系统上不同，包括一些字符串列表输出以及网络感染的特征都会有些不同。这主要是由于两个操作系统在架构设计、安全机制、更新策略以及用户交互方式等多个方面存在显著区别。Win10采用了更为现代化和安全的架构设计，包括增强的内核保护、更严格的权限管理和更完善的内存管理机制，这些改进使得Win10在抵御恶意代码攻击方面表现出更强的能力。初步接触恶意代码分析之后在进行恶意代码分析时还是使用WindowsXP操作系统，可以观察到完整稳定的恶意代码执行过程和执行目的。

(3)`WinVMX32`**的互斥量**：

互斥量（Mutex）是一种同步机制，用于防止多个线程或进程同时访问共享资源。恶意软件创建互斥量通常是为了确保只有一个实例在运行，避免重复执行或资源冲突。`WinVMX32`可能是这个恶意软件用来标识自己或确保唯一性的名称。

(4)**Netcat监听端口**

80端口和443端口是网络通信中常用的两个端口，它们之间有几个关键的区别：

80端口：

- **协议**：用于HTTP（HyperText Transport Protocol，超文本传输协议）通信。
- **用途**：主要用于传输未加密的网页数据。当你在浏览器中输入一个网址并访问网站时（如果网址以“http://”开头），通常是通过80端口进行通信的。

443端口：

- **协议**：用于HTTPS（HyperText Transport Protocol Secure，安全超文本传输协议）通信。HTTPS是在HTTP基础上加入了SSL/TLS加密层，用于确保数据在客户端和服务器之间的传输是安全的。
- **用途**：主要用于传输加密的网页数据，以保护用户隐私和数据安全。几乎所有使用HTTPS协议的网站（网址以“https://”开头）都是通过443端口进行通信的。

2. 心得体会

在本次实验中，更完整的了解了ProcessExplorer、ProcessMonitor等分析工具的使用方法，初步掌握了动态分析恶意代码的方法。

可以看到不同的恶意代码的隐藏技术的多样性以及在主机上的不同感染迹象特征的差异。