

# 恶意代码分析与防治技术实验报告

## Lab 1

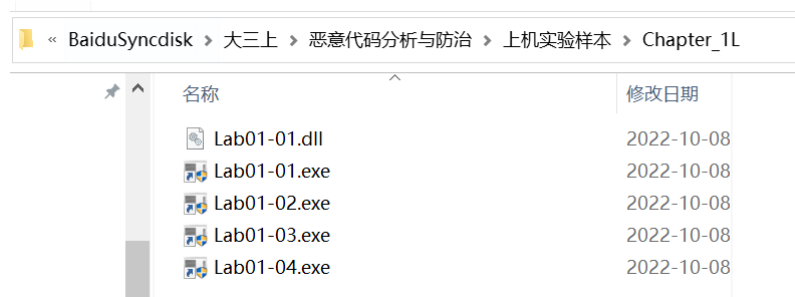
学号： 姓名： 专业：信息安全

### 一. 实验环境

1. 已关闭病毒防护的 Windows10
2. VMware+Windows XP （由于Windows上使用Dependency Walker出现卡顿情况，因此在虚拟机 Windows XP环境下使用该软件）

### 二. 实验工具

1. 实验样本



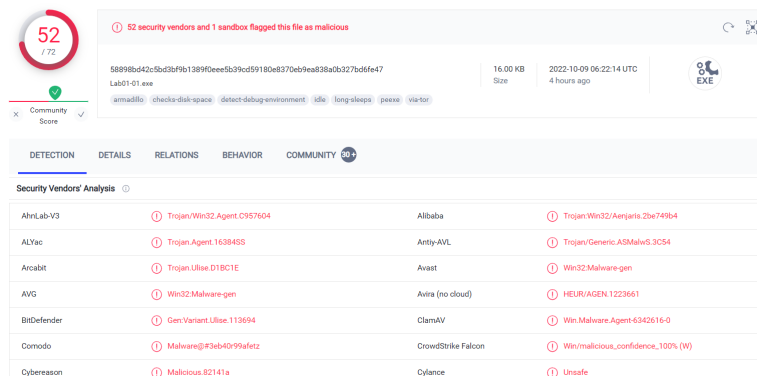
2. PView、PEiD、Dependency Walker、Resource Hacker、Strings、UPX

### 三. 实验过程

#### Lab1-1

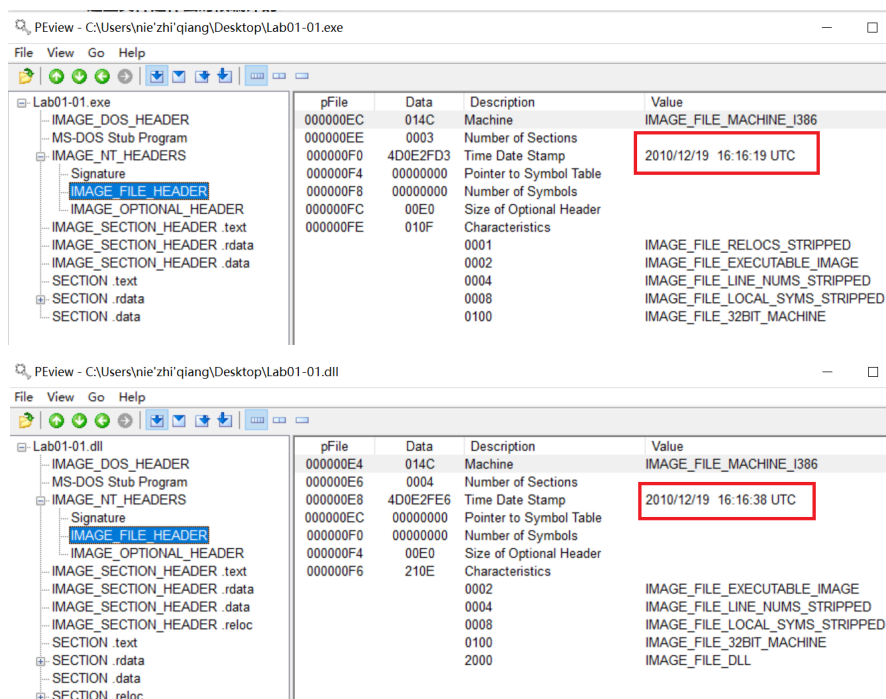
1. 将文件上传至 <https://www.virustotal.com/> 进行分析并查看报告。文件匹配到了已有的反病毒软件特征吗？

- 匹配到了
- 当把Lab01-01.exe上传到<https://www.virustotal.com/>，结果如下图所示，可以观察到这个文件匹配到52个反病毒引擎



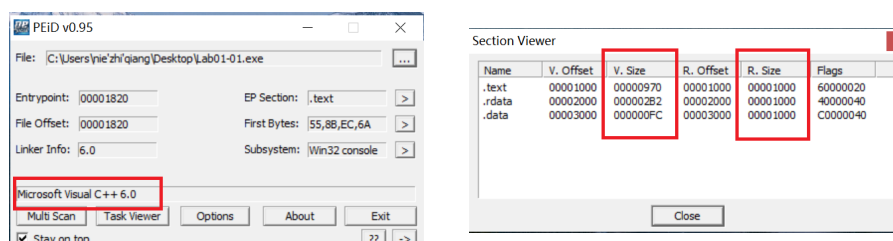
## 2. 这些文件是什么时候编译的？

- 使用PEview打开Lab01-01.exe和Lab01-01.dll，通过IMAGE\_NT\_HEADERS -> IMAGE\_FILE\_HEADER 发现报错（无法看到所有信息），从github上找到相应的补丁进行更后解决该问题，查看到编译时间均为2010-12-19，且相差在1min内。编译时间非常接近说明是同一作者在同一时间创建了这些文件，这个.exe 很有可能是用来是使用或安装.dll 文件的，因为DLL动态链接库文件无法运行自己。

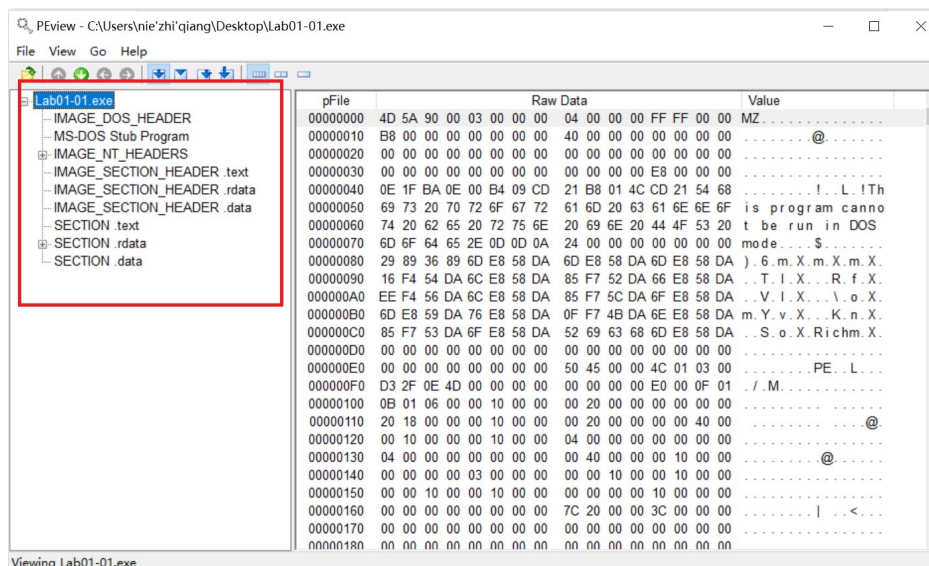


## 3. 这两个文件中是否存在迹象说明它们是否被加壳或混淆了？如果是，这些迹象在哪里？

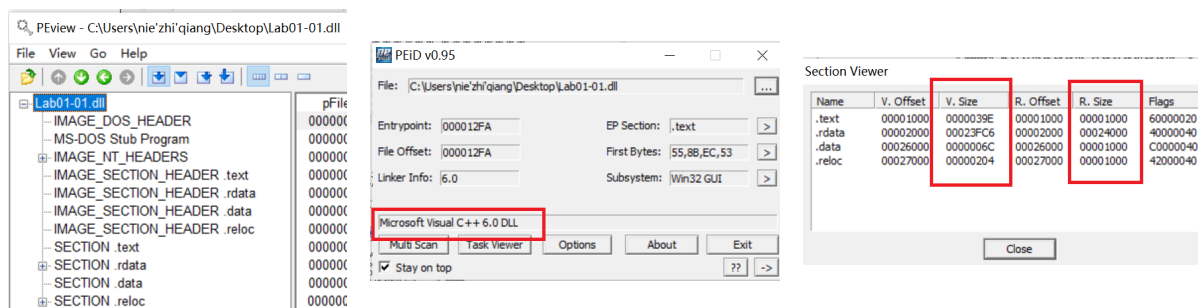
- 将Lab01-01.exe 导入PEiD后如下图所示，并没有出现加壳信息，可以观察到该文件由 Microsoft Visual C++ 6.0编译，并且文件中的分节显示，虚拟大小并没有出现比原始数据大很多的情况。



- 将Lab01-01.exe 导入 PEview后结果如下图所示，观察到PE头部中有着适当大小良好组织的文件节，虽然只有少量导出程序，说明它们可能只是一些小程序

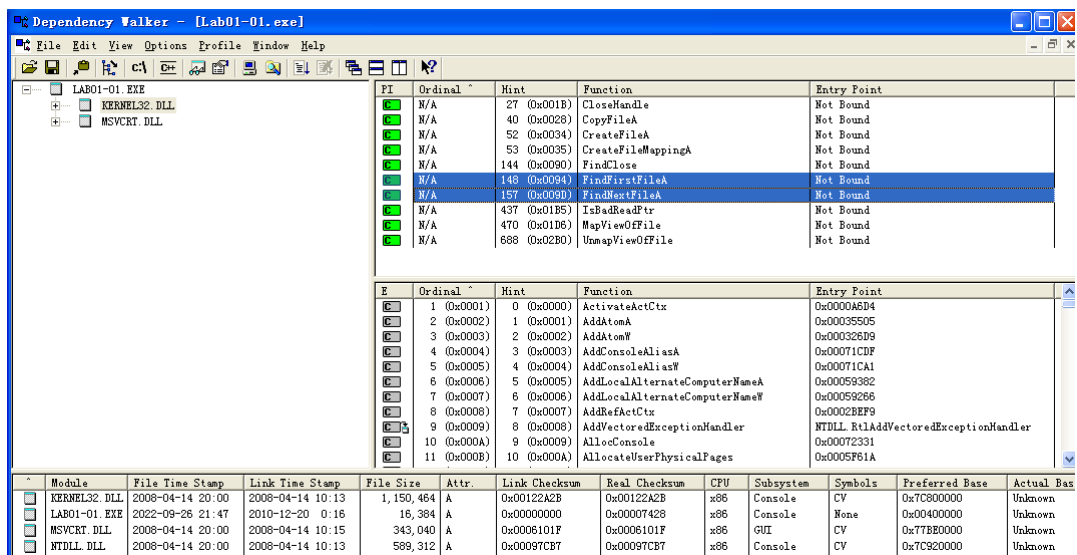


- Lab01-01.dll同理，由此可知，两个文件都没有被加壳或者混淆的迹象。

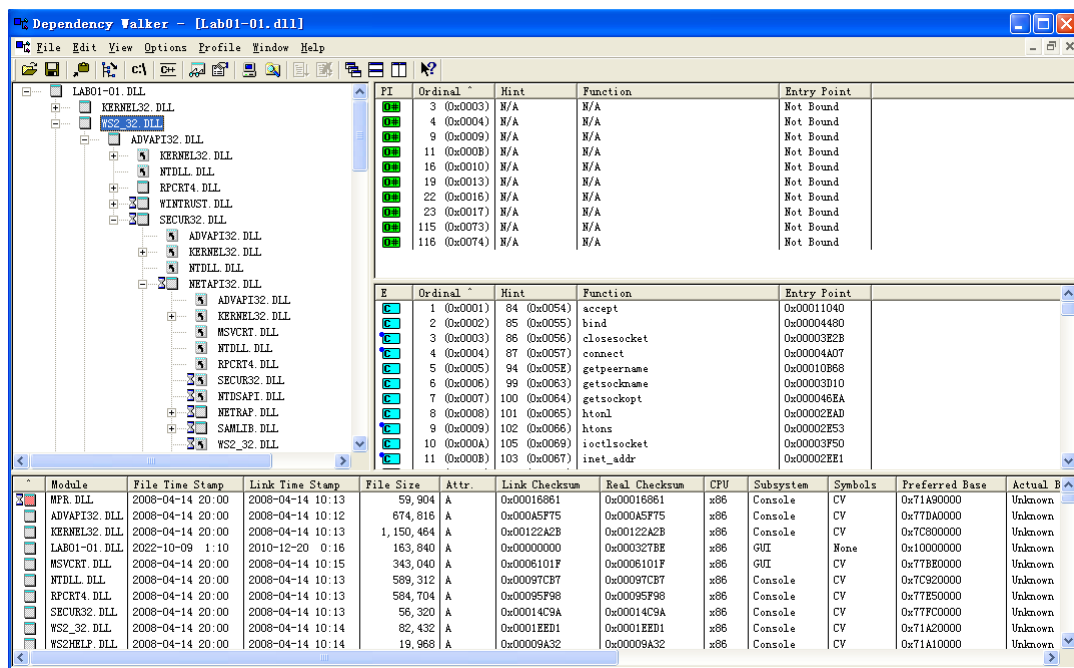
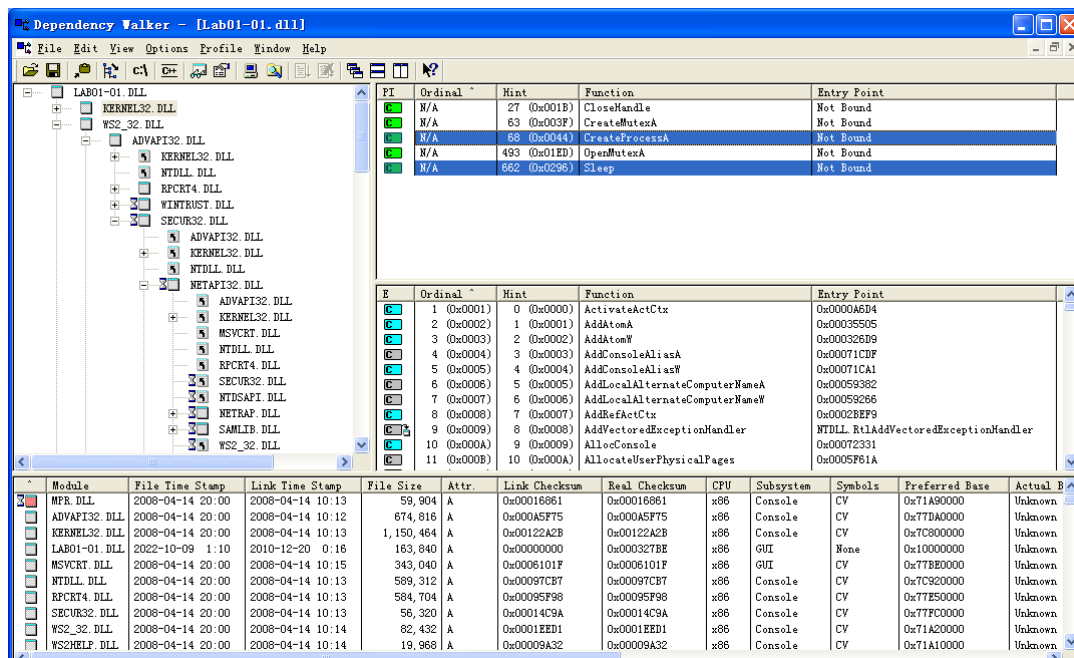


#### 4. 是否有导入函数显示出了这个恶意代码是做什么的？如果是，有哪些导入函数

- 在虚拟机Windows XP系统下使用Dependency Walker软件打开 Lab01-01.exe，发现 KERNEL32.DLL和MSVCRT.DLL两个动态链接库，从MSVCRT.DLL导入的函数通常是被每一个可执行文件都包含，因为它们是作为包装代码被编译器加入可执行文件的。查看从 KERNEL32.DLL导入的函数时，可以看到一些打开与操作文件的函数以及FindFirstFile、FindNextFile 和 CopyFile，这些函数意味着，恶意代码可以对文件进行搜索、打开、复制和修改文件，虽然不能确定恶意代码在搜索什么文件，但.exe字符串说明，恶意代码正在寻找搜索目标系统上的可执行文件。



- 在虚拟机Windows XP系统下使用Dependency Walker软件打开 Lab01-01.dll显示了导入DLL列表，发现KERNEL32.DLL和WS2\_32.DLL两个动态链接库。WS2\_32.dll 则提供了联网功能，打开WS2\_32.DLL发现这些导入函数都是按照序号进行导入的。打开KERNEL32.DLL可以看到CreateProcess 和 Sleep两个函数，这两个函数普遍在后门程序中使用。



## 5. 是否有任何其他文件或基于主机的迹象，让你可以在受感染系统上查找

- 通过Srnigs检查Lab01-01.exe时发现，同时观察到 C:\Windows\System32\kernel32.dll 和 C:\Windows\System32\kerne132.dll，文件kerne132.dll，用数字1代替了字母l，是为了看起来像系统文件kernel32.dll而自己冒充混淆为Windows系统文件，因此这个文件可以用来在主机作为恶意代码感染的迹象进行搜索。
- 通过Srnigs检查Lab01-01.dll 的字符串。除了已知的 CreateProcessA 和 Sleep，我们还注意到 exec 和 sleep。exec 可能用于通过网络给后门程序传送命令，再利用 CreateProcess 函数运行某个程序。sleep 可能用于让后门程序进入休眠模式。

```

except handlers
_controlfp
_stricmp
kernel32.dll
kernel32.dll
.exe
C:\*
C:\windows\system32\kernel32.dll
kernel32.
Lab01-01.dll
C:\Windows\System32\Kernel32.dll
WARNING_THIS_WILL_DESTROY_YOUR_MACHINE
D:\Malware\Strings>

```

```

_incterm
malloc
adjust_fdiv
exec
sleep
hello
127.26.152.13
SADFHUHF
/OI0[0h0p0
14IG1[111
1Y2a2g2r2
3!3}3
D:\Malware\Strings>

```

## 6. 是否有基于网络的迹象，可以用来发现受感染机器上的这个恶意代码？

- 通过Srings检查Lab01-01.dll时发现其中包含一个私有子网IP地址 127.26.152.13的字符串，结合其调用的 WS2\_32.dll，猜测该程序可能联网通信。尽管 127 开头的 IP 为本地地址，在此处表明是用于教学目的，现实中很可能指向一个具体的外网 IP。这是一个很好的基于网络的恶意代码感染迹象，可以用来识别这个恶意代码。

## 7. 你猜这些文件的目的是什么？

- .dll 文件可能是一个后门，而.exe文件是用来安装与运行DLL文件的。

## 二. Lab1-2

### 1. 将Lab01-02.exe 文件上传至 <https://www.virustotal.com/> 进行分析并查看报告。文件匹配到了已有的反病毒软件特征吗？

- 匹配到了
- 当把Lab01-02.exe上传到<https://www.virustotal.com/>，结果如下图所示，这个文件匹配到55个反病毒引擎

55 / 72

55 security vendors and no sandboxes flagged this file as malicious

c876a332d7dd8da331cb8eee7ab7bf32752834d4b2b54eaa362674a2a48f64a6

Lab01-02.exe

3.00 KB Size

2022-10-07 05:43:11 UTC 1 day ago

checks-disk-space detect-debug-environment idle long-sleeps peexe upx via-tor

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 30+

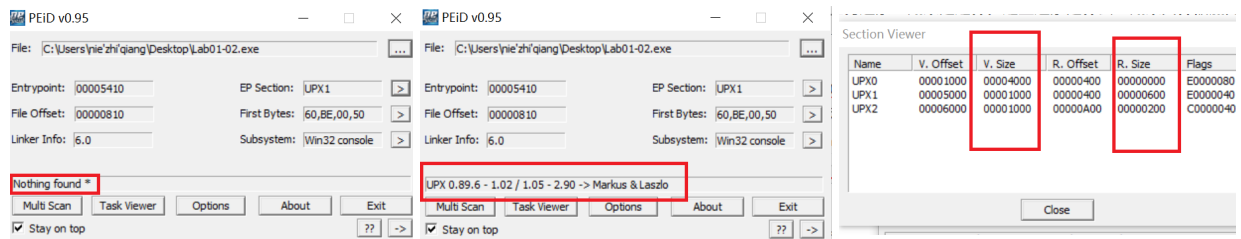
Security Vendors' Analysis

AhnLab-V3	Trojan.Win32.StartPage.C26214	Alibaba	TrojanClicker.Win32/Generic.1ba1980f
ALYac	Trojan.Startpage.3072	Antiy-AVL	Trojan/Generic.ASMalwS.330C
Arcabit	Trojan.Ser.Ulise.216	Avast	Win32:Malware-gen
AVG	Win32:Malware-gen	Avira (no cloud)	TR/Downloader.Gen
Baidu	Win32.Trojan-Clicker.Agent.ad	BitDefender	Gen:Variant.Ser.Ulise.216
BitDefenderTheta	Gen:NN.ZexaF.34698.amGfaWi867f	ClamAV	Win.Malware.Agent-6350563-0
Comodo	Malware@#22epuiwh8vym	CrowdStrike Falcon	Win/malicious_confidence_100% (W)
Cybereason	Malicious.878404	Cylance	Unsafe

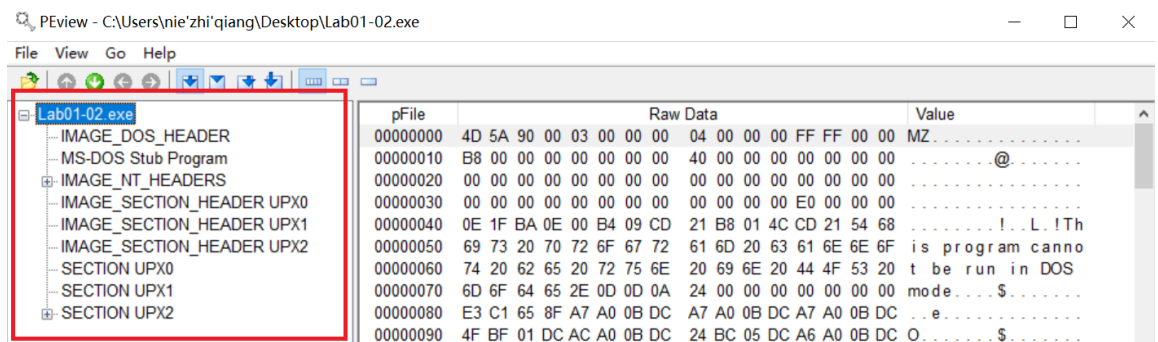


2. 是否有这个文件被加壳或混淆的任何迹象？如果是这样，这些迹象是什么？如果文件被加壳，请进行脱壳，如果可能的话。

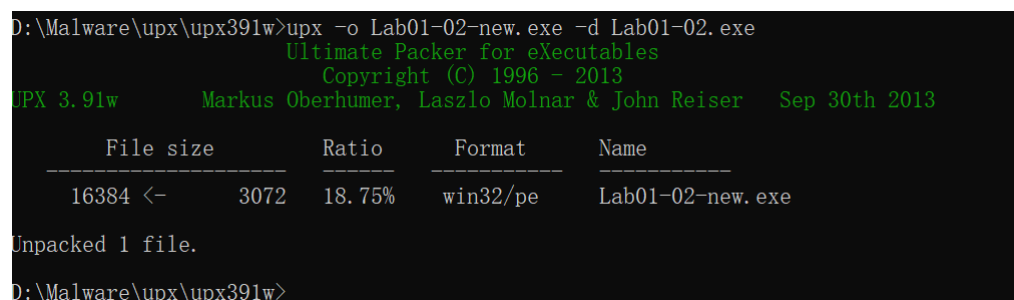
- 使用PEiD打开Lab01-02.exe, 显示Nothing found \*, 在PEiD的Options里选择Deep Scan, 发现是UPX加的壳。其次可以观察到UPX0段, 虚拟大小为0x4000, 而原始数据大小却为0, UPX0是长度最大的节, 标记为可执行, 因此其中可能包含了原始的未加壳代码。



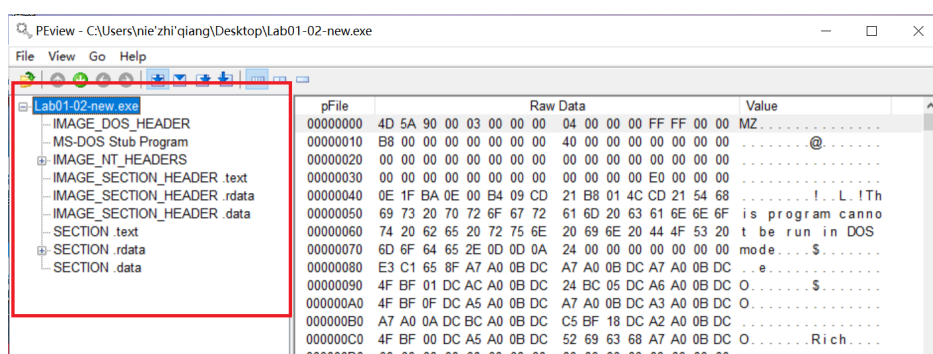
- 使用PEview打开Lab01-02.exe查看节区，出现UPX字段，明显是由UPX进行加壳后恶意代码程序的节名称

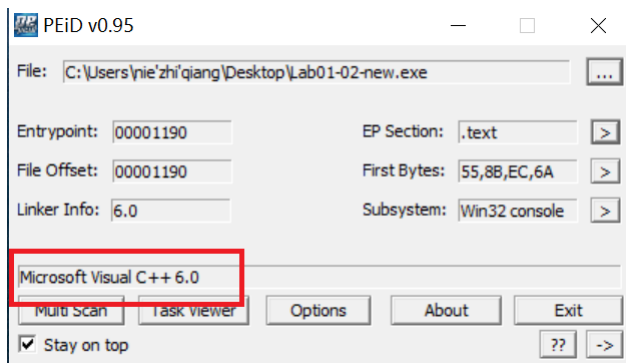


- 综上可以推断是UPX壳
- 通过UPX工具, 执行 `upx -o Lab01-02-new.exe -d Lab01-02.exe` 指令对Lab01-02.exe 进行脱壳



- 脱壳后再次通过PEiD和PEview，通过一下Microsoft Visual C++ 6.0编译等信息，可以看出此时没有加壳或者混淆的痕迹了，说明脱壳成功



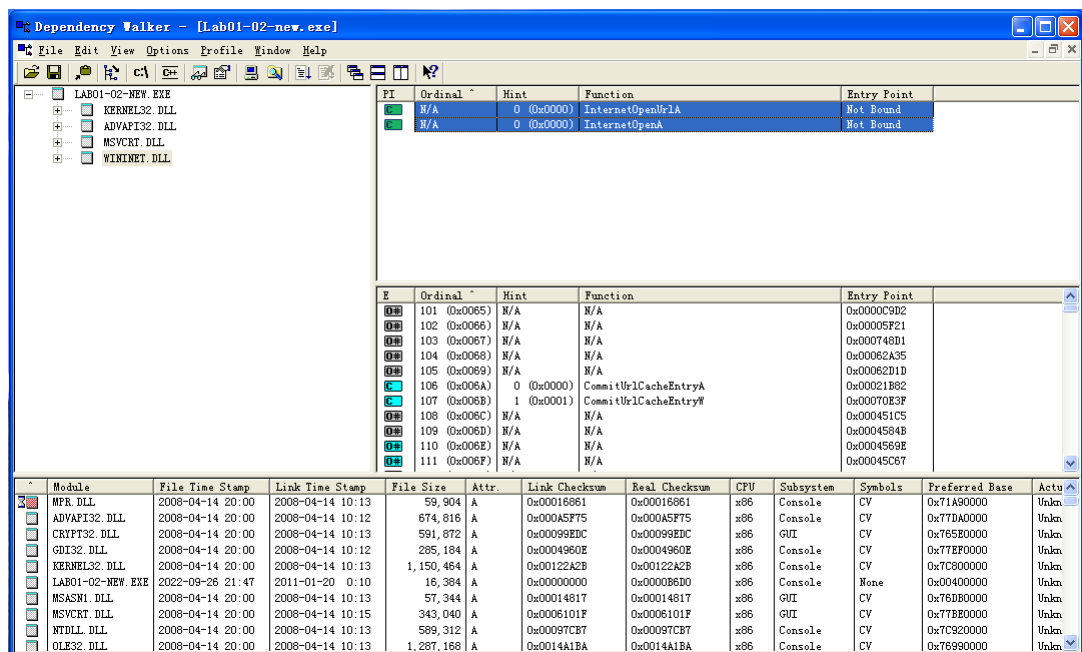


Section Viewer

Name	V. Offset	V. Size	R. Offset	R. Size	Flags
.text	00001000	000002DC	00001000	00001000	60000020
.rdata	00002000	00000372	00002000	00001000	40000040
.data	00003000	0000008C	00003000	00001000	C0000040

### 3. 有没有任何导入函数能够暗示出这个程序的功能？如果是，是哪些导入函数，他们会告诉你什么？

- 脱壳之后，在虚拟机Windows XP系统下使用Dependency Walker软件打开 Lab01-02-new.exe，KERNEL32.DLL和MSVCRT.DLL两个动态链接库中的函数通常是被每一个可执行文件都包含，所以它们能告诉我们关于这个恶意代码的信息很少。从WININET.DLL导入的函数InternetOpen和InternetOpenURL表示这个恶意代码会进行联网操作。从ADVAPI32.DLL导入的函数CreateService表示这个代码会创建一个服务。



### 4. 哪些基于主机或基于网络的迹象，可以被用来确定这个恶意代码所感染的机器？

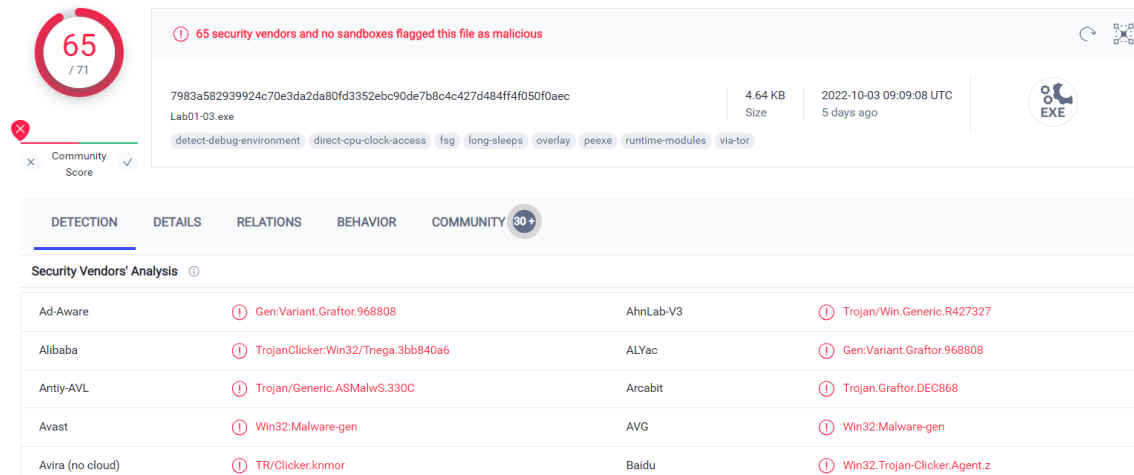
- 通过Strings工具检查字符串列表，发现http://www.malwareanalysisbook.com，这可能是InternetOpenURL函数中所打开的URL，还发现Malservice字符串，所以应该通过一个名为Malservice的服务，并通过http://www.malwareanalysisbook.com的网络流量，来检查恶意代码感染的主机

```
InternetOpenUrlA
InternetOpenA
Malservice
Malservice
HCL 345
http://www.malwareanalysisbook.com
Internet Explorer 8.0
```

### 三. Lab1-3

1. 将Lab01-03.exe 文件上传至 <https://www.virustotal.com/> 进行分析并查看报告。文件匹配到了已有的反病毒软件特征吗？

- 匹配到了
- 当把Lab01-02.exe上传到<https://www.virustotal.com/>，结果如下图所示，这个文件匹配到65个反病毒引擎

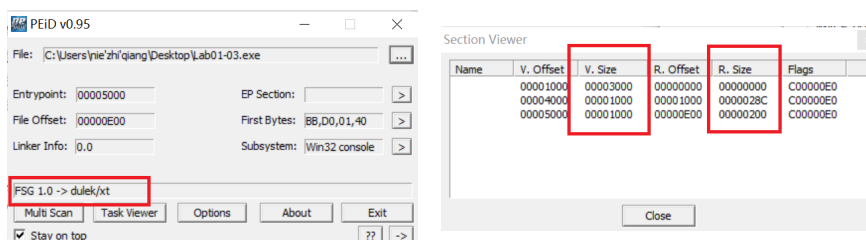


The image shows a VirusTotal analysis report for the file Lab01-03.exe. At the top, a red circle indicates a score of 65/71. A message states: "65 security vendors and no sandboxes flagged this file as malicious". The file's SHA-256 hash is 7983a582939924c70e3da2da80fd3352ebc90de7b8c4c427d484ff4f050f0aec. The file size is 4.64 KB, and it was uploaded 5 days ago. Below this, a table lists detections from various security vendors:

Detection	Details
Ad-Aware	Gen.Variant.Graftor.968808
Alibaba	TrojanClicker.Win32/Tnega.3bb840a6
Antiy-AVL	Trojan/Generic.ASMalwS.330C
Avast	Win32:Malware-gen
Avira (no cloud)	TR/Clicker.knmor
AhnLab-V3	Trojan/Win.Generic.R427327
ALYac	Gen.Variant.Graftor.968808
Arcabit	Trojan.Graftor.DEC868
AVG	Win32:Malware-gen
Baidu	Win32.Trojan-Clicker.Agent.z

2. 是否有这个文件被加壳或混淆的任何迹象？如果是这样，这些迹象是什么？如果文件被加壳，请进行脱壳，如果可能的话。

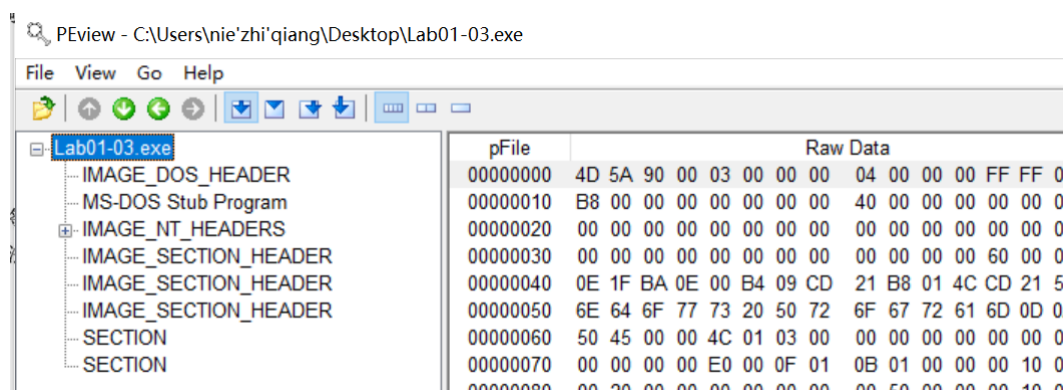
- 使用PEiD打开Lab01-03.exe观察到，其将加壳器标识为FSG 1.0-> dulek/xt，并且注意到首节虚拟大小为0x3000，而原始数据大小却为0，这意味着Windows将会为.text分配内存空间，加壳器将会脱出可执行代码到一个分配的.text节中



The image contains two screenshots. The left screenshot shows the PEiD v0.95 interface with the file C:\Users\nie'zhi'qiang\Desktop\Lab01-03.exe loaded. The 'FSG 1.0 -> dulek/xt' entry is highlighted in a red box. The right screenshot shows the 'Section Viewer' window with a table of sections. The 'V. Size' and 'R. Size' columns for the first three sections are highlighted in red boxes:

Name	V. Offset	V. Size	R. Offset	R. Size	Flags
00001000	00003000	00000000	00000000	C00000E0	
00004000	00001000	00001000	0000028C	C00000E0	
00005000	00001000	00000E00	00000200	C00000E0	

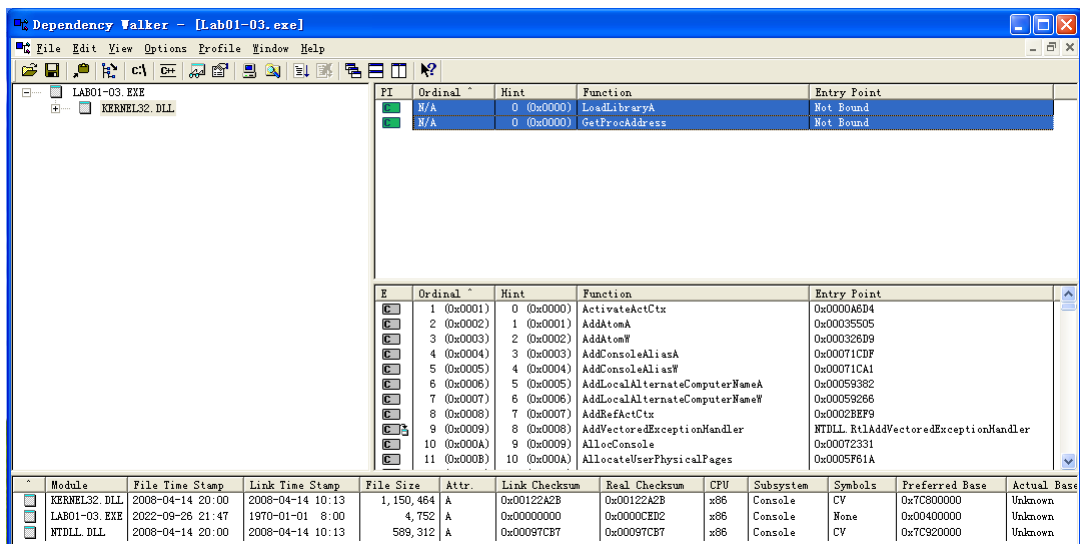
- 使用PEview打开Lab01-03.exe，可以发现文件得节没有名字



The image shows a screenshot of the PEview application. The file C:\Users\nie'zhi'qiang\Desktop\Lab01-03.exe is open. The 'pFile' pane on the left shows the file structure, including 'IMAGE\_DOS\_HEADER', 'MS-DOS Stub Program', 'IMAGE\_NT\_HEADERS', and several 'IMAGE\_SECTION\_HEADER' entries. The 'Raw Data' pane on the right shows the raw data of the file, with the first few bytes being 00 00 00 00 00 00 00 00.



- 由上述特征可以判断这个文件是加壳的，未脱壳前，只能在 kernel32.dll 中看到 LoadLibrary 和 GetProcAddress 导入函数。经过寻找万能脱壳软件、自学部分手动脱壳知识等还未能成功进行脱壳。因此后两问目前无法回答。



#####

## 四. Lab1-4

- 将Lab01-03.exe 文件上传至 <https://www.virustotal.com/> 进行分析并查看报告。文件匹配到了已有的反病毒软件特征吗？

- 匹配到了
- 62个反病毒引擎和1个沙箱

62  
/ 72

62 security vendors and 1 sandbox flagged this file as malicious

0fa1498340fca6562cfa389ad3e93395f44c72fd128d7ba08579a69aaf3b126

Lab01-04.exe

armadillo idle peexe via-tor

36.00 KB  
Size

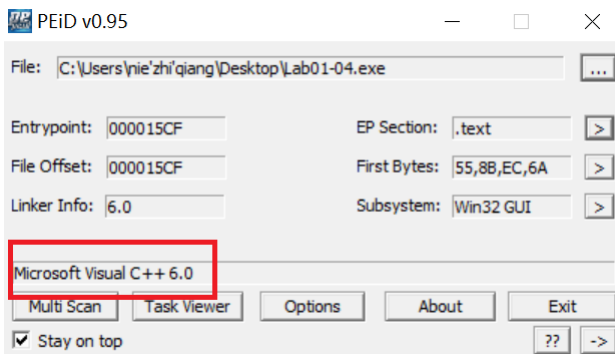
2022-09-29 20:54:10 UTC  
9 days ago

EXE

DETECTION	DETAILS	RELATIONS	BEHAVIOR	COMMUNITY
30+				
Security Vendors' Analysis ⓘ				
Ad-Aware	🔴 Gen:Variant.Cerbu.64782		Alibaba	🔴 TrojanDownloader:Win32/DownLdr.080f6...
ALYac	🔴 Gen:Variant.Cerbu.64782		Antiy-AVL	🔴 Trojan/Generic.ASMalwS.3304
Arcabit	🔴 Trojan.Generic		Avast	🔴 Win32:DropperX-gen [Drp]
AVG	🔴 Win32:DropperX-gen [Drp]		Avira (no cloud)	🔴 TR/Dldr.Small.romlh
BitDefender	🔴 Gen:Variant.Cerbu.64782		BitDefenderTheta	🔴 AI:Packer.6911D1B71F
Bkav Pro	🔴 W32.AI.Detect.malware2		ClamAV	🔴 Win.Trojan.Agent-375080

- 是否有这个文件被加壳或混淆的任何迹象？如果是这样，这些迹象是什么？如果文件被加壳，请进行脱壳，如果可能的话。

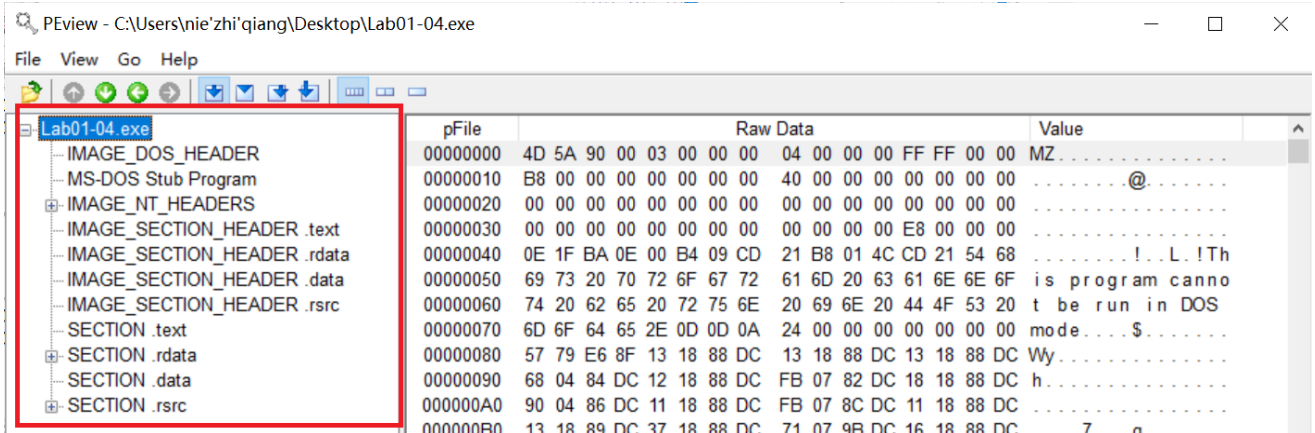
- 将 Lab01-04.exe 导入到PEview和PEiD软件中，发现PE头部分组织良好并且没有提示加壳信息，虚拟大小和原始数据大小基本一致，因此没有迹象显示这个文件是加壳或混淆的。



Section Viewer

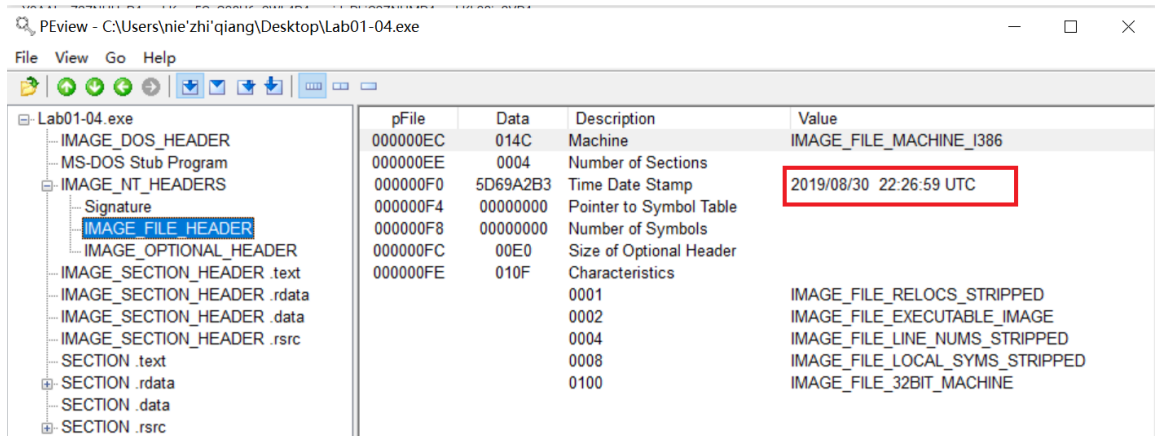
Name	V. Offset	V. Size	R. Offset	R. Size	Flags
.text	00001000	00000720	00001000	00001000	60000020
.rdata	00002000	000003D2	00002000	00001000	40000040
.data	00003000	0000014C	00003000	00001000	C0000040
.rsrc	00004000	00004060	00004000	00005000	40000040

Close



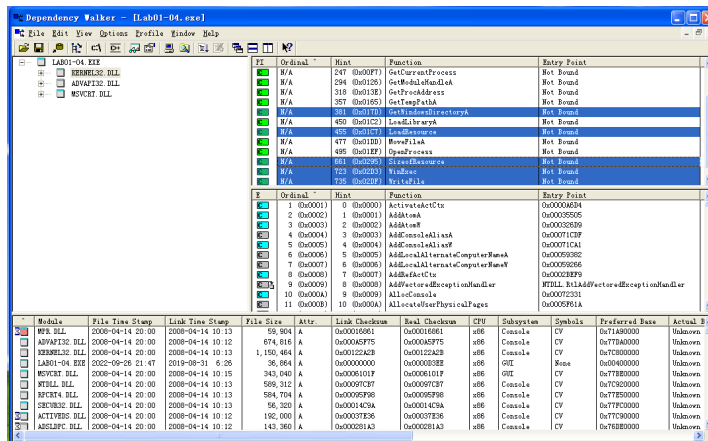
### 3. 这个文件是什么时候编译的？

- 通过PVIEW -> IMAGE\_NT\_HEADERS -> IMAGE\_FILE\_HEADER查看到该文件编译时间为2019年8月30日，同其创建等时间相比，很有可能这个编译时间是伪造的，目前还不能确定这个文件到底是什么时候编译的。



### 4. 有没有任何导入函数能够暗示出这个程序的功能？如果是，是哪些导入函数，他们会告诉你什么？

- 从ADVAPI32.DLL导入的函数表示程序做了一些与权限有关的事情，可以假设它试图访问使用了特殊权限进行保护的文件。从KERNEL32.DLL的导入函数告诉我们这个程序从资源节中装载数据（LoadResource、FindResource和SizeofResource），并写一个文件到磁盘上（CreateFile和WriteFile），接着执行一个磁盘上的文件（WinExec）。我们也可以猜测这个程序将文件写入到了系统目录，因为它调用了GetWindowDirectory函数



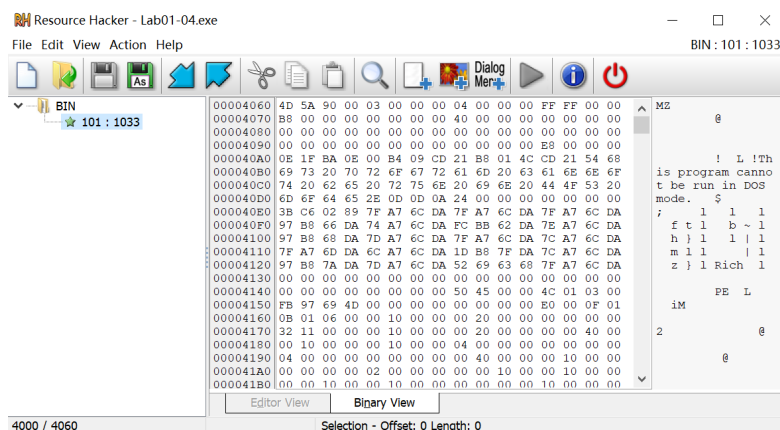
## 5. 哪些基于主机或基于网络的迹象，可以被用来确定这个恶意代码所感染的机器？

- 检查字符串，如下图所示
- 出现\system32\wupdmgrd.exe（Windows 升级管理器），结合GetWindowDirectory函数调用，表明恶意代码在C:\windows\system32\wupdmgrd.exe位置创建或修改一个文件。
- www.malwareanalysisbook.com/updater.exe 很可能是要下载的恶意代码的存储位置，或者是伪装成这个文件。URLDownloadToFile 则间接印证了下载器的功能。

```
adjust_rdiv
p_commode
p_fmode
set_app_type
except_handler3
controlfp
\winup.exe
%$%$
\system32\wupdmgrd.exe
%$%$
http://www.practicalmalwareanalysis.com/updater.exe
D:\Malware\Strings>
```

## 6. 这个文件在资源段中包含一个资源，使用Resource Hacker工具来检查资源，然后抽取资源，资源中你能发现什么吗？

- 可以看到资源段中还有一个可执行文件（101:1033），观察到字符串！This program cannot be run in DOS mode，这个字符串是在所有PE文件处的DOS头部中都会包含错误消息，于是我认为这一资源其实是在Lab01-04.exe资源节中存储的另一个可执行文件。



- 右键 101:1033，选择 Save Resource to a BIN file保存，查看导入表，可以看到嵌入文件在访问一下网络函数，它调用了URLDownloadToFile，一个由恶意下载器普遍使用的函数，它调用了WinExec函数，可能执行了下载到的文件。

Dependency Walker - [BIN101.bin]

File	Ordinal	Hint	Function	Entry Point
N/A	357 (0x0165)		GetTempPathA	Not Bound
N/A	381 (0x017D)		GetWindowsDirectoryA	Not Bound
N/A	723 (0x02B3)		WinExec	Not Bound

File	Ordinal	Hint	Function	Entry Point
1 (0x0001)	0 (0x0000)		ActivateActCtx	0x0000A6D4
2 (0x0002)	1 (0x0001)		AddAtomA	0x00035505
3 (0x0003)	2 (0x0002)		AddAtomW	0x000326D9
4 (0x0004)	3 (0x0003)		AddConsoleAliasA	0x00071C1F
5 (0x0005)	4 (0x0004)		AddConsoleAliasW	0x00071CA1
6 (0x0006)	5 (0x0005)		AddLocalAlternateComputerNameA	0x00059382
7 (0x0007)	6 (0x0006)		AddLocalAlternateComputerNameW	0x00059266
8 (0x0008)	7 (0x0007)		AddRefactCtx	0x0002BFF9
9 (0x0009)	8 (0x0008)		AddVectoredExceptionHandler	NTDLL.RtlAddVectoredExceptionHandler
10 (0x000A)	9 (0x0009)		AllocConsole	0x00072331
11 (0x000B)	10 (0x000A)		AllocateUserPhysicalPages	0x0005F61A

Module	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum	CPU	Subsystem	Symbols	Preferred Base	Actual B
MFR.DLL	2008-04-14 20:00	2008-04-14 10:13	59,904	A	0x00016861	0x00016861	x86	Console	CV	0x71A90000	Unknown
ADVAPI32.DLL	2008-04-14 20:00	2008-04-14 10:12	674,816	A	0x000A5F75	0x000A5F75	x86	Console	CV	0x77DA0000	Unknown
BIN101.BIN	2022-10-09 12:22	2011-02-27 8:16	16,384	A	0x00000000	0x00006544	x86	GUI	None	0x00400000	Unknown
GDI32.DLL	2008-04-14 20:00	2008-04-14 10:12	285,184	A	0x0004960E	0x0004960E	x86	Console	CV	0x77EF0000	Unknown
KERNEL32.DLL	2008-04-14 20:00	2008-04-14 10:13	1,150,464	A	0x00122A2B	0x00122A2B	x86	Console	CV	0x7C800000	Unknown
MSVCRT.DLL	2008-04-14 20:00	2008-04-14 10:15	343,040	A	0x0006101F	0x0006101F	x86	GUI	CV	0x77BE0000	Unknown
NTDLL.DLL	2008-04-14 20:00	2008-04-14 10:13	589,312	A	0x00097CB7	0x00097CB7	x86	Console	CV	0x7C920000	Unknown
OLE32.DLL	2008-04-14 20:00	2008-04-14 10:13	1,287,168	A	0x0014A1BA	0x0014A1BA	x86	Console	CV	0x76990000	Unknown
RPCRT4.DLL	2008-04-14 20:00	2008-04-14 10:13	584,704	A	0x00095F98	0x00095F98	x86	Console	CV	0x77E50000	Unknown
SECUR32.DLL	2008-04-14 20:00	2008-04-14 10:13	56,320	A	0x00014C9A	0x00014C9A	x86	Console	CV	0x77FC0000	Unknown

Dependency Walker - [BIN101.bin]

File	Ordinal	Hint	Function	Entry Point
N/A	62 (0x003E)		URLDownloadToFileA	Not Bound

File	Ordinal	Hint	Function	Entry Point
100 (0x0064)	N/A		N/A	0x0005F7C1
101 (0x0065)	0 (0x0000)		AsyncGetClassBits	0x0003EBE9
102 (0x0066)	1 (0x0001)		AsyncInstallDistributionUnit	0x0003EBE9
103 (0x0067)	2 (0x0002)		BindAsyncMoniker	0x0002D1A7
104 (0x0068)	3 (0x0003)		CILGetLongPathNameA	0x0003F0F9
105 (0x0069)	4 (0x0004)		CILGetLongPathNameW	0x0003F124
106 (0x006A)	5 (0x0005)		CoGetClassObjectFromURL	0x0003EB47
107 (0x006B)	6 (0x0006)		CoInstall	0x0003D3BF
108 (0x006C)	7 (0x0007)		CoInternetCombineUrl	0x00016485
109 (0x006D)	8 (0x0008)		CoInternetCompareUrl	0x0002F133
110 (0x006E)	9 (0x0009)		CoInternetCreateSecurityManager	0x000030E7

Module	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum	CPU	Subsystem	Symbols	Preferred Base	Actual B
MFR.DLL	2008-04-14 20:00	2008-04-14 10:13	59,904	A	0x00016861	0x00016861	x86	Console	CV	0x71A90000	Unknown
ADVAPI32.DLL	2008-04-14 20:00	2008-04-14 10:12	674,816	A	0x000A5F75	0x000A5F75	x86	Console	CV	0x77DA0000	Unknown
BIN101.BIN	2022-10-09 12:22	2011-02-27 8:16	16,384	A	0x00000000	0x00006544	x86	GUI	None	0x00400000	Unknown
GDI32.DLL	2008-04-14 20:00	2008-04-14 10:12	285,184	A	0x0004960E	0x0004960E	x86	Console	CV	0x77EF0000	Unknown
KERNEL32.DLL	2008-04-14 20:00	2008-04-14 10:13	1,150,464	A	0x00122A2B	0x00122A2B	x86	Console	CV	0x7C800000	Unknown
MSVCRT.DLL	2008-04-14 20:00	2008-04-14 10:15	343,040	A	0x0006101F	0x0006101F	x86	GUI	CV	0x77BE0000	Unknown
NTDLL.DLL	2008-04-14 20:00	2008-04-14 10:13	589,312	A	0x00097CB7	0x00097CB7	x86	Console	CV	0x7C920000	Unknown
OLE32.DLL	2008-04-14 20:00	2008-04-14 10:13	1,287,168	A	0x0014A1BA	0x0014A1BA	x86	Console	CV	0x76990000	Unknown
RPCRT4.DLL	2008-04-14 20:00	2008-04-14 10:13	584,704	A	0x00095F98	0x00095F98	x86	Console	CV	0x77E50000	Unknown
SECUR32.DLL	2008-04-14 20:00	2008-04-14 10:13	56,320	A	0x00014C9A	0x00014C9A	x86	Console	CV	0x77FC0000	Unknown

## 七. 实验心得

- 对于如何判断加壳方式有了全面的了解，不仅局限在使用PEiD查看加壳器的提示，而是学会根据.text等节的虚拟大小和原始数据大小对比以及
- 对于一些特定的功能性函数有了了解，通过导入函数判断相应恶意代码功能
- 存疑（周一上课准备去问老师）：如果PEiD显示Nothing，一定加壳？如果没有加壳，PEiD一定显示编译方式？