

南開大學

《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_____2112060
姓 名_____孙露
班 级_____信息安全 1 班

《恶意代码分析与防治技术》课程 Lab3 实验报告

一、 实验目的	3
二、 实验原理	3
三、 实验过程	3
(一) Lab3-1	3
(二) Lab3-2	8
(三) Lab3-3	13
(四) Lab3-4	16
四、 实验结论及心得体会	18

一、 实验目的

使用动态分析基础技术来分析恶意代码，包括查找恶意代码的导入函数和字符串列表，监视恶意代码的行为，识别感染迹象特征，以及尝试运行和分析恶意代码的行为。

二、 实验原理

使用各种工具和技术，包括 PView、strings、Process Explorer、Procmon、Wireshark 等，用于动态分析恶意代码的行为和特征。

通过监视进程、文件、注册表等的变化，可以识别恶意代码的行为和感染迹象。

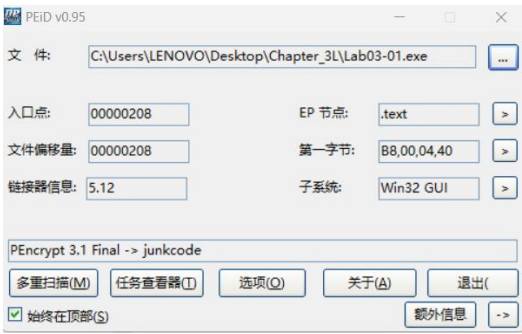
三、 实验过程

(一) Lab3-1

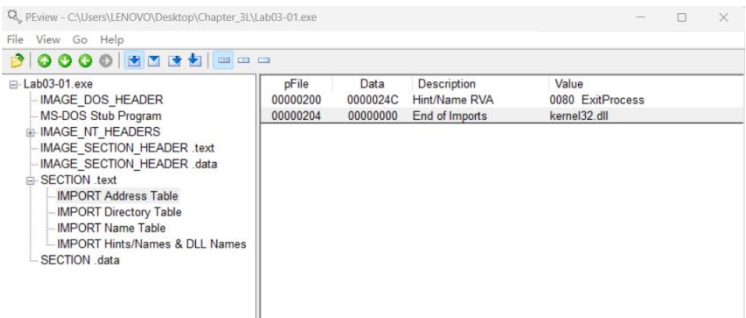
使用动态分析基础技术来分析在 Lab03-01.exe 文件中发现的恶意代码。

(1) 找出这个恶意代码的导入函数与字符串列表。

① 用 PEiD 打开 Lab03-01.exe，可以看到 Lab03-01.exe 是加壳的，是 PEncrypt 3.1 Final -> junkcode 的壳。



② 使用 PView 查看 Lab03-01.exe 的 PE 文件结构和字符串列表，PView 工具显示该程序导入函数只有 ExitProcess，没有其他导入函数。



③ 使用 strings.exe 查看恶意代码中有哪些字符串。使用命令 strings Lab03-01.exe，如下图所示。

```
hK7
Pj
Cz
V
uP
StubPath
SOFTWARE\Classes\http\shell\open\commandV
Software\Microsoft\Active Setup\Installed Components\
test
www.persico.com
admin
VideoDriver
WinVMX32
vmx32to64.exe
U
SOFTWARE\Microsoft\Windows\CurrentVersion\Run
mQ
VSh
V1
VQe
V)e
V)e
V)V
U
ul e
SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
F1
AmDns
ti
B
jth
VQj
Vi#
V%X
t(e
```

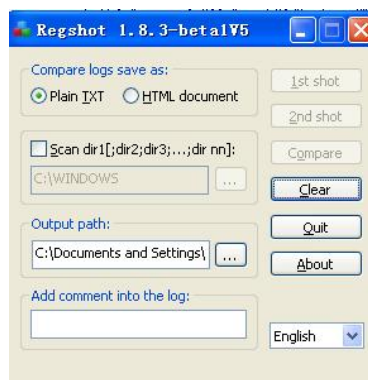
看到了很多的字符串，如注册表位置、域名、WinVMX32、VideoDriver、vmx32to64.exe 等。其中方框标记的内容中包含很多注册表信息、一个域名信息和一个应用程序。

猜测分析可能会通过连接访问该网址下载某些木马文件或者通过 vmx32to64.exe 下载打开某些后门。

(2) 这个恶意代码在主机上的感染迹象特征是什么？

① 使用 RegShot 工具分析注册表变化。

RegShot 是一种注册表比较工具，它通过两次抓取注册表而快速地比较出答案。它还可以将注册表以纯文本方式记录下来，便于浏览；还可以监察 Win.ini, System.ini 中的键值；还可以监察 Windows 目录和 System 目录中文件的变化，为手工卸载某些软件创造条件。通过扫描并保存注册表的“快照”，并对两次快照进行自动的对比，找出快照间存在的不同之处，结果保存成 txt 或者 html 文档。



```

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\VideoDriver: 43 3A 5C 57 49 4E 44 4F 57 53 5C
HKLM\SYSTEM\ControlSet001\Services\PerfDisk\Performance\WbemAdapFileSignature: 82 4D 6F 69 C8 E9
HKLM\SYSTEM\ControlSet001\Services\PerfDisk\Performance\WbemAdapFileTime: 00 A0 48 BA 4B A1 C8 01
HKLM\SYSTEM\ControlSet001\Services\PerfDisk\Performance\WbemAdapFileSize: 0x00006400
HKLM\SYSTEM\ControlSet001\Services\PerfDisk\Performance\WbemAdapStatus: 0x00000000
HKLM\SYSTEM\ControlSet001\Services\PerfNet\Performance\WbemAdapFileSignature: 06 83 58 0D B1 DD
HKLM\SYSTEM\ControlSet001\Services\PerfNet\Performance\WbemAdapFileTime: 00 A0 48 BA 4B A1 C8 01
HKLM\SYSTEM\ControlSet001\Services\PerfNet\Performance\WbemAdapFileSize: 0x00003E00
HKLM\SYSTEM\ControlSet001\Services\PerfNet\Performance\WbemAdapStatus: 0x00000000
HKLM\SYSTEM\ControlSet001\Services\PerfFS\Performance\WbemAdapFileSignature: 59 6E 4F 8C A6 11 5
HKLM\SYSTEM\ControlSet001\Services\PerfFS\Performance\WbemAdapFileTime: 00 A0 48 BA 4B A1 C8 01
HKLM\SYSTEM\ControlSet001\Services\PerfFS\Performance\WbemAdapFileSize: 0x00005A00
HKLM\SYSTEM\ControlSet001\Services\PerfFS\Performance\WbemAdapStatus: 0x00000000
HKLM\SYSTEM\ControlSet001\Services\PerfProc\Performance\WbemAdapFileSignature: 64 DF B9 8A DD 4E
HKLM\SYSTEM\ControlSet001\Services\PerfProc\Performance\WbemAdapFileTime: 00 A0 48 BA 4B A1 C8 01
HKLM\SYSTEM\ControlSet001\Services\PerfProc\Performance\WbemAdapFileSize: 0x00008200
HKLM\SYSTEM\ControlSet001\Services\PerfProc\Performance\WbemAdapStatus: 0x00000000
HKLM\SYSTEM\ControlSet001\Services\PSched\Performance\WbemAdapFileSignature: 81 A2 82 4E 91 DE F
HKLM\SYSTEM\ControlSet001\Services\PSched\Performance\WbemAdapFileTime: 00 A0 48 BA 4B A1 C8 01
HKLM\SYSTEM\ControlSet001\Services\PSched\Performance\WbemAdapFileSize: 0x00002A00
HKLM\SYSTEM\ControlSet001\Services\PSched\Performance\WbemAdapStatus: 0x00000001
HKLM\SYSTEM\CurrentControlSet\Services\PerfDisk\Performance\WbemAdapFileSignature: 82 4D 6F 69 C
HKLM\SYSTEM\CurrentControlSet\Services\PerfDisk\Performance\WbemAdapFileTime: 00 A0 48 BA 4B A1
HKLM\SYSTEM\CurrentControlSet\Services\PerfDisk\Performance\WbemAdapFileSize: 0x00006400
HKLM\SYSTEM\CurrentControlSet\Services\PerfDisk\Performance\WbemAdapStatus: 0x00000000
HKLM\SYSTEM\CurrentControlSet\Services\PerfNet\Performance\WbemAdapFileSignature: 06 83 58 0D B1
HKLM\SYSTEM\CurrentControlSet\Services\PerfNet\Performance\WbemAdapFileTime: 00 A0 48 BA 4B A1 C

```

快照比较分析得出该恶意软件在

HKEY_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\VideoDriver 下添加了一个名为VideoDriver的自启动项：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，程序vmx32to64.exe在开机时会自启动。

② 使用 Process explorer 分析

先运行 procmon 工具，并清除所有事件；启动 Process Explorer，同时配置出一个虚拟网络，包括 ApateDNS、Netcat (监听端口 80 和 443)，以及用于网络数据包捕获的 Wireshark。

Process	CPU Private B...	Working Set	PID	Description	Company Name
System Idle Process	100.00	K	28	K	0
System		K	300	K	4
Interrupts	< 0.01	K			n/a Hardware Interrupts a...
smss.exe	168 K	404 K	356	Windows NT Session Ma...	Microsoft Corporation
csrss.exe	1,764 K	5,076 K	580	Client Server Runtime...	Microsoft Corporation
winlogon.exe	7,112 K	4,164 K	608	Windows NT Logon Appl...	Microsoft Corporation
services.exe	1,752 K	3,452 K	708	Services and Controll...	Microsoft Corporation
vmacthlp.exe	688 K	2,620 K	872	VMware Activation Helper	VMware, Inc.
svchost.exe	3,152 K	4,956 K	888	Generic Host Process ...	Microsoft Corporation
smiprvse.exe	3,888 K	8,768 K	1292	WMI	Microsoft Corporation
smiprvse.exe	2,068 K	5,052 K	244	WMI	Microsoft Corporation
svchost.exe	1,860 K	4,460 K	948	Generic Host Process ...	Microsoft Corporation
svchost.exe	13,396 K	21,324 K	1088	Generic Host Process ...	Microsoft Corporation
wsentfy.exe	668 K	2,496 K	1316	Windows Security Cent...	Microsoft Corporation
wsuclt.exe	6,576 K	6,772 K	380	Automatic Updates	Microsoft Corporation
svchost.exe	1,388 K	3,672 K	1136	Generic Host Process ...	Microsoft Corporation
svchost.exe	1,836 K	4,636 K	1176	Generic Host Process ...	Microsoft Corporation
spoolsv.exe	4,412 K	6,840 K	1388	Spooler SubSystem App	Microsoft Corporation
svchost.exe	2,292 K	3,388 K	2024	Generic Host Process ...	Microsoft Corporation
VGAuthService...	6,280 K	9,136 K	188	VMware Guest Authent...	VMware, Inc.
vmtoolsd.exe	11,072 K	14,192 K	408	VMware Tools Core Ser...	VMware, Inc.
alg.exe	1,264 K	3,724 K	1296	Application Layer Gat...	Microsoft Corporation
lsass.exe	3,976 K	6,264 K	720	LSA Shell (Export Ver...	Microsoft Corporation
mpabaln.exe	1,040 K	3,188 K	1188	Windows WPA Balloon R...	Microsoft Corporation
explorer.exe	13,472 K	19,780 K	1736	Windows Explorer	Microsoft Corporation
rundll32.exe	2,360 K	3,684 K	1844	Run a DLL as an App	Microsoft Corporation
vmtoolsd.exe	9,744 K	14,648 K	1852	VMware Tools Core Ser...	VMware, Inc.
ctfmon.exe	1,056 K	3,716 K	1872	CTF Loader	Microsoft Corporation
Procmon.exe	11,588 K	14,676 K	1688	Process Monitor	Sysinternals - www...
Lab03-01.exe	704 K	2,104 K	972		
procexp.exe	11,192 K	15,396 K	2460	Sysinternals Process ...	Sysinternals - www...

选择 View→Lower Pane View→Handles，可以看到，恶意代码已经创建了一个名为WinVMX32的互斥量。

Type	Name
Desktop	\Default
Directory	\KnownDlls
Directory	\Windows
Directory	\BaseNamedObjects
File	C:\Documents and Settings\lulu\桌面\Practical Malware Analysis Labs\Bin...
File	\Device\KsecDD
Key	HKLM\SYSTEM\ControlSet001\Services\WinSock2\Parameters\Protocol_Catalog9
Key	HKLM\SYSTEM\ControlSet001\Services\WinSock2\Parameters\NameSpace_Catalog5
KernelEvent	\KernelObjects\CritSecOutOfMemoryEvent
Mutant	\BaseNamedObjects\WinVX32
Thread	Lab03-01.exe(1960): 1956
Thread	Lab03-01.exe(1960): 1956
WindowStation	\Windows\WindowStations\WinSta0
WindowStation	\Windows\WindowStations\WinSta0

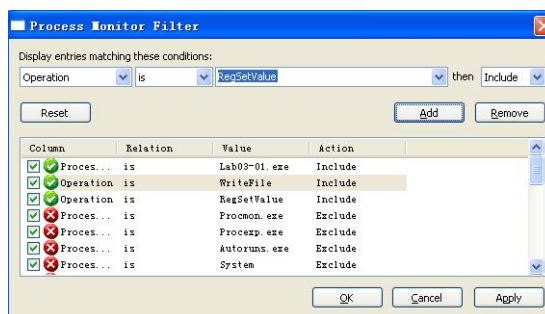
选择 View→Lower Pane View→DLL, 看到恶意代码动态装载的 DLL 文件。

Name	Description	Company Name	Path
Lab03-01.exe			C:\Documents and Settings\lulu\桌面\Practical ...
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
securlib.dll	Security Support Provider ...	Microsoft Corporation	C:\WINDOWS\system32\securlib.dll
setupapi.dll	Windows Setup API	Microsoft Corporation	C:\WINDOWS\system32\setupapi.dll
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
winmr.dll	LDAP RnR Provider DLL	Microsoft Corporation	C:\WINDOWS\system32\winmr.dll
ws2_32.dll	Win32 LDAP API DLL	Microsoft Corporation	C:\WINDOWS\system32\ws2_32.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 ...	Microsoft Corporation	C:\WINDOWS\system32\ws2help.dll
wsbth.dll	Windows Sockets Helper DLL	Microsoft Corporation	C:\WINDOWS\system32\wsbth.dll
wshtcpip.dll	Windows Sockets Helper DLL	Microsoft Corporation	C:\WINDOWS\system32\wshtcpip.dll

ws2_32.dll 和 wshtcpip.dll 库文件的存在说明该样本存在网络方面的操作。

③ 使用 Process Monitor 分析

选择 Filter→Filter, 来呼出过滤器对话框, 然后设置三个过滤器: 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, 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, 其中只有 1 个 WriteFile 操作的条目, 并有 9 个关于 RegSetValue 操作的条目。

HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed 键值上的 RegSetValue 操作是典型的噪声。第四条往后都有 Seed, Seed 说明该程序用了随机数, 随机数发生器的种子会有软件在注册表中不停地更新。

Process Name	PID	Operation	Path	Result	Detail
Lab03-01.exe	972	RegSetValue	HKLM\SOFTWARE\Microsoft\Crypto...	SUCCESS	Type: REG_BIN...
Lab03-01.exe	972	WriteFile	C:\WINDOWS\system32\vmx32to64.exe	SUCCESS	Offset: 0, Le...
Lab03-01.exe	972	RegSetValue	HKLM\SOFTWARE\Microsoft\Window...	SUCCESS	Type: REG_SZ...
Lab03-01.exe	972	RegSetValue	HKLM\SOFTWARE\Microsoft\Crypto...	SUCCESS	Type: REG_BIN...
Lab03-01.exe	972	RegSetValue	HKLM\SOFTWARE\Microsoft\Crypto...	SUCCESS	Type: REG_BIN...
Lab03-01.exe	972	RegSetValue	HKLM\SOFTWARE\Microsoft\Crypto...	SUCCESS	Type: REG_BIN...
Lab03-01.exe	972	RegSetValue	HKLM\SOFTWARE\Microsoft\Crypto...	SUCCESS	Type: REG_BIN...
Lab03-01.exe	972	RegSetValue	HKLM\SOFTWARE\Microsoft\Crypto...	SUCCESS	Type: REG_BIN...
Lab03-01.exe	972	RegSetValue	HKLM\SOFTWARE\Microsoft\Crypto...	SUCCESS	Type: REG_BIN...
Lab03-01.exe	972	RegSetValue	HKLM\SOFTWARE\Microsoft\Crypto...	SUCCESS	Type: REG_BIN...

双击 WriteFile 操作条目，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，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 WinMD5 查看 vmx32to64.exe 与 Lab03-01.exe 的 MD5 值，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，这告诉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。



双击红框中的第二行，发现恶意代码在 Windows\System32 路径下创建写入来 vmx32to64.exe 文件并且在 CurrentVersion\Run 下创建了 VideoDriver 自启动项键值，用于在系统启动时自动运行 vmx32to64.exe。

综上所述，经过初步分析，可以认为在主机上创建互斥量、尝试复制自身并加入系统自启动项，创建注册表键值等，这些都可以作为其在主机上感染的迹象特征。

(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？

使用 wireshark 进行抓包分析，可以看到如下结果：

192.168.175.255	NBNS	92	Name query NB MSHOME<lb>	
Broadcast	ARP	60	who has 192.168.175.2? Tell 192.168.175.1	
Broadcast	ARP	60	who has 192.168.175.2? Tell 192.168.175.1	
192.168.175.2	DNS	92	Standard query 0xa37c A www.practicalmalwareanalysis.com	
Broadcast	ARP	60	who has 192.168.175.137? Tell 192.168.175.2	
vmware_F9:56:5d	ARP	42	192.168.175.137 is at 00:0c:29:e5:3d:f0	
192.168.175.137	DNS	138	Standard query response 0xa37c CNAME practicalmalwareanalysis.com A 15.197.142.173 A 3.33.152.147	
15.197.142.173	TCP	82	nimreg > https [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1	
Broadcast	ARP	60	who has 192.168.175.2? Tell 192.168.175.1	
192.168.175.255	BROWSE	216	Get Backup List Request	
192.168.175.2	NBNS	92	Name query NB MSHOME<lb>	
Broadcast	ARP	60	who has 192.168.175.2? Tell 192.168.175.1	
192.168.175.2	NBNS	92	Name query NB MSHOME<lb>	
15.197.142.173	TCP	82	[TCP Reset] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM=1	
192.168.175.2	NBNS	92	Name query NB MSHOME<lb>	
192.168.175.255	NBNS	92	Name query NB MSHOME<lb>	
Broadcast	ARP	60	who has 192.168.175.2? Tell 192.168.175.1	
192.168.175.255	NBNS	92	Name query NB MSHOME<lb>	
Broadcast	ARP	60	who has 192.168.175.2? Tell 192.168.175.1	
192.168.175.255	NBNS	92	Name query NB MSHOME<lb>	

出现了网址 www.practicalmalwareanalysis.com。

综上所述，Lab03-01.exe 访问了网址

www.practicalmalwareanalysis.com，可以作为网络依据。

（二） Lab3-2

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？

使用 PEView 查看 Lab03-02.dll 的导出函数

	pFile	Data	Description	Value
Lab03-02.dll	00004D28	00004706	Function RVA	0001 Install
IMAGE_DOS_HEADER	00004D2C	00003196	Function RVA	0002 ServiceMain
MS-DOS Stub Program	00004D30	00004B18	Function RVA	0003 UninstallService
IMAGE_NT_HEADERS	00004D34	00004B0B	Function RVA	0004 installA
IMAGE_SECTION_HEADER .text	00004D38	00004C2B	Function RVA	0005 uninstallA
IMAGE_SECTION_HEADER .rdata				
IMAGE_SECTION_HEADER .data				
IMAGE_SECTION_HEADER .reloc				
SECTION .text				
SECTION .rdata				
IMPORT Address Table				
IMPORT Directory Table				
IMPORT Name Table				
IMPORT Hints/Names & DLL Names				
IMAGE_EXPORT_DIRECTORY				
EXPORT Address Table				
EXPORT Name Pointer Table				
EXPORT Ordinal Table				
EXPORT Names				
SECTION .data				
SECTION .reloc				

查看 directory table,发现了一些依赖导入函数动态链接库

	pFile	Data	Description	Value
Lab03-02.dll	00004E80	000032C0	Import Name Table RVA	
IMAGE_DOS_HEADER	00004E84	00000000	Time Date Stamp	
MS-DOS Stub Program	00004E88	00000000	Forwarder Chain	
IMAGE_NT_HEADERS	00004E9C	00000000	Name RVA	KERNEL32.dll
IMAGE_SECTION_HEADER .text	00004EA0	000032C0	Import Address Table RVA	
IMAGE_SECTION_HEADER .rdata	00004EA4	000032C0	Import Name Table RVA	
IMAGE_SECTION_HEADER .data	00004EA8	00000000	Time Date Stamp	
IMAGE_SECTION_HEADER .reloc	00004EAC	00000000	Forwarder Chain	
SECTION .text	00004EAD	00000000	Name RVA	ADVAPI32.dll
SECTION .rdata	00004EAE	00000000	Import Address Table RVA	
IMPORT Address Table	00004EAF	000032C0	Import Name Table RVA	
IMPORT Directory Table	00004EB0	00000000	Time Date Stamp	
IMPORT Name Table	00004EB4	00000000	Forwarder Chain	
IMPORT Hints/Names & DLL Names	00004EB8	00000000	Name RVA	USER32.dll
IMAGE_EXPORT_DIRECTORY	00004EB9	00000000	Import Address Table RVA	
EXPORT Address Table	00004EBC	00000000	Time Date Stamp	
EXPORT Name Pointer Table	00004EBD	00000000	Forwarder Chain	
EXPORT Ordinal Table	00004EBE	00000000	Name RVA	USER32.dll
EXPORT Names	00004EBF	00000000	Import Address Table RVA	
SECTION .data	00004EC0	00000000	Time Date Stamp	
SECTION .reloc	00004EC1	00000000	Forwarder Chain	
	00004EC2	00000000	Name RVA	MSVCRT.dll
	00004EC3	00000000	Import Address Table RVA	
	00004EC4	00000000	Time Date Stamp	
	00004EC5	00000000	Forwarder Chain	
	00004EC6	00000000	Name RVA	
	00004EC7	00000000	Import Address Table RVA	
	00004EC8	00000000	Time Date Stamp	
	00004EC9	00000000	Forwarder Chain	
	00004ECA	00000000	Name RVA	
	00004ECB	00000000	Import Address Table RVA	
	00004ECC	00000000	Time Date Stamp	
	00004ECD	00000000	Forwarder Chain	
	00004ECE	00000000	Name RVA	
	00004ECF	00000000	Import Address Table RVA	
	00004ED0	00000000	Time Date Stamp	
	00004ED1	00000000	Forwarder Chain	
	00004ED2	00000000	Name RVA	
	00004ED3	00000000	Import Address Table RVA	
	00004ED4	00000000	Time Date Stamp	
	00004ED5	00000000	Forwarder Chain	
	00004ED6	00000000	Name RVA	
	00004ED7	00000000	Import Address Table RVA	
	00004ED8	00000000	Time Date Stamp	
	00004ED9	00000000	Forwarder Chain	
	00004EDA	00000000	Name RVA	
	00004EDB	00000000	Import Address Table RVA	
	00004EDC	00000000	Time Date Stamp	
	00004EDD	00000000	Forwarder Chain	
	00004EDE	00000000	Name RVA	
	00004EDF	00000000	Import Address Table RVA	
	00004EE0	00000000	Time Date Stamp	
	00004EE1	00000000	Forwarder Chain	
	00004EE2	00000000	Name RVA	
	00004EE3	00000000	Import Address Table RVA	
	00004EE4	00000000	Time Date Stamp	
	00004EE5	00000000	Forwarder Chain	
	00004EE6	00000000	Name RVA	
	00004EE7	00000000	Import Address Table RVA	
	00004EE8	00000000	Time Date Stamp	
	00004EE9	00000000	Forwarder Chain	
	00004EEA	00000000	Name RVA	
	00004EEB	00000000	Import Address Table RVA	
	00004EEC	00000000	Time Date Stamp	
	00004EED	00000000	Forwarder Chain	
	00004EEE	00000000	Name RVA	
	00004EEF	00000000	Import Address Table RVA	
	00004EF0	00000000	Time Date Stamp	
	00004EF1	00000000	Forwarder Chain	
	00004EF2	00000000	Name RVA	
	00004EF3	00000000	Import Address Table RVA	
	00004EF4	00000000	Time Date Stamp	
	00004EF5	00000000	Forwarder Chain	
	00004EF6	00000000	Name RVA	
	00004EF7	00000000	Import Address Table RVA	
	00004EF8	00000000	Time Date Stamp	
	00004EF9	00000000	Forwarder Chain	
	00004EFA	00000000	Name RVA	
	00004EFB	00000000	Import Address Table RVA	
	00004EFC	00000000	Time Date Stamp	
	00004EFD	00000000	Forwarder Chain	
	00004EFE	00000000	Name RVA	
	00004EFF	00000000	Import Address Table RVA	
	00004F00	00000000	Time Date Stamp	
	00004F01	00000000	Forwarder Chain	
	00004F02	00000000	Name RVA	
	00004F03	00000000	Import Address Table RVA	
	00004F04	00000000	Time Date Stamp	
	00004F05	00000000	Forwarder Chain	
	00004F06	00000000	Name RVA	
	00004F07	00000000	Import Address Table RVA	
	00004F08	00000000	Time Date Stamp	
	00004F09	00000000	Forwarder Chain	
	00004F0A	00000000	Name RVA	
	00004F0B	00000000	Import Address Table RVA	
	00004F0C	00000000	Time Date Stamp	
	00004F0D	00000000	Forwarder Chain	
	00004F0E	00000000	Name RVA	
	00004F0F	00000000	Import Address Table RVA	
	00004F10	00000000	Time Date Stamp	
	00004F11	00000000	Forwarder Chain	
	00004F12	00000000	Name RVA	
	00004F13	00000000	Import Address Table RVA	
	00004F14	00000000	Time Date Stamp	
	00004F15	00000000	Forwarder Chain	
	00004F16	00000000	Name RVA	
	00004F17	00000000	Import Address Table RVA	
	00004F18	00000000	Time Date Stamp	
	00004F19	00000000	Forwarder Chain	
	00004F1A	00000000	Name RVA	
	00004F1B	00000000	Import Address Table RVA	
	00004F1C	00000000	Time Date Stamp	
	00004F1D	00000000	Forwarder Chain	
	00004F1E	00000000	Name RVA	
	00004F1F	00000000	Import Address Table RVA	
	00004F20	00000000	Time Date Stamp	
	00004F21	00000000	Forwarder Chain	
	00004F22	00000000	Name RVA	
	00004F23	00000000	Import Address Table RVA	
	00004F24	00000000	Time Date Stamp	
	00004F25	00000000	Forwarder Chain	
	00004F26	00000000	Name RVA	
	00004F27	00000000	Import Address Table RVA	
	00004F28	00000000	Time Date Stamp	
	00004F29	00000000	Forwarder Chain	
	00004F2A	00000000	Name RVA	
	00004F2B	00000000	Import Address Table RVA	
	00004F2C	00000000	Time Date Stamp	
	00004F2D	00000000	Forwarder Chain	
	00004F2E	00000000	Name RVA	
	00004F2F	00000000	Import Address Table RVA	
	00004F30	00000000	Time Date Stamp	
	00004F31	00000000	Forwarder Chain	
	00004F32	00000000	Name RVA	
	00004F33	00000000	Import Address Table RVA	
	00004F34	00000000	Time Date Stamp	
	00004F35	00000000	Forwarder Chain	
	00004F36	00000000	Name RVA	
	00004F37	00000000	Import Address Table RVA	
	00004F38	00000000	Time Date Stamp	
	00004F39	00000000	Forwarder Chain	
	00004F3A	00000000	Name RVA	
	00004F3B	00000000	Import Address Table RVA	
	00004F3C	00000000	Time Date Stamp	
	00004F3D	00000000	Forwarder Chain	
	00004F3E	00000000	Name RVA	
	00004F3F	00000000	Import Address Table RVA	
	00004F40	00000000	Time Date Stamp	
	00004F41	00000000	Forwarder Chain	
	00004F42	00000000	Name RVA	
	00004F43	00000000	Import Address Table RVA	
	00004F44	00000000	Time Date Stamp	
	00004F45	00000000	Forwarder Chain	
	00004F46	00000000	Name RVA	
	00004F47	00000000	Import Address Table RVA	
	00004F48	00000000	Time Date Stamp	
	00004F49	00000000	Forwarder Chain	
	00004F4A	00000000	Name RVA	
	00004F4B	00000000	Import Address Table RVA	
	00004F4C	00000000	Time Date Stamp	
	00004F4D	00000000	Forwarder Chain	
	00004F4E	00000000	Name RVA	
	00004F4F	00000000	Import Address Table RVA	
	00004F50	00000000	Time Date Stamp	
	00004F51	00000000	Forwarder Chain	
	00004F52	00000000	Name RVA	
	00004F53	00000000	Import Address Table RVA	
	00004F54	00000000	Time Date Stamp	
	00004F55	00000000	Forwarder Chain	
	00004F56	00000000	Name RVA	
	00004F57	00000000	Import Address Table RVA	
	00004F58	00000000	Time Date Stamp	
	00004F59	00000000	Forwarder Chain	
	00004F5A	00000000	Name RVA	
	00004F5B	00000000	Import Address Table RVA	
	00004F5C	00000000	Time Date Stamp	
	00004F5D	00000000	Forwarder Chain	
	00004F5E	00000000	Name RVA	
	00004F5F	00000000	Import Address Table RVA	
	00004F60	00000000	Time Date Stamp	
	00004F61	00000000	Forwarder Chain	
	00004F62	00000000	Name RVA	
	00004F63	00000000	Import Address Table RVA	
	00004F64	00000000	Time Date Stamp	
	00004F65	00000000	Forwarder Chain	
	00004F66	00000000	Name RVA	
	00004F67	00000000	Import Address Table RVA	
	00004F68	00000000	Time Date Stamp	
	00004F69	00000000	Forwarder Chain	
	00004F6A	00000000	Name RVA	
	00004F6B	00000000	Import Address Table RVA	
	00004F6C	00000000	Time Date Stamp	
	00004F6D	00000000	Forwarder Chain	
	00004F6E	00000000	Name RVA	
	00004F6F	00000000	Import Address Table RVA	
	00004F70	00000000	Time Date Stamp	
	00004F71	00000000	Forwarder Chain	
	00004F72	00000000	Name RVA	
	00004F73	00000000	Import Address Table RVA	
	00004F74	00000000	Time Date Stamp	
	00004F75	00000000	Forwarder Chain	
	00004F76	00000000	Name RVA	
	00004F77	00000000	Import Address Table RVA	
	00004F78	00000000	Time Date Stamp	
	00004F79	00000000	Forwarder Chain	
	00004F7A	00000000	Name RVA	
	00004F7B	00000000	Import Address Table RVA	
	00004F7C	00000000	Time Date Stamp	
	00004F7D	00000000	Forwarder Chain	
	00004F7E	00000000	Name RVA	
	00004F7F	00000000	Import Address Table RVA	
	00004F80	00000000	Time Date Stamp	
	00004F81	00000000	Forwarder Chain	
	00004F82	00000000	Name RVA	
	00004F83	00000000	Import Address Table RVA	
	00004F84	00000000	Time Date Stamp	
	00004F85	00000000	Forwarder Chain	
	00004F86	00000000	Name RVA	
	00004F87	00000000	Import Address Table RVA	
	00004F88	00000000	Time Date Stamp	
	00004F89	00000000	Forwarder Chain	
	00004F8A	00000000	Name RVA	
	00004F8B	00000000	Import Address Table RVA	
	00004F8C	00000000	Time Date Stamp	
	00004F8D	00000000	Forwarder Chain	
	00004F8E	00000000	Name RVA	
	00004F8F	00000000	Import Address Table RVA	
	00004F90	00000000	Time Date Stamp	
	00004F91	00000000	Forwarder Chain	
	00004F92	00000000	Name RVA	
	00004F93	00000000	Import Address Table RVA	
	00004F94	00000000	Time Date Stamp	
	00004F95	00000000	Forwarder Chain	
	00004F96	00000000	Name RVA	
	00004F97	00000000	Import Address Table RVA	
	00004F98	00000000	Time Date Stamp	
	00004F99	00000000	Forwarder Chain	
	00004F9A	00000000	Name RVA	
	00004F9B	00000000	Import Address Table RVA	
	00004F9C	00000000	Time Date Stamp	
	00004F9D	00000000	Forwarder Chain	
	00004F9E	00000000	Name RVA	
	00004F9F	00000000	Import Address Table RVA	
	00004FA0	00000000	Time Date Stamp	
	00004FA1	00000000	Forwarder Chain	
	00004FA2	00000000	Name RVA	
	00004FA3	00000000	Import Address Table RVA	
	00004FA4	00000000	Time Date Stamp	
	00004FA5	00000000	Forwarder Chain	
	00004FA6	00000000	Name RVA	
	00004FA7	00000000	Import Address Table RVA	
	00004FA8	00000000	Time Date Stamp	
	00004FA9	00000000	Forwarder Chain	
	00004FAA	00000000	Name RVA	
	00004FAB	00000000	Import Address Table RVA	
	00004FAC	00000000	Time Date Stamp	
	00004FAD	00000000	Forwarder Chain	
	00004FAE	00000000	Name RVA	
	00004FAF	00000000	Import Address Table RVA	
	00004FB0	00000000	Time Date Stamp	
	00004FB1	00000000	Forwarder Chain	
	00004FB2	00000000	Name RVA	
	00004FB3	00000000	Import Address Table RVA	
	00004FB4	00000000	Time Date Stamp	
	00004FB5	00000000	Forwarder Chain	
	00004FB6	00000000	Name RVA	
	00004FB7	00000000	Import Address Table RVA	
	00004FB8	00000000	Time Date Stamp	
	00004FB9	00000000	Forwarder Chain	
	00004FBA			

供了一些功能，例如创建和管理用户帐户、验证用户凭据、创建和管理安全上下文、操纵 Windows 服务、管理系统注册表等。

b) MSVCRT.dll:

msvcrt.dll 是 windows 操作系统中提供的 C 语言运行库执行文件 (Microsoft Visual C Runtime Library)，其中提供了 printf, malloc, strcpy 等 C 语言库函数的具体运行实现，并且为使用 C/C++ (Vc) 编译的程序提供了初始化（如获取命令行参数）以及退出等功能。

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。

查看导入函数，发现程序可以创建进程（CreateProcessA）和线程（CreateThread）服务，创建操控服务（CreateServiceA、OpenServiceA）、操控注册表（RegCreateKeyA、RegSetValueExA、RegisterServiceCtrlHandlerA、RegOpenKeyExA、RegQueryValueExA、RegCloseKey 等），对网络（InternetOpenA、InternetReadFile、InternetConnectA、InternetCloseHandle、HttpOpenRequest、HttpSendRequestA、HttpQueryInfoA 等）进行操作等。

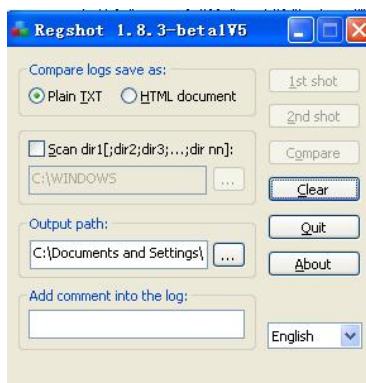
Offset	File	Ordinal	Description	Value
00004030	00000000	1047	OpenServiceA	
00004034	00000000	1078	CloseServiceHandle	
00004038	00000000	1172	RegOpenKeyExA	
0000403C	00000000	1173	RegQueryValueExA	
00004040	00000000	1168	RegCloseKey	
00004044	00000000	1145	OpenSCManagerA	
00004048	00000000	104C	CreateServiceA	
0000404C	00000000	1034	CloseServiceHandle	
00004050	00000000	115E	RegCreateKeyA	
00004054	00000000	1188	RegisterServiceCtrlHandlerA	
00004058	00000000	118E	RegisterServiceCtrlHandlerA	
0000405C	00000000	1194	RegisterServiceCtrlHandlerA	
00004060	00000000	1196	RegisterServiceCtrlHandlerA	
00004064	00000000	1198	RegisterServiceCtrlHandlerA	
00004068	00000000	119A	RegisterServiceCtrlHandlerA	
0000406C	00000000	119C	RegisterServiceCtrlHandlerA	
00004070	00000000	119E	RegisterServiceCtrlHandlerA	
00004074	00000000	11A0	RegisterServiceCtrlHandlerA	
00004078	00000000	11A2	RegisterServiceCtrlHandlerA	
0000407C	00000000	11A4	RegisterServiceCtrlHandlerA	
00004080	00000000	11A6	RegisterServiceCtrlHandlerA	
00004084	00000000	11A8	RegisterServiceCtrlHandlerA	
00004088	00000000	11AA	RegisterServiceCtrlHandlerA	
0000408C	00000000	11AC	RegisterServiceCtrlHandlerA	
00004090	00000000	11AE	RegisterServiceCtrlHandlerA	
00004094	00000000	11B0	RegisterServiceCtrlHandlerA	
00004098	00000000	11B2	RegisterServiceCtrlHandlerA	
0000409C	00000000	11B4	RegisterServiceCtrlHandlerA	
000040A0	00000000	11B6	RegisterServiceCtrlHandlerA	
000040A4	00000000	11B8	RegisterServiceCtrlHandlerA	
000040A8	00000000	11BA	RegisterServiceCtrlHandlerA	
000040AC	00000000	11BC	RegisterServiceCtrlHandlerA	
000040B0	00000000	11BE	RegisterServiceCtrlHandlerA	
000040B4	00000000	11C0	RegisterServiceCtrlHandlerA	
000040B8	00000000	11C2	RegisterServiceCtrlHandlerA	
000040BC	00000000	11C4	RegisterServiceCtrlHandlerA	
000040C0	00000000	11C6	RegisterServiceCtrlHandlerA	
000040C4	00000000	11C8	RegisterServiceCtrlHandlerA	
000040C8	00000000	11CA	RegisterServiceCtrlHandlerA	
000040CC	00000000	11CC	RegisterServiceCtrlHandlerA	
000040D0	00000000	11CE	RegisterServiceCtrlHandlerA	
000040D4	00000000	11D0	RegisterServiceCtrlHandlerA	
000040D8	00000000	11D2	RegisterServiceCtrlHandlerA	
000040DC	00000000	11D4	RegisterServiceCtrlHandlerA	
000040E0	00000000	11D6	RegisterServiceCtrlHandlerA	
000040E4	00000000	11D8	RegisterServiceCtrlHandlerA	
000040E8	00000000	11DA	RegisterServiceCtrlHandlerA	
000040EC	00000000	11DC	RegisterServiceCtrlHandlerA	
000040F0	00000000	11DE	RegisterServiceCtrlHandlerA	
000040F4	00000000	11E0	RegisterServiceCtrlHandlerA	
000040F8	00000000	11E2	RegisterServiceCtrlHandlerA	
000040FC	00000000	11E4	RegisterServiceCtrlHandlerA	
00004100	00000000	11E6	RegisterServiceCtrlHandlerA	
00004104	00000000	11E8	RegisterServiceCtrlHandlerA	
00004108	00000000	11EA	RegisterServiceCtrlHandlerA	
0000410C	00000000	11EC	RegisterServiceCtrlHandlerA	
00004110	00000000	11EE	RegisterServiceCtrlHandlerA	
00004114	00000000	11F0	RegisterServiceCtrlHandlerA	
00004118	00000000	11F2	RegisterServiceCtrlHandlerA	
0000411C	00000000	11F4	RegisterServiceCtrlHandlerA	
00004120	00000000	11F6	RegisterServiceCtrlHandlerA	
00004124	00000000	11F8	RegisterServiceCtrlHandlerA	
00004128	00000000	11FA	RegisterServiceCtrlHandlerA	
0000412C	00000000	11FC	RegisterServiceCtrlHandlerA	
00004130	00000000	11FE	RegisterServiceCtrlHandlerA	
00004134	00000000	1200	RegisterServiceCtrlHandlerA	
00004138	00000000	1202	RegisterServiceCtrlHandlerA	
0000413C	00000000	1204	RegisterServiceCtrlHandlerA	
00004140	00000000	1206	RegisterServiceCtrlHandlerA	
00004144	00000000	1208	RegisterServiceCtrlHandlerA	
00004148	00000000	120A	RegisterServiceCtrlHandlerA	
0000414C	00000000	120C	RegisterServiceCtrlHandlerA	
00004150	00000000	120E	RegisterServiceCtrlHandlerA	
00004154	00000000	1210	RegisterServiceCtrlHandlerA	
00004158	00000000	1212	RegisterServiceCtrlHandlerA	
0000415C	00000000	1214	RegisterServiceCtrlHandlerA	
00004160	00000000	1216	RegisterServiceCtrlHandlerA	
00004164	00000000	1218	RegisterServiceCtrlHandlerA	
00004168	00000000	121A	RegisterServiceCtrlHandlerA	
0000416C	00000000	121C	RegisterServiceCtrlHandlerA	
00004170	00000000	121E	RegisterServiceCtrlHandlerA	
00004174	00000000	1220	RegisterServiceCtrlHandlerA	
00004178	00000000	1222	RegisterServiceCtrlHandlerA	
0000417C	00000000	1224	RegisterServiceCtrlHandlerA	
00004180	00000000	1226	RegisterServiceCtrlHandlerA	
00004184	00000000	1228	RegisterServiceCtrlHandlerA	
00004188	00000000	122A	RegisterServiceCtrlHandlerA	
0000418C	00000000	122C	RegisterServiceCtrlHandlerA	
00004190	00000000	122E	RegisterServiceCtrlHandlerA	
00004194	00000000	1230	RegisterServiceCtrlHandlerA	
00004198	00000000	1232	RegisterServiceCtrlHandlerA	
0000419C	00000000	1234	RegisterServiceCtrlHandlerA	
000041A0	00000000	1236	RegisterServiceCtrlHandlerA	
000041A4	00000000	1238	RegisterServiceCtrlHandlerA	
000041A8	00000000	123A	RegisterServiceCtrlHandlerA	
000041AC	00000000	123C	RegisterServiceCtrlHandlerA	
000041B0	00000000	123E	RegisterServiceCtrlHandlerA	
000041B4	00000000	1240	RegisterServiceCtrlHandlerA	
000041B8	00000000	1242	RegisterServiceCtrlHandlerA	
000041BC	00000000	1244	RegisterServiceCtrlHandlerA	
000041C0	00000000	1246	RegisterServiceCtrlHandlerA	
000041C4	00000000	1248	RegisterServiceCtrlHandlerA	
000041C8	00000000	124A	RegisterServiceCtrlHandlerA	
000041CC	00000000	124C	RegisterServiceCtrlHandlerA	
000041D0	00000000	124E	RegisterServiceCtrlHandlerA	
000041D4	00000000	1250	RegisterServiceCtrlHandlerA	
000041D8	00000000	1252	RegisterServiceCtrlHandlerA	
000041DC	00000000	1254	RegisterServiceCtrlHandlerA	
000041E0	00000000	1256	RegisterServiceCtrlHandlerA	
000041E4	00000000	1258	RegisterServiceCtrlHandlerA	
000041E8	00000000	125A	RegisterServiceCtrlHandlerA	
000041EC	00000000	125C	RegisterServiceCtrlHandlerA	
000041F0	00000000	125E	RegisterServiceCtrlHandlerA	
000041F4	00000000	1260	RegisterServiceCtrlHandlerA	
000041F8	00000000	1262	RegisterServiceCtrlHandlerA	
000041FC	00000000	1264	RegisterServiceCtrlHandlerA	
00004200	00000000	1266	RegisterServiceCtrlHandlerA	
00004204	00000000	1268	RegisterServiceCtrlHandlerA	
00004208	00000000	126A	RegisterServiceCtrlHandlerA	
0000420C	00000000	126C	RegisterServiceCtrlHandlerA	
00004210	00000000	126E	RegisterServiceCtrlHandlerA	
00004214	00000000	1270	RegisterServiceCtrlHandlerA	
00004218	00000000	1272	RegisterServiceCtrlHandlerA	
0000421C	00000000	1274	RegisterServiceCtrlHandlerA	
00004220	00000000	1276	RegisterServiceCtrlHandlerA	
00004224	00000000	1278	RegisterServiceCtrlHandlerA	
00004228	00000000	127A	RegisterServiceCtrlHandlerA	
0000422C	00000000	127C	RegisterServiceCtrlHandlerA	
00004230	00000000	127E	RegisterServiceCtrlHandlerA	
00004234	00000000	1280	RegisterServiceCtrlHandlerA	
00004238	00000000	1282	RegisterServiceCtrlHandlerA	
0000423C	00000000	1284	RegisterServiceCtrlHandlerA	
00004240	00000000	1286	RegisterServiceCtrlHandlerA	
00004244	00000000	1288	RegisterServiceCtrlHandlerA	
00004248	00000000	128A	RegisterServiceCtrlHandlerA	
0000424C	00000000	128C	RegisterServiceCtrlHandlerA	
00004250	00000000	128E	RegisterServiceCtrlHandlerA	
00004254	00000000	1290	RegisterServiceCtrlHandlerA	
00004258	00000000	1292	RegisterServiceCtrlHandlerA	
0000425C	00000000	1294	RegisterServiceCtrlHandlerA	
00004260	00000000	1296	RegisterServiceCtrlHandlerA	
00004264	00000000	1298	RegisterServiceCtrlHandlerA	
00004268	00000000	129A	RegisterServiceCtrlHandlerA	
0000426C	00000000	129C	RegisterServiceCtrlHandlerA	
00004270	00000000	129E	RegisterServiceCtrlHandlerA	
00004274	00000000	12A0	RegisterServiceCtrlHandlerA	
00004278	00000000	12A2	RegisterServiceCtrlHandlerA	
0000427C	00000000	12A4	RegisterServiceCtrlHandlerA	
00004280	00000000	12A6	RegisterServiceCtrlHandlerA	
00004284	00000000	12A8	RegisterServiceCtrlHandlerA	
00004288	00000000	12AA	RegisterServiceCtrlHandlerA	
0000428C	00000000	12AC	RegisterServiceCtrlHandlerA	
00004290	00000000	12AE	RegisterServiceCtrlHandlerA	
00004294	00000000	12B0	RegisterServiceCtrlHandlerA	
00004298	00000000	12B2	RegisterServiceCtrlHandlerA	
0000429C	00000000	12B4	RegisterServiceCtrlHandlerA	
000042A0	00000000	12B6	RegisterServiceCtrlHandlerA	
000042A4	00000000	12B8	RegisterServiceCtrlHandlerA	
000042A8	00000000	12BA	RegisterServiceCtrlHandlerA	
000042AC	00000000	12BC	RegisterServiceCtrlHandlerA	
000042B0	00000000	12BE	RegisterServiceCtrlHandlerA	
000042B4	00000000	12C0	RegisterServiceCtrlHandlerA	
000042B8	00000000	12C2	RegisterServiceCtrlHandlerA	
000042BC	00000000	12C4	RegisterServiceCtrlHandlerA	
000042C0	00000000	12C6	RegisterServiceCtrlHandlerA	
000042C4	00000000	12C8	RegisterServiceCtrlHandlerA	
000042C8	00000000	12CA	RegisterServiceCtrlHandlerA	
000042CC	00000000	12CC	RegisterServiceCtrlHandlerA	
000042D0	00000000	12CE	RegisterServiceCtrlHandlerA	
000042D4	00000000	12D0	RegisterServiceCtrlHandlerA	
000042D8	00000000	12D2	RegisterServiceCtrlHandlerA	
000042DC	00000000	12D4	RegisterServiceCtrlHandlerA	
000042E0	00000000	12D6	RegisterServiceCtrlHandlerA	
000042E4	00000000	12D8	RegisterServiceCtrlHandlerA	
000042E8	00000000	12DA	RegisterServiceCtrlHandlerA	
000042EC	00000000	12DC	RegisterServiceCtrlHandlerA	
000042F0	00000000	12DE	RegisterServiceCtrlHandlerA	
000042F4	00000000	12E0	RegisterServiceCtrlHandlerA	
000042F8	00000000	12E2	RegisterServiceCtrlHandlerA	
000042FC	00000000	12E4	RegisterServiceCtrlHandlerA	
00004300	00000000	12E6	RegisterServiceCtrlHandlerA	
00004304	00000000	12E8	RegisterServiceCtrlHandlerA	
00004308	00000000	12EA	RegisterServiceCtrlHandlerA	
0000430C	00000000	12EC	RegisterServiceCtrlHandlerA	
00004310	00000000	12EE	RegisterServiceCtrlHandlerA	
00004314	00000000	12F0	RegisterServiceCtrlHandlerA	
00004318	00000000	12F2	RegisterServiceCtrlHandlerA	
0000431C	00000000	12F4	RegisterServiceCtrlHandlerA	
00004320	00000000	12F6	RegisterServiceCtrlHandlerA	
00004324	00000000	12F8	RegisterServiceCtrlHandlerA	
00004328	00000000	12FA	RegisterServiceCtrlHandlerA	
0000432C	00000000	12FC	RegisterServiceCtrlHandlerA	
00004330	00000000	12FE	RegisterServiceCtrlHandlerA	
00004334	00000000	1300	RegisterServiceCtrlHandlerA	
00004338	00000000	1302	RegisterServiceCtrlHandlerA	
0000433C	00000000	1304	RegisterServiceCtrlHandlerA	
00004340	00000000	1306	RegisterServiceCtrlHandlerA	
00004344	00000000	1308	RegisterServiceCtrlHandlerA	
00004348	00000000	130A	RegisterServiceCtrlHandlerA	
0000434C	00000000	130C	RegisterServiceCtrlHandlerA	
00004350	00000000	130E	RegisterServiceCtrlHandlerA	
00004354	00000000	1310	RegisterServiceCtrlHandlerA	
00004358	00000000	1312	RegisterServiceCtrlHandlerA	
0000435C	00000000	1314	RegisterServiceCtrlHandlerA	
00004360	00000000	1316	RegisterServiceCtrlHandlerA	
00004364	00000000	1318	RegisterServiceCtrlHandlerA	
00004368	00000000	131A	RegisterServiceCtrlHandlerA	
0000436C	00000000	131C	RegisterServiceCtrlHandlerA	
00004370	00000000	131E	RegisterServiceCtrlHandlerA	
00004374	00000000	1320	RegisterServiceCtrlHandlerA	
00004378	00000000	1322	RegisterServiceCtrlHandlerA	
0000437C	00000000	1324	RegisterServiceCtrlHandlerA	
00004380	00000000	1326	RegisterServiceCtrlHandlerA	
00004384	00000000	1328	RegisterServiceCtrlHandlerA	
00004388	00000000	132A	RegisterServiceCtrlHandlerA	
0000438C	00000000	132C	RegisterServiceCtrlHandlerA	
00004390	00000000	132E	RegisterServiceCtrlHandlerA	
00004394	00000000	1330	RegisterServiceCtrlHandlerA	
00004398	00000000	1332	RegisterServiceCtrlHandlerA	
0000439C				

文件。推测这个恶意代码需要使用导出函数 `installA` 将自身注册为恶意代码分析实战一个服务，我们将使用这个函数来试图安装恶意代码。

根据上述分析可知，可以利用 `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 `Process Explorer` 监控系统上运行的进程。在使用命令 `rundll32.exe Lab03-02.dll,installA` 运行 `Lab03-02.dll` 文件前后，分别用 `regshot` 记录一下注册表信息，点击 `compare` 对比运行 `Lab03-02.dll` 的变化。



观察编译后的 `.txt` 文件，在 `Keys added` 一节中，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；看到 `ImagePath` 被设置为 `svchost.exe` 意味着这个恶意代码将会在一个 `svchost.exe` 进程中启动。

```
-----  
HKLM\SYSTEM\ControlSet001\Services\IPRIP  
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Parameters  
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Security  
HKLM\SYSTEM\CurrentControlSet\Services\IPRIP  
HKLM\SYSTEM\CurrentControlSet\Services\IPRIP\Parameters  
HKLM\SYSTEM\CurrentControlSet\Services\IPRIP\Security
```

可以看到恶意代码将为自身安装 `IPRIP` 服务。使用 `net start IPRIP` 启动服务，就可以成功运行了。

```
C:\Documents and Settings\lulu\桌面\Practical Malware Analysis Labs\BinaryCollection\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。
```

(3) 你怎么能找到这个恶意代码是哪个进程下运行的？

l:%CurrentDirectory%\Lab03-02.dll, 在 Process Explorer 中搜索 DLL 即可查看到 Intranet Network Awareness 这一服务。

如果将 Lab03-02.dll 重命名为其他文件名, 如 malware.dll, 然后这个恶意代码就会把 malware.dll 写入到注册表项中。

综上所述, 创建 IPRIIP 服务可以是这个代码在主机上的感染迹象特征。

(6) 这个恶意代码是否存在一些有用的网络特征码?

查看网络情况, 发现了 1088 的端口的网络迹象。

对 Wireshark 的结果进行进一步分析

Time	Source	Destination	Protocol	Length	Info
32.8944540	192.168.175.137	192.168.175.2	DNS	88	Standard query 0xe779 A practicalmalwareanalysis.com

反映了该程序申请解析域名 practicalmalwareanalysis.com, 通过 80 端口连接到这台主机, 使用 HTTP 协议, 做一个 GET 请求 serve.html, 可以作为网络上的一种判断迹象。

(三) Lab3-3

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?

首先使用 Process Explorer 和 procmon 工具。使用 File→Capture Events 将事件捕获先关闭, 直到所有动态分析程序都就位, 准备好执行恶意代码的时候, 再打开开关。用 Filter→Filter 呼出过滤器对话框, 然后按一下 Reset 按钮, 确保只有默认过滤器被启用。

从命令提示符或双击图标运行 Lab03-03.exe, 在 Process Explorer 中观察进程变化。

执行恶意代码, 观察 Process Explorer 发现, Lab03-03.exe 刚开始运行几秒钟便退出了, 并创建了一个子进程 svchost.exe。过滤 procmon 的结果, 分析 Lab03-03.exe 的行为, 发现它可以创建进程、加载映像、创建文件、关闭文件等, 通过进一步对 operation 的过滤, 我们可以发现它确实创建了一个子进程 svchost.exe, PID 为 2768。

注意到存在大量的 svchost 进程, 但瞬间会变为红色, 也就是创建子进程 svchost.exe 之后便自动退出, 将 svchost.exe 进程继续作为一个“孤儿”进程执行。svchost.exe 作为“孤儿”进程是极不寻常而且是高度可疑的。

Lab03-03.exe	3.13	280 K	1,068 K	2760
svchost.exe		1,008 K	2,600 K	2768 Generic Host Process ... Microsoft Corporation

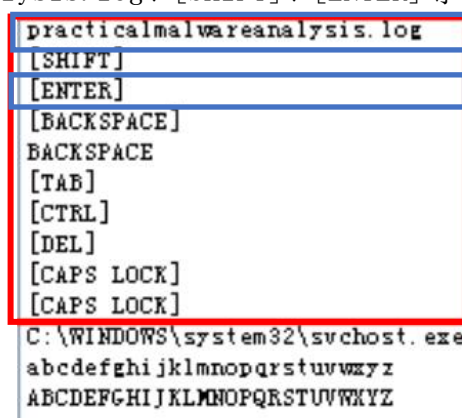
(2) 你可以找出任何的内存修改行为嘛？

对 svchost.exe “孤儿”进程右击选择 Properties 进行进一步分析，查看它的特征。svchost.exe 文件大小只有 14KB，但在 process explorer 中大小远大于 14KB。



进程看起来像是一个 PID 为 2768 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。

svchost.exe 在 Image 和 Memory 上是非常不同的，Memory 拥有 practicalmalwareanalysis.log、[SHIFT]、[ENTER]等字符，而 Image 中没有。



(Memory 镜像)

```
System\CurrentControlSet\Services
nServiceMain
ServiceDll
EPEPE
ServiceDllUnloadOnStop
uWM
jVE
eventlog
```

(Image 镜像)

从这个相同的属性页面中，选择 Strings 同时显示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。

观察 Memory 镜像框中标出的内容。最上方是一个写入的文件名，然后是一系列键盘符。分析可知，这个程序在尝试创建 practicalmalwareanalysis.log 文件，这就是尝试修改内存的行为。Memory 镜像中蓝框的内容一般都不会在磁盘镜像中一个典型的 svchost.exe 文件中出现。磁盘中镜像，确实也没有这样的指令，所以可以判定是恶意代码修改的。



(3) 这个恶意代码在主机上的感染迹象特征是什么？

该恶意软件在 Windows\System32 目录下创建 svchost.exe，且在 Lab03-03.exe 的同一目录下创建了一个 practicalmalwareanalysis.log 日志文件。

practicalmalwareanalysis.log 字符串的存在，再加上出现了[ENTER]、[CAPS LOCK]、[BACKSPACE]等字符串，表明这个程序很可能是一个击键记录器。使用进程 svchost.exe 的在可 Process Explorer 中发现的 PID(2768)在 procmon 工具中创建一个过滤器，并使用过滤器进行过滤，只显示从 PID(2768)进程的事件。

svchost.exe	2768	CreateFile	C:\Documents and Settings\lulu...	SUCCESS	Desired Acces...
svchost.exe	2768	QueryStand...	C:\Documents and Settings\lulu...	SUCCESS	AllocationSiz...
svchost.exe	2768	WriteFile	C:\Documents and Settings\lulu...	SUCCESS	Offset: 0, Le...
svchost.exe	2768	WriteFile	C:\Documents and Settings\lulu...	SUCCESS	Offset: 12, L...
svchost.exe	2768	WriteFile	C:\Documents and Settings\lulu...	SUCCESS	Offset: 34, L...
svchost.exe	2768	WriteFile	C:\Documents and Settings\lulu...	SUCCESS	Offset: 38, L...
svchost.exe	2768	CloseFile	C:\Documents and Settings\lulu...	SUCCESS	
svchost.exe	2768	CreateFile	C:\Documents and Settings\lulu...	SUCCESS	Desired Acces...
svchost.exe	2768	QueryStand...	C:\Documents and Settings\lulu...	SUCCESS	AllocationSiz...
svchost.exe	2768	WriteFile	C:\Documents and Settings\lulu...	SUCCESS	Offset: 39, L...
svchost.exe	2768	CloseFile	C:\Documents and Settings\lulu...	SUCCESS	
svchost.exe	2768	CreateFile	C:\Documents and Settings\lulu...	SUCCESS	Desired Acces...
svchost.exe	2768	QueryStand...	C:\Documents and Settings\lulu...	SUCCESS	AllocationSiz...
svchost.exe	2768	WriteFile	C:\Documents and Settings\lulu...	SUCCESS	Offset: 40, L...
svchost.exe	2768	CloseFile	C:\Documents and Settings\lulu...	SUCCESS	
svchost.exe	2768	CreateFile	C:\Documents and Settings\lulu...	SUCCESS	Desired Acces...
svchost.exe	2768	QueryStand...	C:\Documents and Settings\lulu...	SUCCESS	AllocationSiz...
svchost.exe	2768	WriteFile	C:\Documents and Settings\lulu...	SUCCESS	Offset: 41, L...
svchost.exe	2768	CloseFile	C:\Documents and Settings\lulu...	SUCCESS	
svchost.exe	2768	CreateFile	C:\Documents and Settings\lulu...	SUCCESS	Desired Acces...
svchost.exe	2768	QueryStand...	C:\Documents and Settings\lulu...	SUCCESS	AllocationSiz...
svchost.exe	2768	WriteFile	C:\Documents and Settings\lulu...	SUCCESS	Offset: 42, L...
svchost.exe	2768	CloseFile	C:\Documents and Settings\lulu...	SUCCESS	

发现 swchostexe 的 CreateFile 和 WriteFile 事件在频繁的写一个名为 practicalmalwareanalysis.log 的文件。它在不断地尝试对自己创建的日志文件 practicalmalwareanalysis.log 进行编辑，这可以作为感染的迹象特征。

(4) 这个恶意代码的目的是什么？

用记事本打开 practicalmalwareanalysis.log, 便显示出了刚刚在记事本中输入的击键记录。



这个程序首先创建了一个 `svchost.exe` 进程，然后就自动退出，可以认为这是尝试在伪装。而其创建的进程试图对原有的进程 `svchost.exe` 进行替换，实现一个击键记录器。创建 `practicalmalwareanalysis.log` 文件，内存映像中的字符串信息中会存在大量键盘指令，该文件是为了记录用户对键盘的操作。

(四) Lab3-4

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。

(这个程序还会在第 9 章的实验作业中进一步分析)

(1) 当你运行这个文件时，会发生什么呢？

当尝试双击打开或者是命令行运行该程序时，发现该程序会打开一个命令行窗口，迅速关掉，最后该程序都会进行自我删除。

在 `Process Explorer` 中可以看到，快速运行了 `Lab03-04.exe`，打开了 `cmd.exe`，然后进程自己消失。

(2) 是什么原因造成动态分析无法有效实施？

使用 `strings Lab03-04.exe` 查看字符串，进行静态分析

```
cmd.exe
>> NUL
/c del
ups
http://www.practicalmalwareanalysis.com
Manager Service
.exe
%SYSTEMROOT%\system32\
k:%s h:%s p:%s per:%s
```

观察到 `cmd` 命令和 `https://www.practicalmalwareanalysis.com` 网址，还看到了 `/c del` 删除操作。

```
LZ@
Configuration
SOFTWARE\Microsoft \XPS
\kernel32.dll
HTTP/1.0
GET
XXXX
XXXX
```

看到域名、注册表位置，如字符串 SOFTWARE\Microsoft \XPS;DOWNLOAD、UPLOAD 这样的命令字符串，以及 HTTP/1.0 字符串等。推测可能是在命令行中运行某些指令，来修改注册表、联网等。这些表明恶意代码可能是一个 HTTP 后门程序。

```
Manager Service
.exe
%SYSTEMROOT%\system32\
k:%s h:%s p:%s per:%s
-cc
-re
-in
@@@
      CCCCC      H
`y!
@~€
```

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)。

```
.cmd
@@
GetLastActivePopup
GetActiveWindow
MessageBoxA
user32.dll
PATH
\0'@
@@
CloseHandle
SetFileTime
GetFileTime
CreateFileA
GetSystemDirectoryA
GetLastError
ReadFile
WriteFile
Sleep
GetShortPathNameA
GetModuleFileNameA
CopyFileA
ExpandEnvironmentStringsA
DeleteFileA
USER32.dll
RegQueryValueExA
RegOpenKeyExA
RegSetValueExA
RegCreateKeyExA
RegDeleteValueA
CreateServiceA
RegCreateKeyExA
RegDeleteValueA
CreateServiceA
CloseServiceHandle
ChangeServiceConfigA
OpenServiceA
OpenSCManagerA
DeleteService
ADVAPI32.dll
ShellExecuteA
SHELL32.dll
WS2_32.dll
ExitProcess
TerminateProcess
GetCurrentProcess
GetTimeZoneInformation
GetSystemTime
GetLocalTime
DuplicateHandle
GetCommandLineA
GetVersion
SetStdHandle
GetFileType
GetHandleCount
GetStdHandle
GetStartupInfoA
CreatePipe
GetExitCodeProcess
WaitForSingleObject
HeapReAlloc

GetOSError
UnhandledExceptionFilter
FreeEnvironmentStringsA
FreeEnvironmentStringsW
WideCharToMultiByte
GetEnvironmentStrings
GetEnvironmentStringsW
GetModuleHandleA
GetEnvironmentVariableA
GetVersionExA
HeapDestroy
HeapCreate
VirtualFree
HeapFree
RtlUnwind
MultiByteToWideChar
GetStringTypeA
GetStringTypeW
SetFilePointer
VirtualAlloc
LChapStringA
LChapStringW
GetProcAddress
LoadLibraryA
FlushFileBuffers
GetFileAttributesA
CreateProcessA
CompareStringA
CompareStringW
SetEnvironmentVariableA
```

观察导入函数可知，很多函数与之前的相同，但然没有如 WriteFile、RegSetValue 之类的事件。但是发现了一个 CreateProcessA 及与其相关的条目。分析可知，该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。

综上所述，猜测动态分析无法有效实施的原因可能是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？

尝试使用在字符串列表中显示的一些命令行参数，比如-in、-re、-cc，但这样做却没有得到有效的结果，程序还是会删除自身，需要更深入的分析。

四、实验结论及心得体会

(1) 实验结论

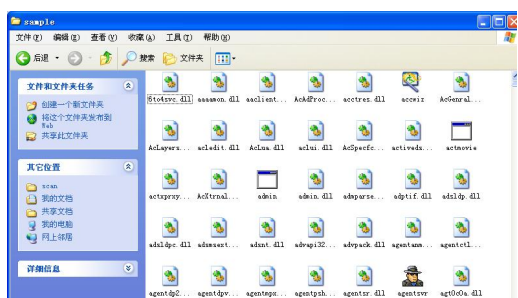
- 1) 恶意代码可以通过修改注册表、创建自启动项、创建互斥量等方式在主机上感染迹象特征。
- 2) 恶意代码可能会通过访问特定的域名或 URL 来与远程服务器通信，这可以作为网络特征码。
- 3) 恶意代码可以创建服务或进程，以隐藏自身并运行在系统中。
- 4) 恶意代码可以修改内存中的内容，如键盘记录器可以记录用户的击键信息。
- 5) 有些恶意代码可能会自我删除以避免被检测。

(2) 心得体会

通过本次实验了解了恶意代码分析和感染迹象特征的识别方法。动态分析是一种有力的工具，可以帮助我们理解恶意代码的行为和特征，从而加强网络安全防护和恶意代码防治技术。实验中的工具和技术对于安全专业人员来说是非常重要的，能够帮助我们更好地应对潜在的安全威胁。

(3) 写 yara 规则进行扫描

使用 scan.py 将所有 PE 结构的文件保存到 scan.py 文件所在目录的 sample 文件夹中。



运行结果：

```
C:\Documents and Settings\lulu\桌面\scan>python Lab3.py
样本文件夹中的文件数量：2514
匹配的文件数量：1881
扫描时间：22.42 秒
```

Lab03. py：

import os
import yara
import time
定义 YARA 规则
rules = yara.compile("Lab03.yara")
扫描的样本文件夹路径
sample_folder = "sample"
初始化变量
matched_files = 0
start_time = time.time()
遍历样本文件夹中的文件
for root, dirs, files in os.walk(sample_folder):
for file_name in files:
file_path = os.path.join(root, file_name)
try:
使用 YARA 规则扫描文件
matches = rules.match(file_path)
如果有匹配的规则，记录匹配的文件数量
if matches:
matched_files += 1
except Exception as e:
处理可能的异常
pass # 这里可以加入处理异常的代码
计算扫描时间

end_time = time.time()
scan_time = end_time - start_time
输出结果
print("样本文件夹中的文件数量: {}".format(len(files)))
print("匹配的文件数量: {}".format(matched_files))
print("扫描时间: {:.2f} 秒".format(scan_time))

Lab03. yara

rule Detect_Lab03 {
meta:
description = "YARA rule to detect Lab03 malicious code"
strings:
\$url = "www.practicalmalwareanalysis.com"
\$registry_key = "HKEY_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\VideoDriver"
\$mutex = "WinVMX32"
\$file_creation = "C:\WINDOWS\system32\vmx32to64.exe"
\$export_function = "installA"
\$svchost_exe = "svchost.exe"
\$practicalmalwareanalysis_log = "practicalmalwareanalysis.log"
\$keyboard_strings = "\[CTRL\]\[ENTER\]\[BACKSPACE\]\[TAB\]\[DEL\]\[CAPS LOCK\]"
\$cmd = "cmd.exe"
\$del = "/c del"
\$http_string = "HTTP/1.0"
\$command_strings = "/DOWNLOAD UPLOAD/"
\$regedit = "SOFTWARE\Microsoft \XPS"
condition:
any of (\$url, \$registry_key, \$mutex, \$file_creation, \$export_function, \$svchost_exe, \$practicalmalwareanalysis_log, \$keyboard_strings, \$cmd, \$del, \$http_string, \$command_strings, \$regedit)
}