

南开大学

恶意代码分析与防治技术课程实验报告

实验三



学院：网络空间安全学院

专业：信息安全

学号：2113997

姓名：齐明杰

班级：信安2班

1 实验目的

- 完成教材Lab3的实验内容，编写Lab3样本的Yara引擎规则，并测试规则的执行效率
- 提高编写Yara规则的能力和熟练度
- 学习使用各项工具来进行病毒识别和分析

2 实验原理

- **动态分析：**恶意代码的动态分析是一种分析恶意软件（如病毒、恶意软件、木马等）行为和功能的方法，通过运行它们并监视其在系统内部的行为来进行分析。与静态分析不同，静态分析是通过分析恶意代码的源代码或二进制文件而不运行它们来进行的。

动态分析的关键特点和步骤：

1. 运行恶意代码：分析人员将恶意代码运行在一个受控环境中，通常是一个虚拟机或沙盒环境，以防止它对真实系统造成损害。
2. 监视行为：在运行期间，恶意代码的行为被监视和记录下来。这包括文件操作、网络通信、系统调用、注册表访问、进程创建等。
3. 收集数据：分析人员会收集有关恶意代码行为的大量数据，以便后续分析。这可能包括系统调用跟踪、网络流量捕获、内存转储等。
4. 行为分析：通过分析监视到的数据，分析人员可以识别恶意代码的行为模式。这有助于理解恶意代码的目的和功能，例如窃取敏感信息、传播自身或其他恶意活动。
5. 环境模拟：有时，分析人员可能会尝试模拟受感染系统的特定环境，以更好地理解恶意代码的行为如何受到不同条件的影响。
6. 恶意代码交互：在某些情况下，分析人员可能会尝试与恶意代码进行交互，以深入了解其功能。但这需要极高的谨慎，以避免恶意代码对实际系统造成损害。

3 实验过程

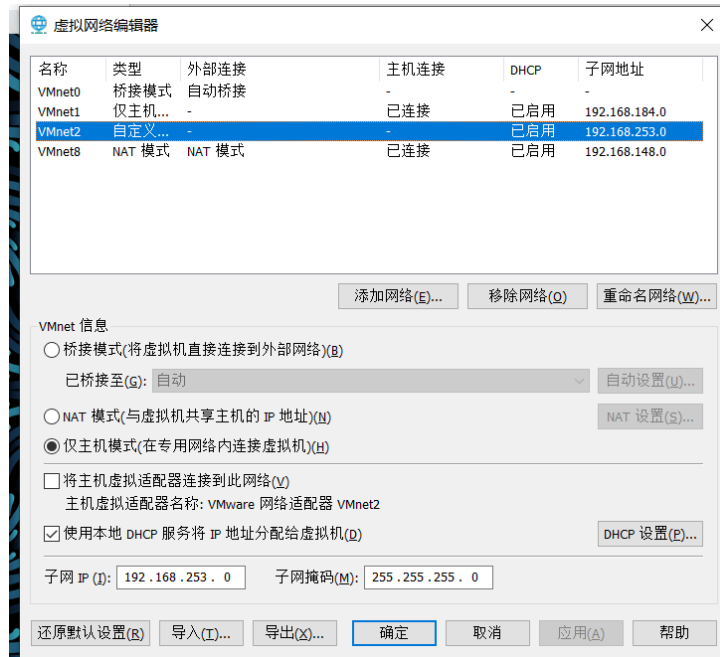
3.1 配置虚拟网络环境

在本次实验我将使用kali linux虚拟机和win xp虚拟机进行实验。

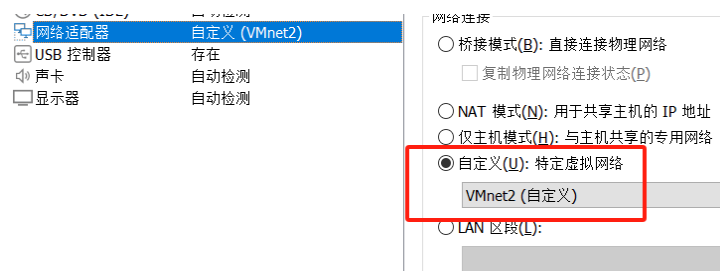
在本次实验中，我们需要动态分析恶意代码，但某些恶意代码需要网络环境，如果我们单纯地在什么网络环境都没有的虚拟机内分析恶意代码，将一无所获，达不到分析的目的。

因此，为了得到恶意代码的网络行为，我们需要使用linux下的InetSim软件，这个软件是一款免费的模拟网络环境的软件，我们在kali linux虚拟机内使用它，作为虚拟网络的服务器。

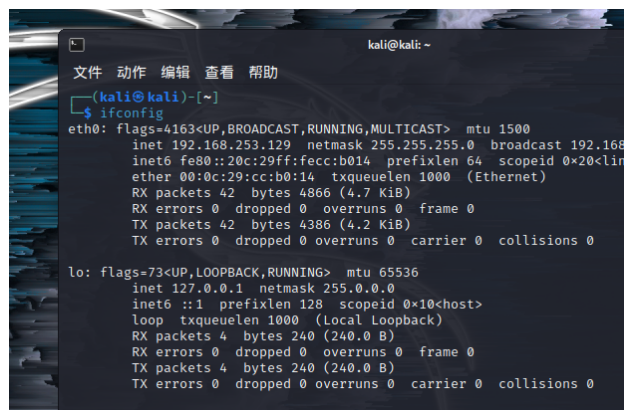
打开vmware的虚拟网络编辑器，添加一个不连接主机网络的net:



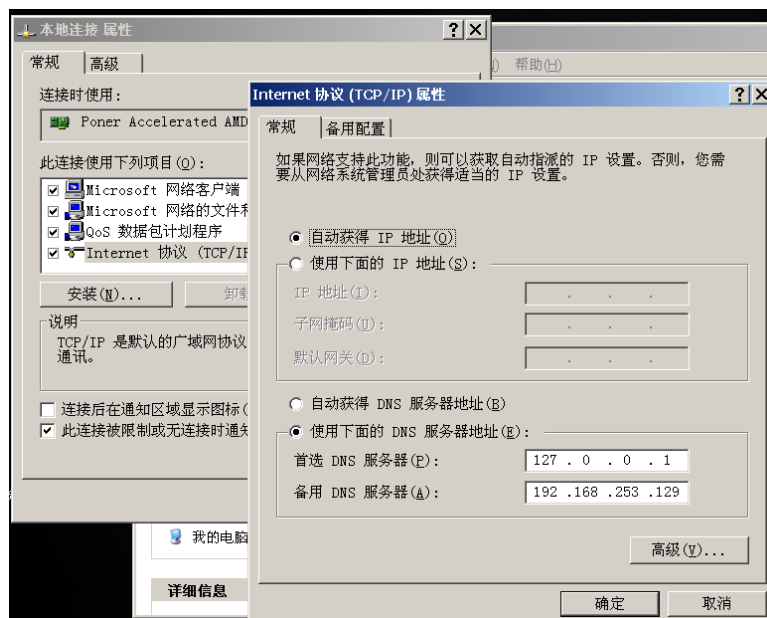
将kali linux和xp虚拟机的网络配置为此新建网络:



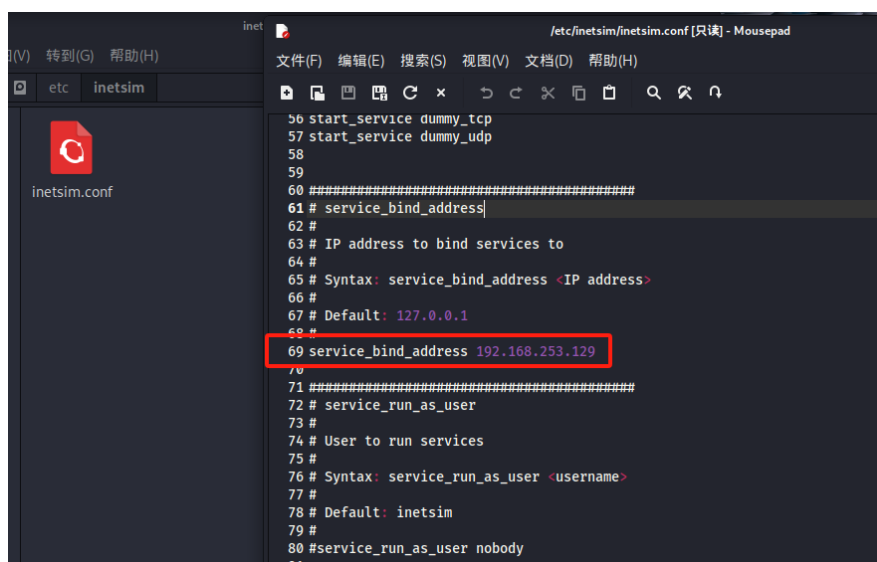
获取kali虚拟机的ip地址, 为 192.168.253.129:



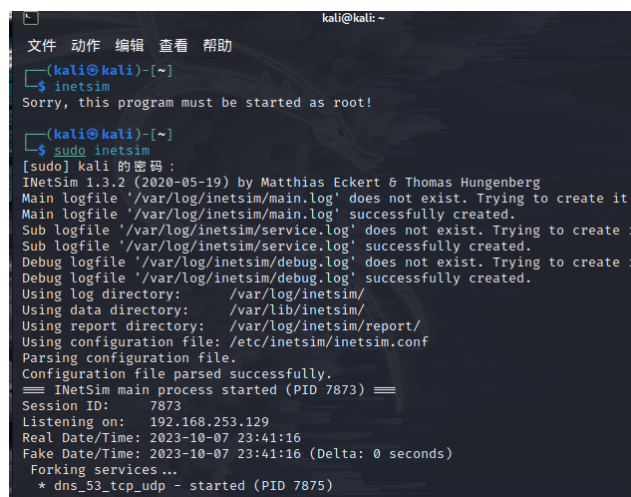
然后更改xp虚拟机的DNS设置:



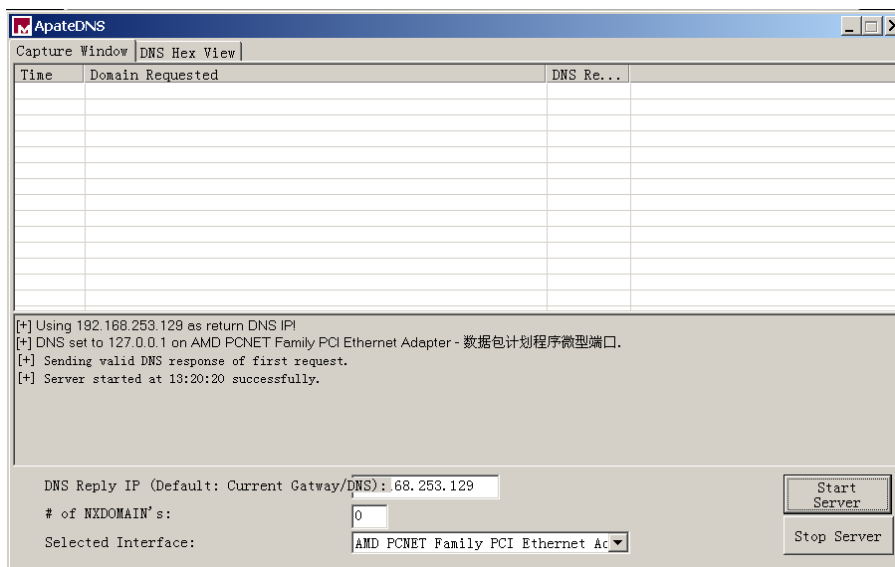
配置kali的 `/etc/inetsim/inetsim.conf` 文件:



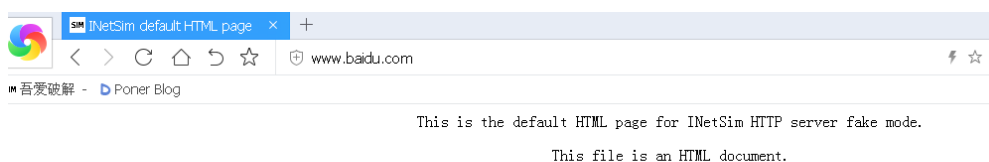
然后在kali启动 `inetsim` :



在xp虚拟机打开 `ApateDNS` 工具, 输入kali的ip地址后 `Start Server` :



随便进入一个域名(如www.baidu.com), 可以看到变成了 [INetsim](#) 产生的虚假网页:



这样, 虚拟网络环境就准备就绪了。

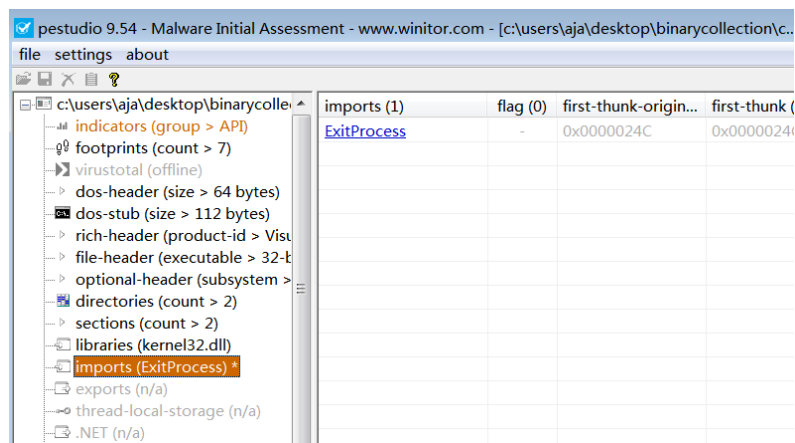
3.2 Lab03-01.exe

经过 [PEiD](#) 查壳, 发现这个exe经过 [PEncrypt](#) 加壳处理:



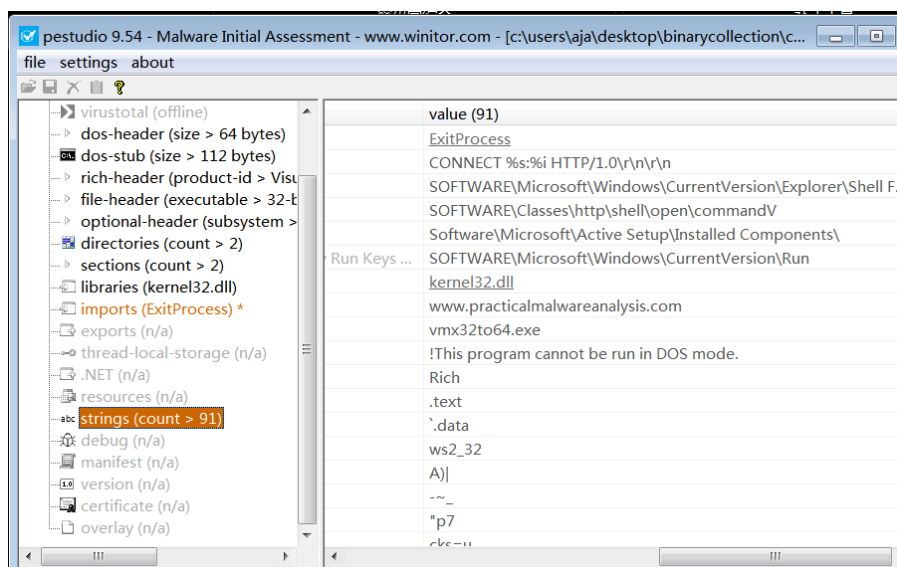
- Q1: 找出这个恶意代码的导入函数与字符串列表?

使用 [pestudio](#) 查看导入函数:



发现只有一个函数 `Kernel32.ExitProcess`。

同样查看字符串列表如下所示：



其中发现了一个有趣的字符串 `vmx32to64.exe`，似乎指向一个可执行文件。

- Q2: 这个恶意代码在主机上的感染迹象特征是什么？

由于该恶意代码经过加壳处理，静态分析难以获得有效的结果，因此我们进行动态分析。

首先为虚拟机保存快照，然后再继续操作。

我们打开 `ProcessExplorer` 和 `ProcessMonitor`（后面分别简称为 *procexp* 和 *procmon*），以及 `ApateDNS`，执行 `Lab03-01.exe`。

在 *procexp* 可以看到该恶意代码的运行时信息，如句柄，DLL等：

Handle:

Process Explorer - Sysinternals: www.sysinternals.com [WINXP-S2POJIE-2\Administrator]

File Options View Process Find Handle Users Help

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
System Idle Process	95.71	K	28 K	0		
System	0.36	K	292 K	4		
explorer.exe	0.36	14,788 K	8,584 K	1568	Windows Explorer	Microsoft Corporat
vmtoolsd.exe		9,748 K	14,708 K	1724	VMware Tools Core Service	VMware, Inc.
ctfmon.exe		1,048 K	3,360 K	1732	CTF Loader	Microsoft Corporat
Poner.exe		20,424 K	964 K	1828	Poner	Poner
apatedNS.exe		21,184 K	6,612 K	1356	Mandiant	Mandiant
procexp.exe	1.79	11,600 K	17,972 K	404	Sysinternals Process Ex...	Sysinternals - ww
Procmon.exe	0.36	8,464 K	11,108 K	1236	Process Monitor	Sysinternals - ww
Lab03-01.exe		864 K	2,312 K	580		
conime.exe		1,008 K	2,956 K	2004	Console IME	Microsoft Corporat

Type	Name
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
KeyedEvent	\KernelObjects\CritSecOutOfMemoryEvent
Mutant	\BaseNamedObjects\WinVMX32
Port	
Semaphore	
Semaphore	
Semaphore	
Semaphore	
Thread	Lab03-01.exe (580): 608
Thread	Lab03-01.exe (580): 608
WindowStation	\Windows\WindowStations\WinSta0
WindowStation	\Windows\WindowStations\WinSta0

CPU Usage: 4.29% Commit Charge: 5.73% Processes: 25 Physical Usage: 11.35%

DLL:

Process Explorer - Sysinternals: www.sysinternals.com [WINXP-S2POJIE-2\Administrator]

File Options View Process Find DLL Users Help

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
vmtoolsd.exe		11,348 K	16,304 K	1724	VMware Tools Core Service	VMware, Inc.
ctfmon.exe		1,048 K	3,360 K	1732	CTF Loader	Microsoft Corporation
Poner.exe		20,424 K	964 K	1828	Poner	Poner
apatedNS.exe		21,184 K	6,612 K	1356	Mandiant	Mandiant
procexp.exe	6.56	11,600 K	17,972 K	404	Sysinternals Process Ex...	Sysinternals - www.sy...
Procmon.exe	0.31	8,712 K	11,132 K	1236	Process Monitor	Sysinternals - www.sy...
Lab03-01.exe		864 K	2,312 K	580		
conime.exe		1,008 K	2,956 K	2004	Console IME	Microsoft Corporation

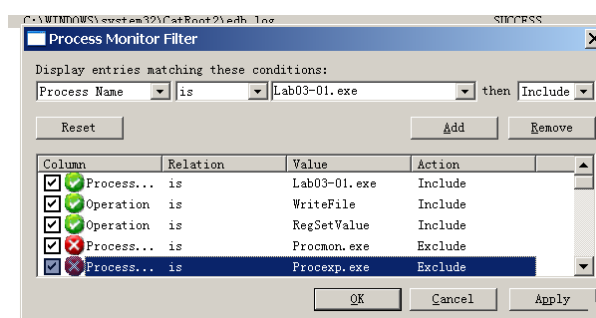
Name	Description	Company Name	Path
iphlpapi.dll	IP Helper API	Microsoft Corporation	C:\WINDOWS\system32\iphlpapi.dll
kernel32.dll	Windows NT BASE API Client DLL	Microsoft Corporation	C:\WINDOWS\system32\kernel32.dll
Lab03-01.exe			C:\Documents and Settings\Administrator\桌面\Binar...
locale.nls			C:\WINDOWS\system32\locale.nls
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 2...	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
rport4.dll	Remote Procedure Call Runtime	Microsoft Corporation	C:\WINDOWS\system32\rport4.dll
secur32.dll	Security Support Provider In...	Microsoft Corporation	C:\WINDOWS\system32\secur32.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 DLL	Microsoft Corporation	C:\WINDOWS\system32\user32.dll
usp10.dll	Uniscribe Unicode script pro...	Microsoft Corporation	C:\WINDOWS\system32\usp10.dll
version.dll	Version Checking and File In...	Microsoft Corporation	C:\WINDOWS\system32\version.dll
winmr.dll	LDAP RnR Provider DLL	Microsoft Corporation	C:\WINDOWS\system32\winmr.dll
wldap32.dll	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
ws2help.dll	Windows Socket 2.0 Helper fo...	Microsoft Corporation	C:\WINDOWS\system32\ws2help.dll
wshtcpip.dll	Windows Sockets Helper DLL	Microsoft Corporation	C:\WINDOWS\system32\wshtcpip.dll

CPU Usage: 9.05% Commit Charge: 5.77% Processes: 25 Physical Usage: 11.43%

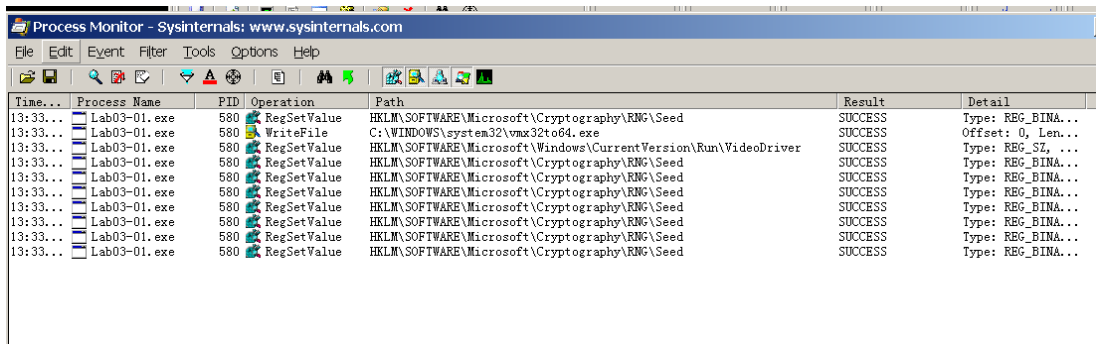
发现该文件创建了一个互斥量WinVMX32，并且使用了一些具有联网功能的DLL，如

ws2_32.dll 和 wshtcpip.dll。

另外为了寻找更多信息，我们使用procmon设置过滤器，观察该恶意代码对系统和注册表是否有修改：



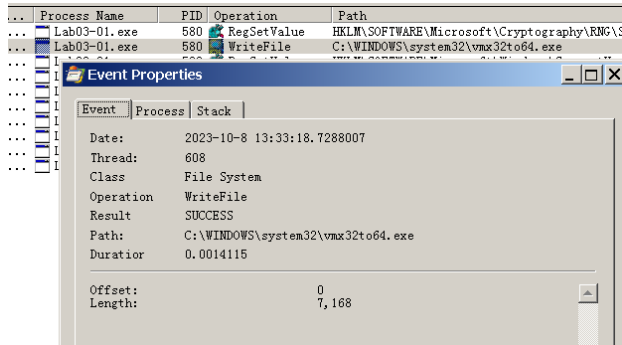
我们设置了三个条件，分别是进程名，写文件操作和写注册表操作，来定位恶意代码的行为。获得了十条结果：



Time...	Process Name	PID	Operation	Path	Result	Detail
13:33...	Lab03-01.exe	580	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS	Type: REG_BINARY...
13:33...	Lab03-01.exe	580	WriteFile	C:\WINDOWS\system32\vmx32to64.exe	SUCCESS	Offset: 0, Len...
13:33...	Lab03-01.exe	580	RegSetValue	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\VideoDriver	SUCCESS	Type: REG_SZ, ...
13:33...	Lab03-01.exe	580	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS	Type: REG_BINARY...
13:33...	Lab03-01.exe	580	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS	Type: REG_BINARY...
13:33...	Lab03-01.exe	580	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS	Type: REG_BINARY...
13:33...	Lab03-01.exe	580	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS	Type: REG_BINARY...
13:33...	Lab03-01.exe	580	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS	Type: REG_BINARY...
13:33...	Lab03-01.exe	580	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS	Type: REG_BINARY...
13:33...	Lab03-01.exe	580	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS	Type: REG_BINARY...

其中若干个以Seed为结尾的均为无效噪声，不用理会，重点观察第二、三条。

第二条是文件写入操作，双击查看详细信息后发现恶意代码向
C:\WINDOWS\system32\vmx32to64.exe写入了7168字节的二进制数据：



Process Name	PID	Operation	Path
Lab03-01.exe	580	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed
Lab03-01.exe	580	WriteFile	C:\WINDOWS\system32\vmx32to64.exe

Event	Process	Stack
Date:	2023-10-8 13:33:18.7288007	
Thread:	608	
Class	File System	
Operation	WriteFile	
Result	SUCCESS	
Path:	C:\WINDOWS\system32\vmx32to64.exe	
Duration	0.0014115	
Offset:	0	
Length:	7,168	

对比一下该恶意代码的大小：

Lab03-01.exe 属性

常规

兼容性

摘要

Lab03-01.exe

文件类型: 应用程序

描述: Lab03-01

位置: C:\Documents and Settings\Administrato

大小: 7.00 KB (7,168 字节)

占用空间: 8.00 KB (8,192 字节)

创建时间: 2023年9月17日 星期日, 17:50:44

修改时间: 2011年4月22日 星期五, 12:55:50

访问时间: 2023年9月17日 星期日, 17:50:44

属性:

☐

只读(R)

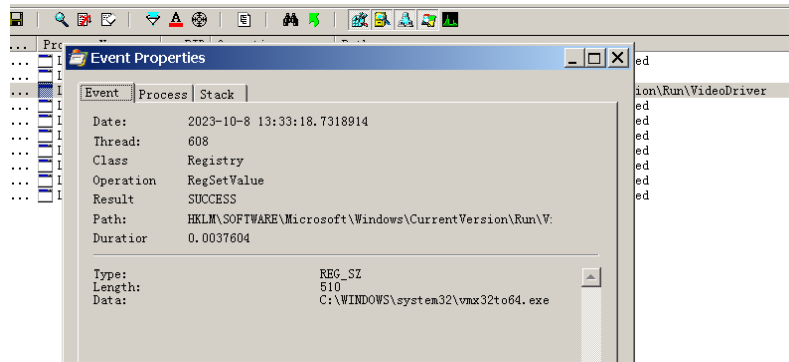
☐

隐藏(H)

高级(D)

发现正好相等，可以说明该恶意代码复制了自身到系统的上述位置，这可以作为主机被感染的迹象，因为该恶意代码写死了这个复制的文件名。

同样查看第三条记录，发现恶意代码向注册表内写入了信息：

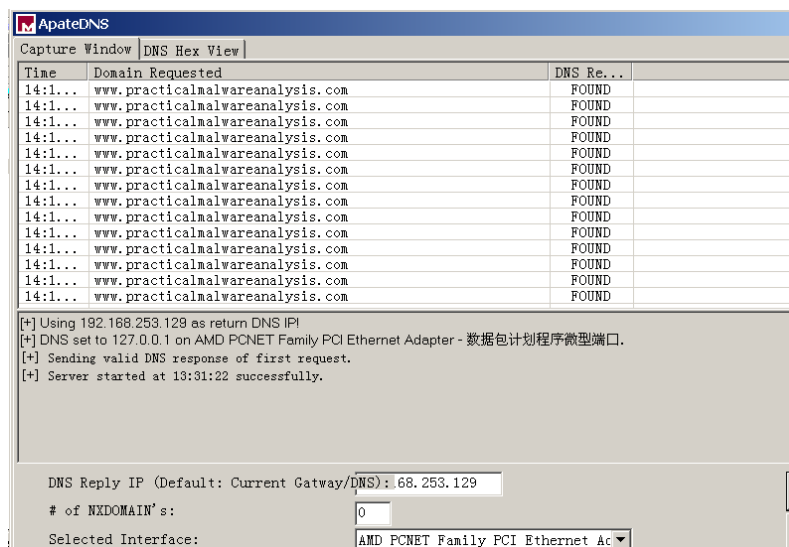


新建的注册表项位于 `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` ,名为 `VideoDriver` ,可以用于在系统启动时自动启动 `vmx32to64.exe` ,也就是Lab03-01.exe的副本。

据此可以得出结论：该恶意代码拷贝自身到 `C:\WINDOWS\system32\vmx32to64.exe` 内，然后修改注册表项 `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\VideoDriver` ,达到开机自启的效果，这便是主机感染迹象。

- Q3: 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?

打开刚才已经正在运行的ApateDNS, 查看结果：



发现该恶意代码不断对域名 `www.practicalmalwareanalysis.com` 发起了请求，与上述字符串列表相呼应。

同时使用netcat进行查看，看443和80端口是否有发出信息，发现在80端口无信息，在443端口有一堆乱码：

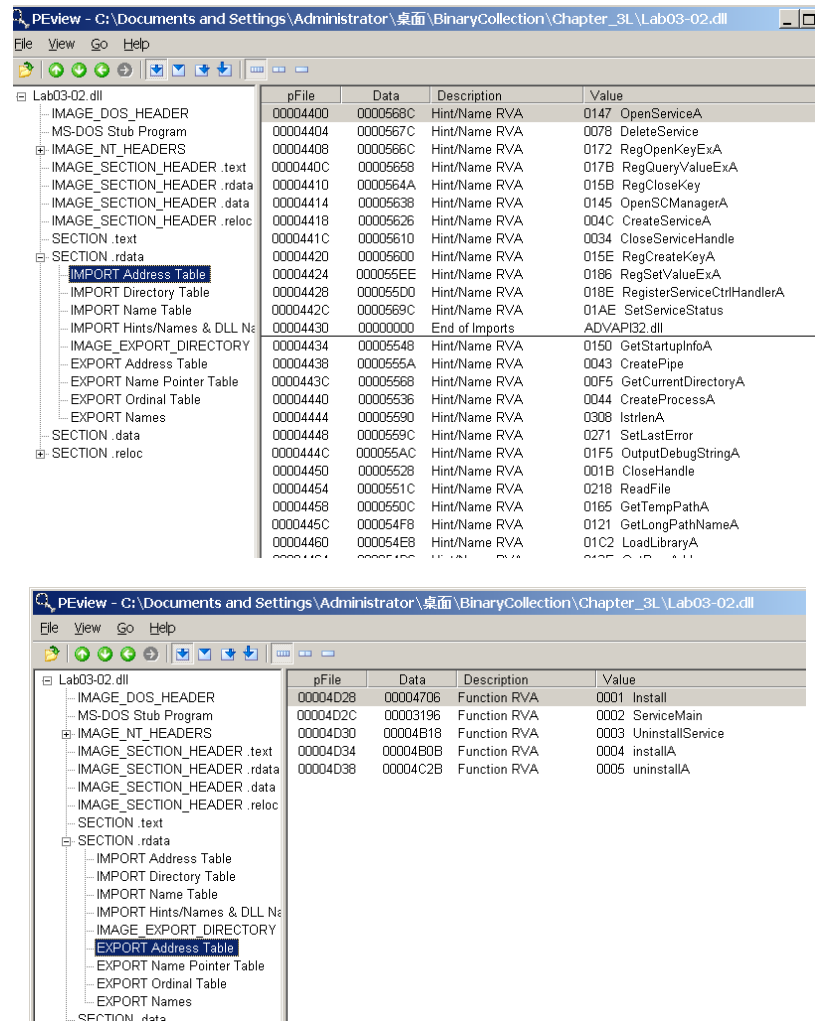


但经过多次测试，其中的数据流似乎是随机的，并没有什么有效的信息。

3.3 Lab03-02.dll

- Q1: 你怎样才能让这个恶意代码自行安装?

先进行静态分析，先使用PEview查看该恶意代码的导入，导出函数：



导入函数包含一些服务操作函数，如 `CreateServiceA` 等，同时发现导出函数包含 `ServiceMain` 等与服务相关的函数。

使用strings查看字符串：

```
C:\WINDOWS\system32\cmd.exe
ABCDEF GHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/-!>
<!--
.PAX
.PAD
DependOnService
RpcSs
ServiceDll
GetModuleFileName() get dll path
Parameters
Type
Start
ObjectName
LocalSystem
ErrorControl
DisplayName
Description
Depends INA+, Collects and stores network configuration and location information
, and notifies applications when this information changes.
ImagePath
%SystemRoot%\System32\svchost.exe -k
SYSTEM\CurrentControlSet\Services\
CreateService(%s) error %d
Intranet Network Awareness (INA+)
%SystemRoot%\System32\svchost.exe -k netsvcs
```

发现了一些有关注册表位置，域名等字符串。

因此需要为该恶意代码安装一个服务,首先使用 **Regshot** 保存注册表状态，然后使用以下命令：

```
1 | rundll32.exe Lab03-02.dll,installA
```

之后再次使用Regshot，将两次快照进行对比：

```
18 Values added:30
19 -----
20 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 01 00 00 00 02 00 60 00 04 00 00
- 00 00 14 00 FD 01 02 00 01 01 00 00 00 00 00 05 12 00 00 00 00 18 00 FF 01 0F 00 01 02 00 00 00 00 05 20 00
- 00 00 20 02 00 00 00 00 14 00 8D 01 02 00 01 01 00 00 00 00 00 05 0B 00 00 00 00 18 00 FD 01 02 00 01 02 00
- 00 00 00 05 20 00 00 00 23 02 00 00 01 01 00 00 00 00 00 05 12 00 00 00 01 01 00 00 00 00 05 12 00 00 00
21 HKLM\SYSTEM\ControlSet001\Services\IPRIP\Parameters\ServiceDll: "C:\Documents and Settings\Administrator\桌面\
- BinaryCollection\Chapter_3L\Lab03-02.dll"
22 HKLM\SYSTEM\ControlSet001\Services\IPRIP\Type: 0x00000020
23 HKLM\SYSTEM\ControlSet001\Services\IPRIP\Start: 0x00000002
24 HKLM\SYSTEM\ControlSet001\Services\IPRIP\ErrorControl: 0x00000001
25 HKLM\SYSTEM\ControlSet001\Services\IPRIP\ImagePath: "%SystemRoot%\System32\svchost.exe -k netsvcs"
26 HKLM\SYSTEM\ControlSet001\Services\IPRIP\DisplayName: "Intranet Network Awareness (INA+)"
27 HKLM\SYSTEM\ControlSet001\Services\IPRIP\ObjectName: "LocalSystem"
28 HKLM\SYSTEM\ControlSet001\Services\IPRIP>Description: "Depends INA+, Collects and stores network configuration and
- location information, and notifies applications when this information changes."
29 HKLM\SYSTEM\ControlSet001\Services\IPRIP\DependOnService: 'RpcSs'
30 HKLM\SYSTEM\CurrentControlSet\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 00 00 02 00 60 00 04 00 00
- 00 00 00 14 00 FD 01 02 00 01 01 00 00 00 00 00 05 12 00 00 00 00 18 00 FF 01 0F 00 01 02 00 00 00 00 05 20
- 00 00 00 20 02 00 00 00 00 14 00 8D 01 02 00 01 01 00 00 00 00 00 05 0B 00 00 00 00 18 00 FD 01 02 00 01 02 00
- 00 00 00 05 20 00 00 00 23 02 00 00 01 01 00 00 00 00 00 05 12 00 00 00 01 01 00 00 00 00 05 12 00 00 00
31 HKLM\SYSTEM\CurrentControlSet\Services\IPRIP\Parameters\ServiceDll: "C:\Documents and Settings\Administrator\桌面\
- BinaryCollection\Chapter_3L\Lab03-02.dll"
32 HKLM\SYSTEM\CurrentControlSet\Services\IPRIP\Type: 0x00000020
33 HKLM\SYSTEM\CurrentControlSet\Services\IPRIP\Start: 0x00000002
34 HKLM\SYSTEM\CurrentControlSet\Services\IPRIP\ErrorControl: 0x00000001
```

发现了恶意代码将自身安装为IPRIP服务，并将ImagePath设置为svchost.exe路径来执行它，这意味着恶意代码将会在svchost.exe所在进程启动。另外，其中的 **Description** 和 **DisplayName** 可以作为识别该恶意代码服务的特征。

- Q2：在安装之后，你如何让这个恶意代码运行起来？

通过上述分析，恶意代码原本是一个DLL，但其可以通过安装服务，通过svchost.exe来运行，通过 **net start IPRIP** 可以启动该服务，也就相当于开始运行恶意代码：

```
C:\WINDOWS\system32\cmd.exe

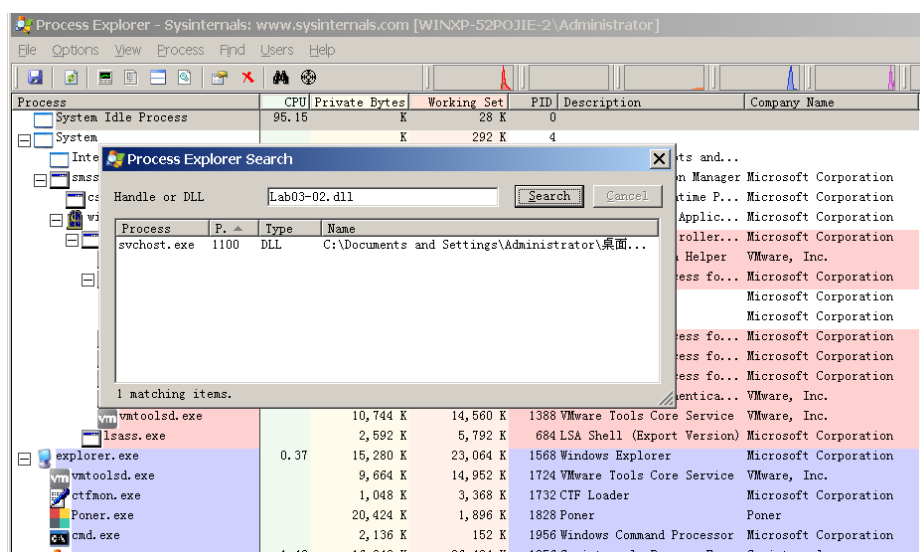
C:\Documents and Settings\Administrator\桌面\BinaryCollection\Chapter_3>rundll1
2.exe Lab03-02.dll install1A

C:\Documents and Settings\Administrator\桌面\BinaryCollection\Chapter_3>net st
rt IPRIP
Intranet Network Awareness (INA+) 服务正在启动 .
Intranet Network Awareness (INA+) 服务已经启动成功。

C:\Documents and Settings\Administrator\桌面\BinaryCollection\Chapter_3>
```

- Q3: 你怎么能找到这个恶意代码是在哪个进程下运行的?

由上述分析, 我们可以在procexp使用查找dll功能, 果然找到了该恶意代码:

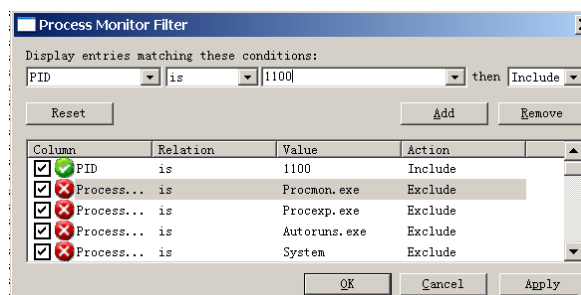


我们可以清楚地看到恶意代码运行在PID为1100的svchost.exe进程下。

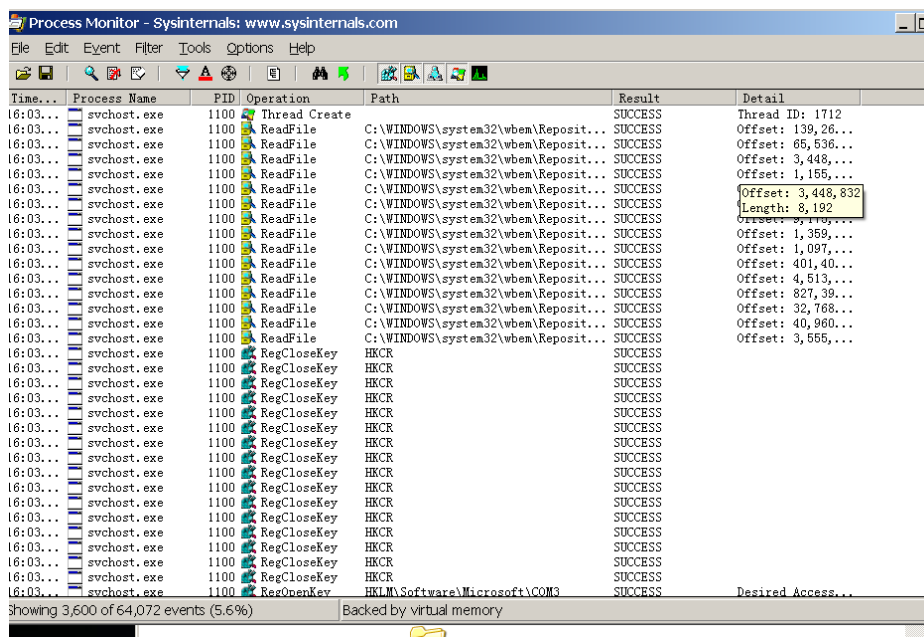
另查阅资料可知, svchost.exe是 Microsoft Windows 操作系统中的一个系统进程, 用于托管和执行多个系统服务和动态链接库 (DLL) 文件。这个进程的名称代表 "Service Host", 它允许多个系统服务在单个进程中运行, 以提高系统的效率和稳定性。

- Q4: 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?

我们知道该DLL运行在PID1100下, 因此可以设置如下过滤器:



可以监控到该恶意代码的所有操作:

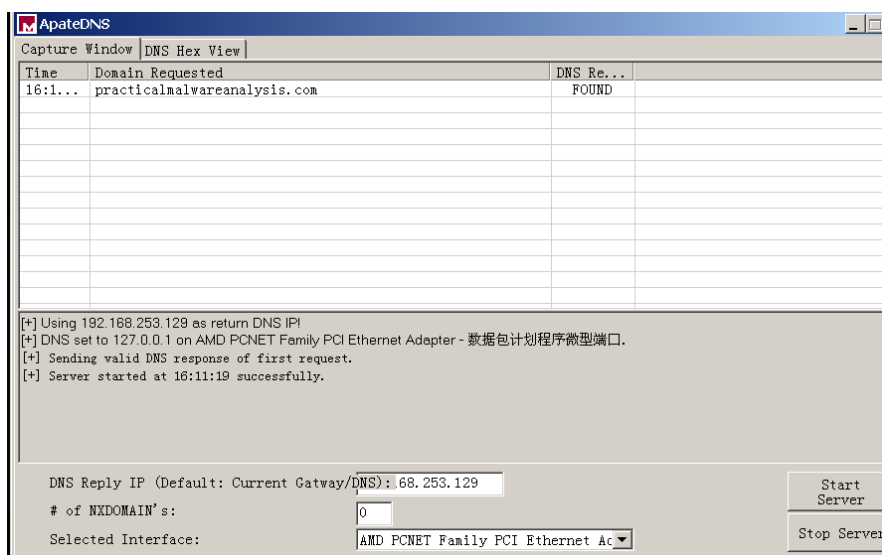


• Q5: 这个恶意代码在主机上的感染迹象特征是什么？

在Q2中我们动态分析了恶意代码安装服务的过程，其中的 **Description** 和 **DisplayName** 可以作为识别该恶意代码服务的特征，即将服务 **Intranet Network Awareness** 安装到了注册表 **HKLM\SYSTEM\CurrentControlSet\Services\IPRIP\Parameters\ServiceDll:%CurrentDirectory%\Lab03-02.dll** 中，可以作为感染迹象特征。

• Q6: 这个恶意代码是否存在一些有用的网络特征码？

我们同样使用ApateDNS来对恶意代码的网络特征进行监控，打开服务一段时间，发现如下信息：



同时使用Netcat对80端口进行监控，获得如下信息：

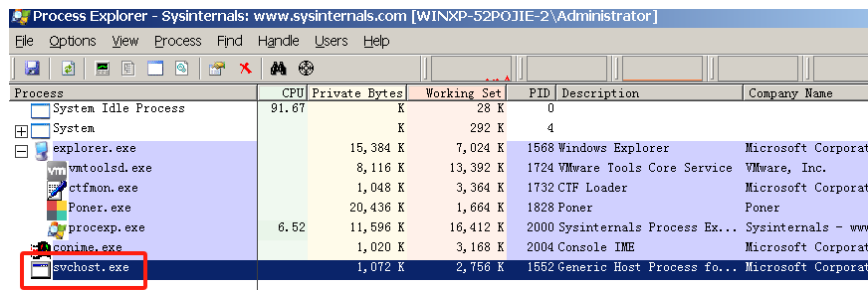
```
GET /serve.html HTTP/1.1
Accept: */*
User-Agent: xp-a70959d1199f Windows XP 6.11
Host: practicalmalwareanalysis.com
```

可见恶意代码对该网址发送了一个GET请求，可以作为其网络特征码。

3.4 Lab03-03.exe

- Q1: 当你使用 Process Explorer 工具进行监视时，你注意到了什么？

先打开procexp进行监视，然后运行Lab03-03.exe，发现新增加了一个进程：



Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
System Idle Process	91.67	K	28 K	0		
System		K	292 K	4		
explorer.exe		15,384 K	7,024 K	1568	Windows Explorer	Microsoft Corporat
vmtoolsd.exe		8,116 K	13,392 K	1724	VMware Tools Core Service	VMware, Inc.
ctfmon.exe		1,048 K	3,364 K	1732	CTF Loader	Microsoft Corporat
Poner.exe		20,436 K	1,664 K	1828	Poner	Poner
procexp.exe	6.52	11,596 K	16,412 K	2000	Sysinternals Process Ex...	Sysinternals - www
conime.exe		1,020 K	3,168 K	2004	Console IME	Microsoft Corporat
svchost.exe		1,072 K	2,756 K	1552	Generic Host Process fo...	Microsoft Corporat

Type	Name
Desktop	\Default
Directory	\KnownDlls
Directory	\Windows
Directory	\BaseNamedObjects
Event	\BaseNamedObjects\userenv: User Profile setup event
File	C:\Documents and Settings\Administrator\桌面\BinaryCollection\Chapter_3L
File	\Device\KsecDD
File	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0...
Key	HKLM
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Drivers32
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Drivers32
Key	HKCU
KeyedEvent	\KernelObjects\CritSecOutOfMemoryEvent
Mutant	\BaseNamedObjects\SHMLIB_LOC_MUTEX
Mutant	\BaseNamedObjects\CTF_LIBES_MutexDefaultS-1-5-21-1844237615-2147222845-725345...

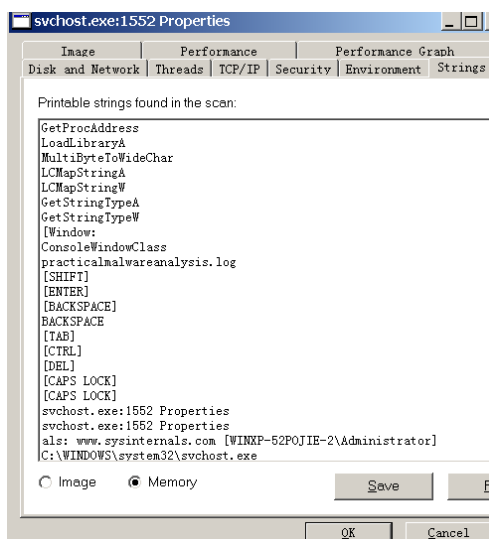
CPU Usage: 8.33% Commit Charge: 3.22% Processes: 24 Physical Usage: 11.05%

即svchost.exe进程，PID为1552，并且这个进程没有父进程，十分可疑，因为svchost.exe通常是service.exe的子进程。

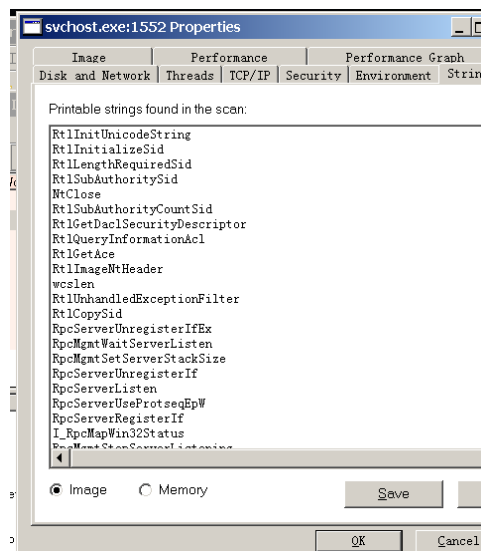
- Q2: 你可以找出任何的内存修改行为吗？

使用procexp对内存中的该svchost.exe和磁盘的svchost.exe作对比如下：

内存中：



磁盘中：



可以看到，内存中的字符串出现了 `practicalmalwareanalysis.log`、`[ENTER]`、`[CTRL]` 等异常字符串，一定和恶意代码的功能相关，据此可以判断恶意代码篡改了内存。

- Q3: 这个恶意代码在主机上的感染迹象特征是什么？

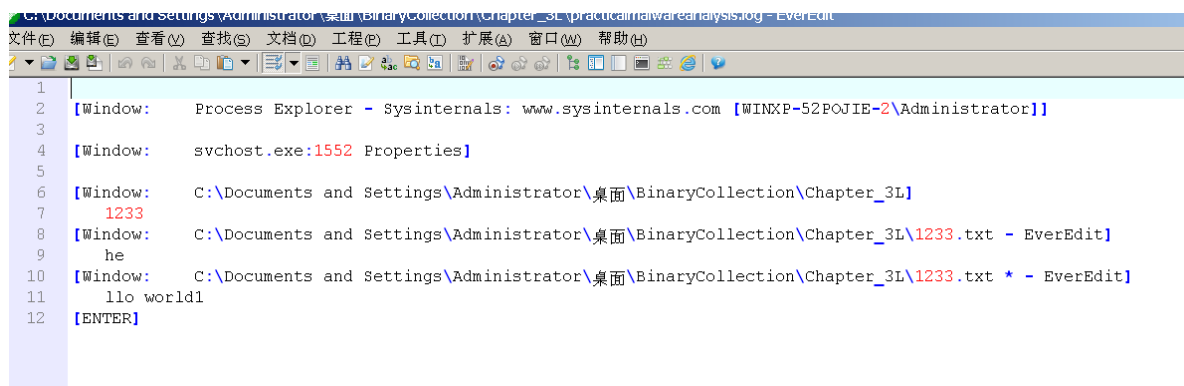
由上述分析，`practicalmalwareanalysis.log` 是一个日志文件名，`[ENTER]`、`[CTRL]` 等是键盘按键事件，可以推测恶意代码正在偷偷捕捉键盘事件，然后保存在上述文件中。实现一个键盘的窃听效果，据此文件可以发现主机的感染迹象特征。

- Q4: 这个恶意代码的目的是什么？

创建任意一个txt文件 `123.txt`，在其中键入 `Hello World!`：



然后在 `practicalmalwareanalysis.log` 就可以发现键盘的记录：

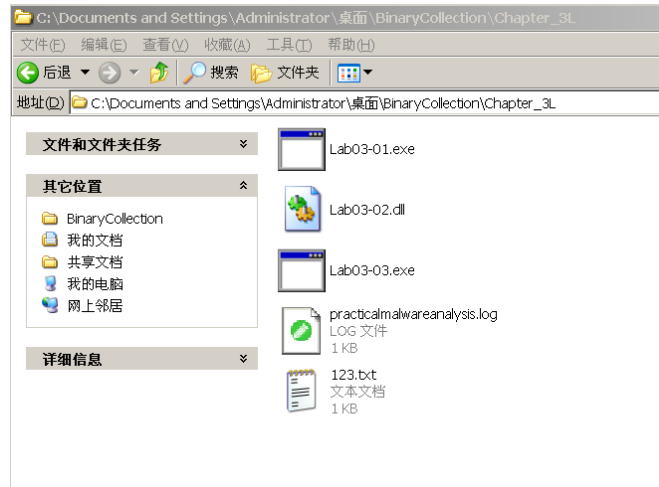


结合上述分析可知该恶意代码是一个键盘按键窃听器。

3.5 Lab03-04.exe

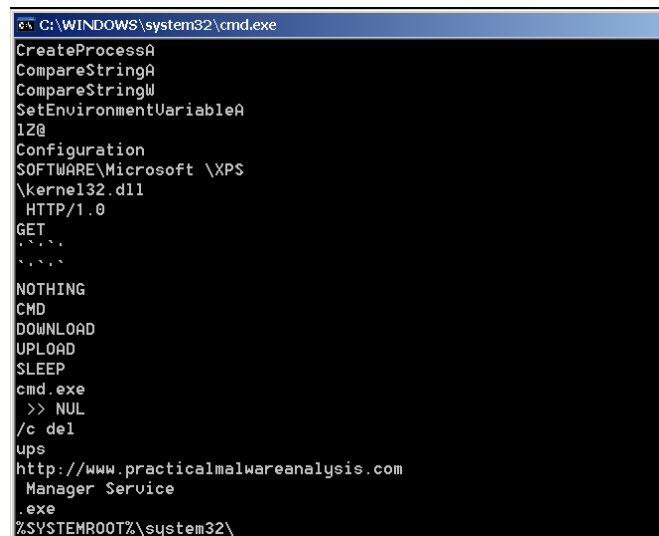
- Q1: 当你运行这个文件时，会发生什么呢？

运行文件，发现该exe被自我删除了：



- Q2: 是什么原因造成动态分析无法有效实施？

执行该文件会直接删除，因此恢复快照后先静态分析，查看字符串：



发现有 `cmd.exe`，`del` 等字样，推测恶意代码使用命令行来对自身进行删除，防止其被动态分析。

我们用Procmon设置进程名是Lab03-04.exe的过滤器，运行代码。虽然程序被删除了，但procmon记录了它的动作：

Time...	Process Name	PID	Operation	Path	Result	Detail
17:15...	Lab03-04.exe	1580	Process Start		SUCCESS	Parent PID: 15...
17:15...	Lab03-04.exe	1580	Thread Create		SUCCESS	Thread ID: 1240
17:15...	Lab03-04.exe	1580	QueryNameInf...	C:\Documents and Settings\Admini...	SUCCESS	Name: \Documen...
17:15...	Lab03-04.exe	1580	Load Image	C:\Documents and Settings\Admini...	SUCCESS	Image Base: 0x...
17:15...	Lab03-04.exe	1580	Load Image	C:\WINDOWS\system32\ntdll.dll	SUCCESS	Image Base: 0x...
17:15...	Lab03-04.exe	1580	QueryNameInf...	C:\Documents and Settings\Admini...	SUCCESS	Name: \Documen...
17:15...	Lab03-04.exe	1580	CreateFile	C:\WINDOWS\Prefetch\LAB03-04.EXE...	NAME NOT FOUND	Desired Access...
17:15...	Lab03-04.exe	1580	RegOpenKey	HKLM\Software\Microsoft\Windows ...	NAME NOT FOUND	Desired Access...
17:15...	Lab03-04.exe	1580	CreateFile	C:\Documents and Settings\Admini...	SUCCESS	Desired Access...
17:15...	Lab03-04.exe	1580	FileSystemCo...	C:\Documents and Settings\Admini...	SUCCESS	Desired Access...
17:15...	Lab03-04.exe	1580	QueryOpen	C:\Documents and Settings\Admini...	NAME NOT FOUND	Control: FSCTL...
17:15...	Lab03-04.exe	1580	Load Image	C:\WINDOWS\system32\kernel32.dll	SUCCESS	Image Base: 0x...
17:15...	Lab03-04.exe	1580	RegOpenKey	HKLM\System\CurrentControlSet\Co...	SUCCESS	Desired Access...
17:15...	Lab03-04.exe	1580	RegQueryValue	HKLM\System\CurrentControlSet\Co...	SUCCESS	Type: REG_DWOR...
17:15...	Lab03-04.exe	1580	RegCloseKey	HKLM\System\CurrentControlSet\Co...	SUCCESS	
17:15...	Lab03-04.exe	1580	RegOpenKey	HKLM\Software\Microsoft\Windows ...	NAME NOT FOUND	Desired Access...
17:15...	Lab03-04.exe	1580	Load Image	C:\WINDOWS\system32\advapi32.dll	SUCCESS	Image Base: 0x...
17:15...	Lab03-04.exe	1580	Load Image	C:\WINDOWS\system32\rpcrt4.dll	SUCCESS	Image Base: 0x...
17:15...	Lab03-04.exe	1580	Load Image	C:\WINDOWS\system32\securl32.dll	SUCCESS	Image Base: 0x...
17:15...	Lab03-04.exe	1580	Load Image	C:\WINDOWS\system32\shell32.dll	SUCCESS	Image Base: 0x...
17:15...	Lab03-04.exe	1580	Load Image	C:\WINDOWS\system32\gdi32.dll	SUCCESS	Image Base: 0x...
17:15...	Lab03-04.exe	1580	Load Image	C:\WINDOWS\system32\user32.dll	SUCCESS	Image Base: 0x...
17:15...	Lab03-04.exe	1580	Load Image	C:\WINDOWS\system32\ole32.dll	SUCCESS	Image Base: 0x...
17:15...	Lab03-04.exe	1580	Load Image	C:\WINDOWS\system32\oleaut32.dll	SUCCESS	Image Base: 0x...
17:15...	Lab03-04.exe	1580	Load Image	C:\WINDOWS\system32\shlwapi.dll	SUCCESS	Image Base: 0x...
17:15...	Lab03-04.exe	1580	QueryOpen	C:\Documents and Settings\Admini...	NAME NOT FOUND	
17:15...	Lab03-04.exe	1580	QueryOpen	C:\WINDOWS\system32\ws2_32.dll	SUCCESS	CreationTime: ...
17:15...	Lab03-04.exe	1580	CreateFile	C:\WINDOWS\system32\ws2_32.dll	SUCCESS	Desired Access...
17:15...	Lab03-04.exe	1580	CreateFileMa...	C:\WINDOWS\system32\ws2_32.dll	SUCCESS	SyncType: Sync...

Showing 1,870 of 51,949 events (3.5%) Backed by virtual memory

该恶意代码可能创建了进程来删除自身，因此继续筛选：

Time...	Process Name	PID	Operation	Path	Result	Detail
17:15...	Lab03-04.exe	1580	Process Create	C:\WINDOWS\system32\cmd.exe	SUCCESS	PID: 432, Com...

发现它通过Process Create的操作，执行了以下命令：

```
1 "C:\WINDOWS\system32\cmd.exe" /c del C:\Documents and
  Settings\Administrator\桌面\BinaryCollection\Chapter_3L\Lab03-04.exe >> NUL
```

通过此命令，恶意代码将自身删除。

- Q3: 是否有其他方式来运行这个程序？

目前暂未发现有效方法能阻止该恶意代码的自我删除。

综合以上分析，我们可以写出如下的Yara规则，对上述的几个样本进行查杀：

```
1 //首先判断是否为PE文件
2 private rule IsPE
3 {
4 condition:
5     filesize < 10MB and //这几个PE均小于10MB
6     uint16(0) == 0x5A4D and //"MZ"头
7     uint32(uint32(0x3C)) == 0x00004550 // "PE"头
8 }
9
10 //Lab03-01
11 rule lab3_1
12 {
13 strings:
14     $s1 = "vmx32to64.exe"
15     $s2 = "WinVMX32-"
16 condition:
17     IsPE and $s1 and $s2
18 }
19
20
21 //Lab03-02
22 rule lab3_2
23 {
24 strings:
25     $s1 = "IPRIP"
26     $s2 = "svchost.exe"
27 condition:
28     IsPE and $s1 and $s2
29 }
30
31 //Lab03-03
32 rule lab3_3
33 {
34 strings:
35     $s1 = "svchost.exe"
36     $s2 = "Sleep"
37     $s3 = "ntdll.dll"
38 condition:
39     IsPE and $s1 and $s2 and $s3
40 }
41
42 //Lab03-04
43 rule lab3_4
```

```

44 {
45     strings:
46         $s1 = "/c del"
47         $s2 = "cmd.exe"
48         $s3 = " >> NUL"
49     condition:
50         IsPE and $s1 and $s2 and $s3
51 }

```

4 实验结论及心得体会

把上述Yara规则保存为 `rule_ex3.yar`，然后在Chapter_3L上一个目录输入以下命令：

```
1 | yara64 -r rule_ex3.yar Chapter_3L
```

结果如下，样本全部检测成功：

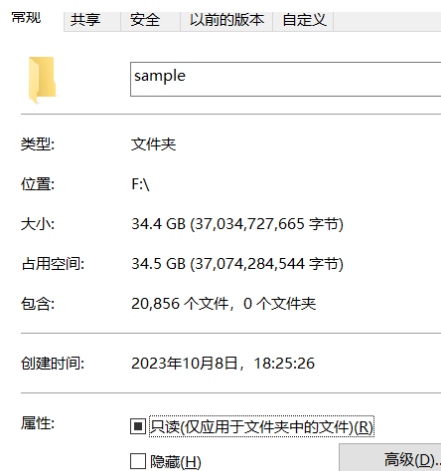
```

(C) Microsoft Corporation. 保留所有权利。
D:\study\大三\恶意代码分析与防治技术\Practical Malware Analysis Labs\BinaryCollection>yara64 -r rule_ex3.yar Chapter_3L
Lab3_1 Chapter_3L\Lab03-01.exe
Lab3_2 Chapter_3L\Lab03-02.dll
Lab3_4 Chapter_3L\Lab03-04.exe
Lab3_3 Chapter_3L\Lab03-03.exe
D:\study\大三\恶意代码分析与防治技术\Practical Malware Analysis Labs\BinaryCollection>_

```

接下来对运行 `Scan.py` 获得的`sample`进行扫描。

sample文件夹大小为34.4GB，含有20856个可执行文件：



我们编写一个脚本 `yara_unittest.py` 来完成扫描：

```

1 import os
2 import yara
3 import datetime
4
5 # 定义YARA规则文件路径
6 rule_file = './rule_ex3.yar'

```

```

7
8 # 定义要扫描的文件夹路径
9 folder_path = './sample/'
10
11 # 加载YARA规则
12 try:
13     rules = yara.compile(rule_file)
14 except yara.SyntaxError as e:
15     print(f"YARA规则语法错误: {e}")
16     exit(1)
17
18 # 获取当前时间
19 start_time = datetime.datetime.now()
20
21 # 扫描文件夹内的所有文件
22 scan_results = []
23
24 for root, dirs, files in os.walk(folder_path):
25     for file in files:
26         file_path = os.path.join(root, file)
27         try:
28             matches = rules.match(file_path)
29             if matches:
30                 scan_results.append({'file_path': file_path, 'matches':
[]}])
31         except Exception as e:
32             print(f"扫描文件时出现错误: {file_path} - {str(e)}")
33
34 # 计算扫描时间
35 end_time = datetime.datetime.now()
36 scan_time = (end_time - start_time).seconds
37
38 # 将扫描结果写入文件
39 output_file = './scan_results.txt'
40
41 with open(output_file, 'w') as f:
42     f.write(f"扫描开始时间: {start_time.strftime('%Y-%m-%d %H:%M:%S')}\n")
43     f.write(f"扫描耗时: {scan_time}s\n")
44     f.write("扫描结果:\n")
45     for result in scan_results:
46         f.write(f"文件路径: {result['file_path']}\n")
47         f.write(f"匹配规则: {', '.join(result['matches'])}\n")
48         f.write('\n')
49
50 print(f"扫描完成, 耗时{scan_time}秒, 结果已保存到 {output_file}")
51

```

运行得到扫描结果文件如下：

```
1 扫描开始时间：2023-10-06 20:17:47
2 扫描耗时：87s
3 扫描结果：
4 文件路径：./sample/daemon.dll
5 匹配规则：lab3_3
6
7 文件路径：./sample/em004_64.dll
8 匹配规则：lab3_3
9
10 文件路径：./sample/HipsDaemon.exe
11 匹配规则：lab3_3
12
13 文件路径：./sample/Lab03-01.exe
14 匹配规则：lab3_1
15
16 文件路径：./sample/Lab03-02.dll
17 匹配规则：lab3_2
18
19 文件路径：./sample/Lab03-03.exe
20 匹配规则：lab3_3
21
22 文件路径：./sample/Lab03-04.exe
23 匹配规则：lab3_4
24
25 文件路径：./sample/Lab05-01.dll
26 匹配规则：lab3_3
27
28 文件路径：./sample/Lab09-01.exe
29 匹配规则：lab3_4
30
31 文件路径：./sample/Lab12-02.exe
32 匹配规则：lab3_3
33
34 文件路径：./sample/Lab16-01.exe
35 匹配规则：lab3_4
36
37 文件路径：./sample/Lab16-03.exe
38 匹配规则：lab3_4
39
40 文件路径：./sample/Lab17-01.exe
41 匹配规则：lab3_4
42
43 文件路径：./sample/Lab17-02.dll
44 匹配规则：lab3_3
```

```
45
46 文件路径: ./sample/Lab17-03.exe
47 匹配规则: lab3_3
48
49 文件路径: ./sample/SangforCSClient.exe
50 匹配规则: lab3_3
51
52 文件路径: ./sample/schedsvc.dll
53 匹配规则: lab3_3
54
55 文件路径: ./sample/SenseTVM.exe
56 匹配规则: lab3_3
57
58
```

可以发现若干可执行文件被扫描识别，共耗时87s，效率还算满意。

心得体会：通过此次实验，我学会了通过动态分析来分析病毒功能的能力，对Process Explorer，Process Monitor等软件有了更深入的了解，更熟练它们的使用，也更进一步掌握了yara编写的方法。