

# 南開大學

## 惡意代碼分析与防治技術課程實驗報告

### 實驗一



学 院 网络空间安全学院  
专 业 信息安全、法学  
学 号 2113203  
姓 名 付政烨  
班 级 信安法班

# 一、实验目的

为学生提供练习本章所教技能的机会。为了模拟真实的恶意软件分析，实验中 will 提供很少或没有有关正在分析的程序的信息。与本书中的所有其他实验一样，基本的静态分析实验室文件都使用通用名称，以模拟通常使用无意义或误导性名称的未知恶意软件。这样的设计有助于学生锻炼并应用在章节中学到的技能来解决具有挑战性的问题。

# 二、实验原理

通过实际操作来强化在恶意软件分析方面的技能，提供一个模拟的环境，使其能够应对真实世界中可能遇到的挑战。这种实践性的学习方法有助于更好地理解和应用所学的理论知识。

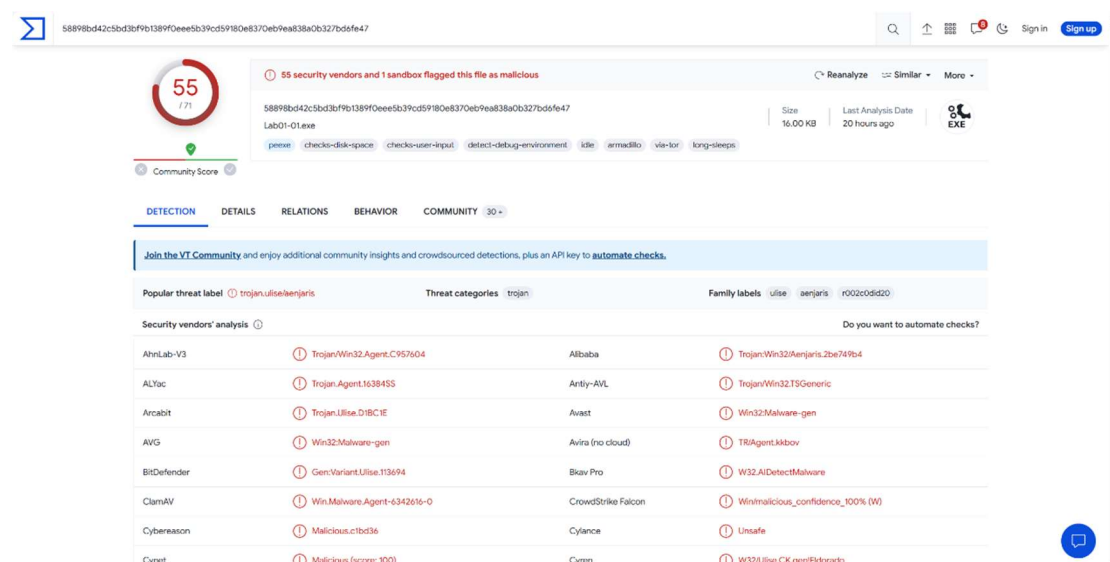
# 三、实验过程

## Chapter\_1L

### Lab 1-1

1. Upload the files to <http://www.VirusTotal.com/> and view the reports. Does either file match any existing antivirus signatures?

Lab01-01.exe:



The screenshot shows the VirusTotal analysis report for the file Lab01-01.exe. The file has a Community Score of 55/71, indicating it is likely malicious. The report shows that 55 security vendors and 1 sandbox flagged this file as malicious. The file is identified as a Trojan, with threat categories including 'trojan' and family labels 'ulise' and 'benjaris'. The report also lists various security vendors that have analyzed the file, such as AhnLab-V3, ALYac, Arcabit, AVG, BitDefender, ClamAV, Cybereason, Cynet, Alibaba, Antiy-AVL, Avast, Avira (no cloud), BitDefender, CrowdStrike Falcon, Cylance, and Cyren. The file is identified as a Trojan, with threat categories including 'trojan' and family labels 'ulise' and 'benjaris'.

Security vendors' analysis	Threat categories	Family labels
AhnLab-V3	Trojan:Win32.Agent.C957604	Alibaba
ALYac	Trojan.Agent.1638455	Antiy-AVL
Arcabit	Trojan.Ulise.DIBCIE	Avast
AVG	Win32/Malware-gen	Avira (no cloud)
BitDefender	Gen:Variant.Ulise.113694	BitDefender
ClamAV	Win.Malware.Agent-6342616-0	CrowdStrike Falcon
Cybereason	Malicious.c1bd36	Cylance
Cynet	Malicious (score: 100)	Cyren

Lab01-01.dll:

44

70

44 security vendors and no sandboxes flagged this file as malicious

f50e42c8dfaab649bde0398867e930b86c2a599e8db83b8260393082268f2dba

Lab01-01.dll

Size  
160.00 KB

Last Analysis Date  
18 minutes ago

DLL

pcid

armed80

via-kor

Community Score

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY 30

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label

trojan.jalk/skeeyah

Threat categories

trojan

Family labels

jalk skeeyah r002c0phf20

Security vendors' analysis

Do you want to automate checks?

Alibaba	Trojan:Win32/Skeeyah.7fb0ebff	ALYac	Trojan.Agent.Waski
Antiy-AVL	Trojan:Win32/BTSGeneric	Arcabit	Trojan.Jalk.D29746
Avast	Win32/Malware-gen	AVG	Win32/Malware-gen
BitDefender	Gen:Variant.Jalk.169798	BitDefender Theta	Gen:NN.Zedsof.36642.kq4@vGkQVtp
Bkav Pro	W32/AIDetect/Malware	ClamAV	Win.Malware.Agent-6369568-0
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	Cylance	Unsafe
Cynet	Malicious (score: 100)	Cyren	W32/Skeeyah.AK.gen/Eldorado
DeepInstinct	MALICIOUS	Elastic	Malicious (High Confidence)

根据反馈的信息，Lab01-01.exe 被 55 个安全厂商和 1 个沙箱标记为恶意文件，Lab01-01.dll 被 44 个安全厂商标记为恶意文件，但没有沙箱标记它为恶意文件。综上所述，这表明这两个文件均已可以被识别为病毒。

2. When were these files compiled?

Lab01-01.dll

History ⓘ	
Creation Time	2010-12-19 16:16:38 UTC
First Seen In The Wild	2010-12-19 11:16:38 UTC
First Submission	2011-07-04 19:57:48 UTC
Last Submission	2023-09-17 03:26:47 UTC
Last Analysis	2023-09-17 03:24:58 UTC

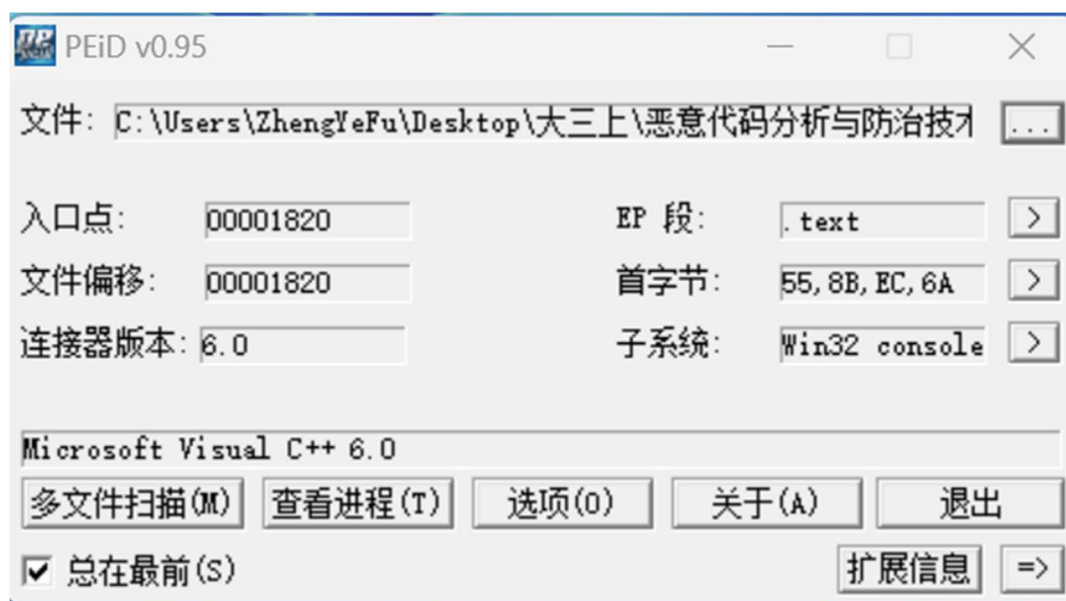
Lab01-01.exe

## History ⓘ

Creation Time	2010-12-19 16:16:19 UTC
First Seen In The Wild	2012-01-08 02:19:06 UTC
First Submission	2012-02-16 07:31:54 UTC
Last Submission	2023-09-17 03:47:45 UTC
Last Analysis	2023-09-16 07:01:40 UTC

根据网站的分析报告可知，两个晚间的编译时间均是 2010 年 12 月 19 日

3. Are there any indications that either of these files is packed or obfuscated? If so, what are these indicators?

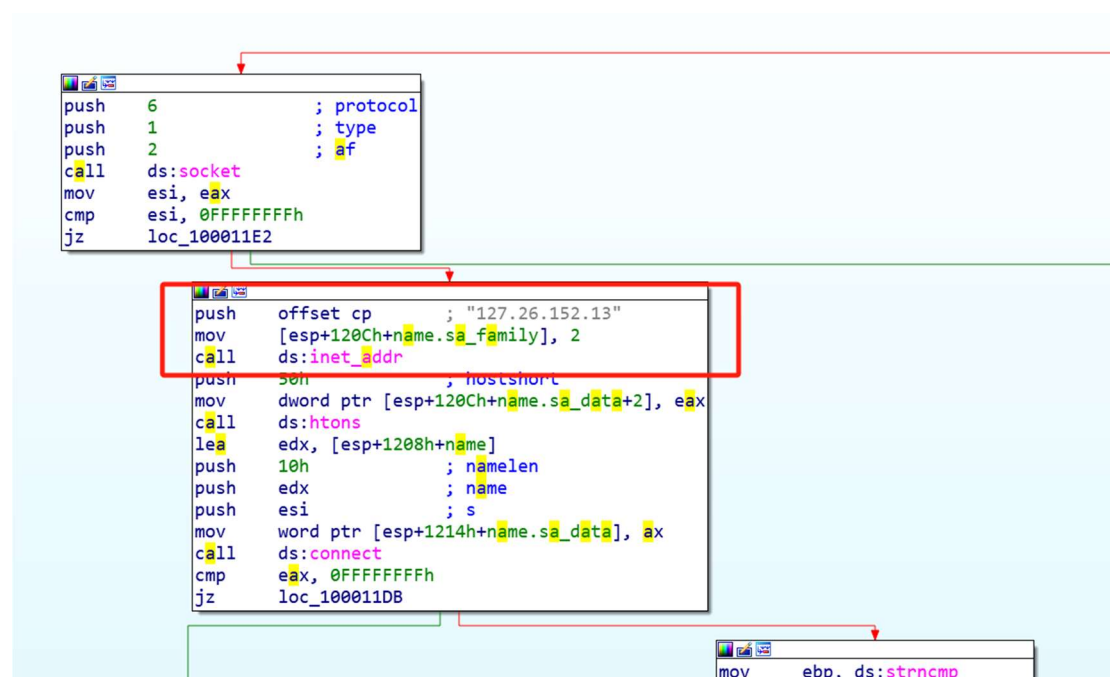


使用 PEiD 工具对这两个文件进行了分析，得到的结果非常相似，如下图所示。分析结果表明，这两个文件都是使用 Visual C++ 编译的，并且没有检测到任何加壳信息。因此，可以得出结论这两个文件没有受到加壳或混淆的影响，没有显示出这方面的迹象。

4. Do any imports hint at what this malware does? If so, which imports are they?



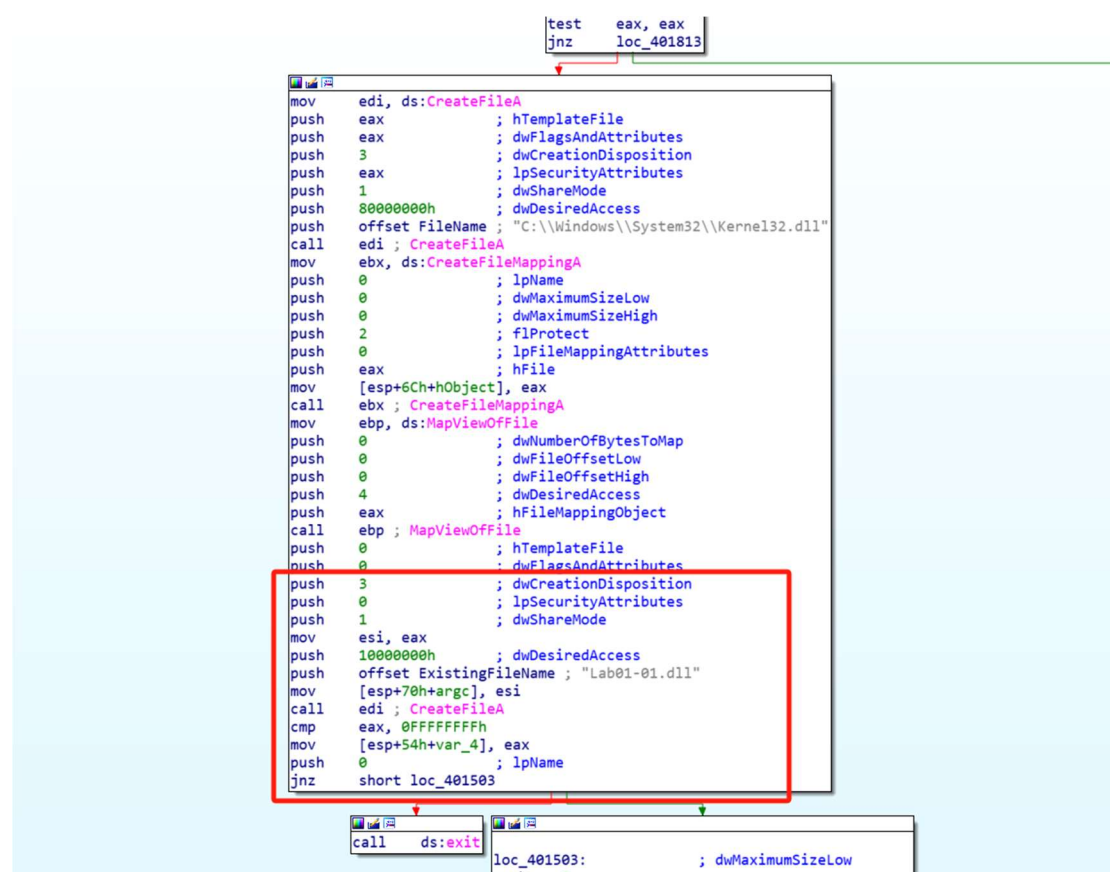
## 6. What network-based indicators could be used to find this malware on infected machines?



通过使用 IDA Pro 对 DLL 文件进行分析，进行字符串分析时发现以下信息：在该文件中存在一个 IP 地址，即 127.26.152.13。然而，需要注意的是，该 IP 地址并非公网 IP 地址，而是一个内部或私有网络 IP 地址。因此，根据分析结果，可以合理地得出结论，该程序不太可能是设计用于进行网络传播攻击的恶意软件

## 7. What would you guess is the purpose of these files?

Segment	Address	Ordinal	Name	Library
.text	10002000		Sleep	KERNEL32
.text	10002004		CreateProcessA	KERNEL32
.text	10002008		CreateMutexA	KERNEL32
.text	1000200C		OpenMutexA	KERNEL32
.text	10002010		CloseHandle	KERNEL32
	10002018		_adjust_fdiv	MSVCRT
	1000201C		malloc	MSVCRT
	10002020		_initterm	MSVCRT
	10002024		free	MSVCRT
	10002028		strncmp	MSVCRT
	10002030	23	socket	WS2_32
	10002034	115	WSAStartup	WS2_32
	10002038	11	inet_addr	WS2_32
	1000203C	4	connect	WS2_32
	10002040	19	send	WS2_32
	10002044	22	shutdown	WS2_32
	10002048	16	recv	WS2_32
	1000204C	3	closesocket	WS2_32
	10002050	116	WSACleanup	WS2_32
	10002054	9	htons	WS2_32



在对 exe 文件进行反汇编代码分析时，发现了以上代码片段。根据这个代码片段以及其他反汇编结果，可以初步推断这个 exe 文件可能用于执行 dll 文件，而该 dll 文件导入了 sleep 和 CreateProcess 等函数。这一迹象可能表明这个 exe 文件潜在地具有后门程序的特征。

## Lab 1-2

1. Upload the Lab01-02.exe file to <http://www.VirusTotal.com/>. Does it match any existing antivirus definitions?



56 / 71

56 security vendors and 1 sandbox flagged this file as malicious

Reanalyze Similar More

c876a332d7d9da331cb8ee7ab7bf32752834d4b2b54eaa362674a2a48f64ad

Lab01-02.exe

Size: 3.00 KB Last Analysis Date: 1 day ago

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

Community Score

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 30

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: trojan.ulise/trojanclicker Threat categories: trojan downloader Family labels: ulise trojanclicker startpage

Security vendors' analysis

AhriLab-V3	Trojan.Win32.StartPage.C26214	Alibaba	Trojan.Clicker.Win32.Generic.47b7b5e4
ALYac	Trojan.Startpage.3072	Antiy-AVL	Trojan.Win32.Generic
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
BitDefender Theta	Gen.NN.Zenaf.36662.amGfaW867f	Bkav Pro	W32.AIDetectMalware
ClamAV	Win.Malware.Agent-6350563-0	CrowdStrike Falcon	Win/malicious_confidence_100% (W)

Do you want to automate checks?

有 56 个安全厂商和 1 个沙箱将此文件标记为恶意文件。

2. Are there any indications that this file is packed or obfuscated? If so, what are these indicators? If the file is packed, unpack it if possible.

PEiD v0.95

文件: Analysis Labs\BinaryCollection\Chapter\_1L\Lab01-02.exe

入口点: 00005410 EP 段: UPX1

文件偏移: 00000810 首字节: 60, BE, 00, 50

连接器版本: 6.0 子系统: Win32 console

什么都没找到 \*

多文件扫描(M) 查看进程(T) 选项(O) 关于(A) 退出

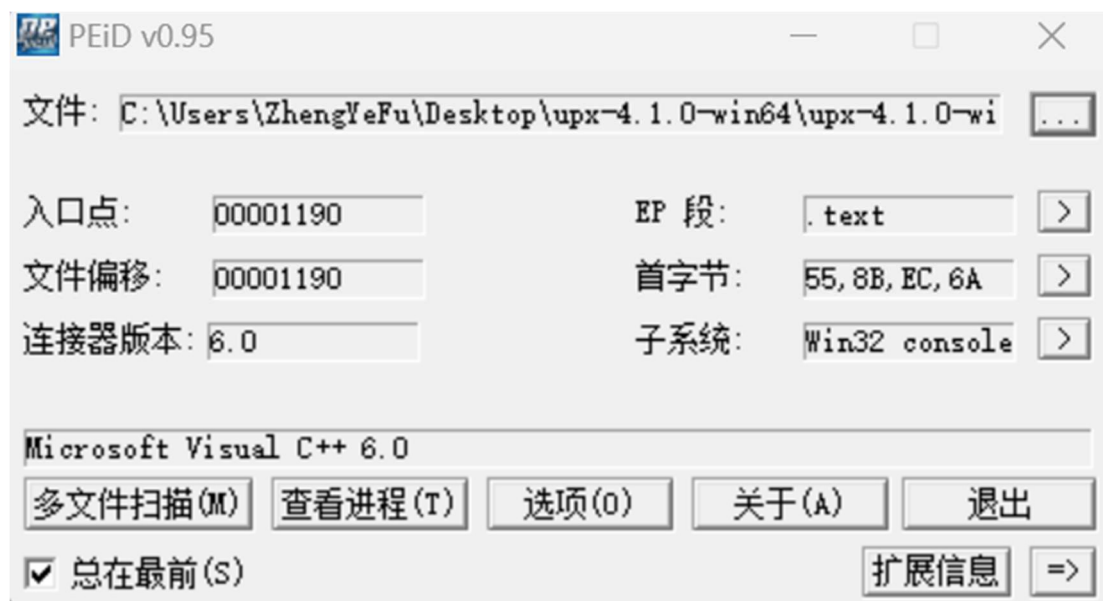
☒ 总在最前(S) 扩展信息 =>

```
C:\Users\ZhengYeFu\Desktop\upx-4.1.0-win64\upx-4.1.0-win64>upx -o up1-2.exe -d Lab01-02.exe
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2023
UPX 4.1.0 Markus Oberhumer, Laszlo Molnar & John Reiser Aug 8th 2023

File size      Ratio      Format      Name
-----
16384 <-      3072      18.75%      win32/pe      up1-2.exe

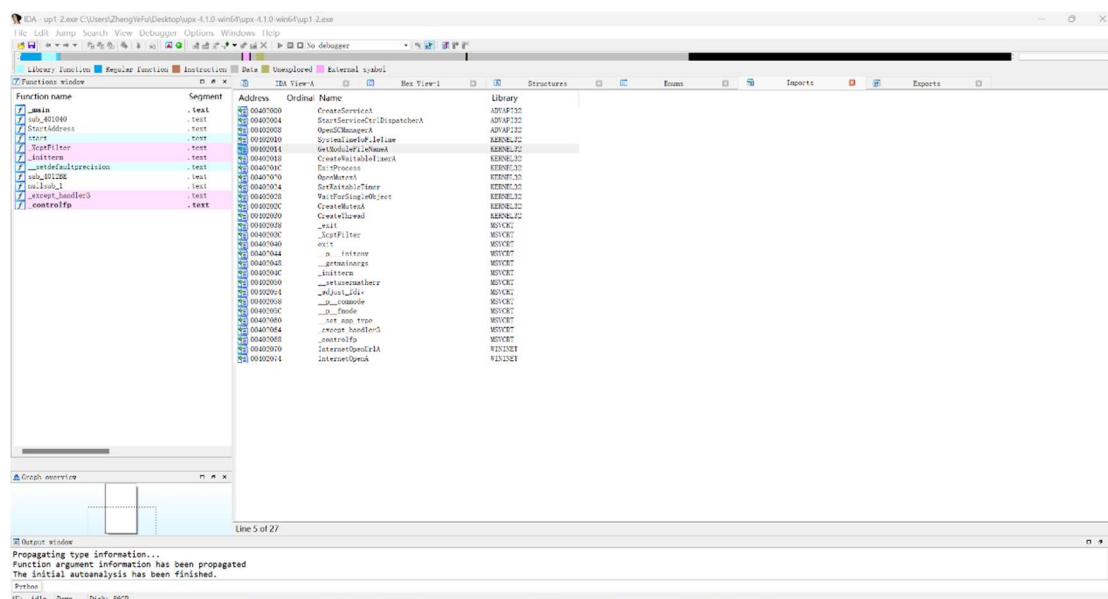
Unpacked 1 file.
```





使用 PEiD 分析可知，该程序使用了 UPX 工具加壳，安装了 upx 后，使用如下指令进行脱壳后，显示已经脱壳成功。

### 3. Do any imports hint at this program's functionality? If so, which imports are they and what do they tell you?



针对脱壳后的程序进行反汇编分析，得到的导入表如下：可以看到，程序导入了一些 CreateService、InternetOpen 函数等，可以初步判断该程序会对网络进行一些操作。

### 4. What host- or network-based indicators could be used to identify this malware on infected machines?

```

ServiceStartTable= SERVICE_TABLE_ENTRYA ptr -10h
var_8= dword ptr -8
var_4= dword ptr -4
argc= dword ptr 4
argv= dword ptr 8
envp= dword ptr 0Ch

sub     esp, 10h
lea     eax, [esp+10h+ServiceStartTable]
mov     [esp+10h+ServiceStartTable.lpServiceName], offset aMalService ; "MalService"
push    eax ; lpServiceStartTable
mov     [esp+14h+ServiceStartTable.lpServiceProc], offset sub_401040
mov     [esp+14h+var_8], 0
mov     [esp+14h+var_4], 0
call    ds:StartServiceCtrlDispatcherA
push    0
push    0
call    sub_401040
add     esp, 18h
retn
_main endp

```

```

loc_40116D: ; CODE XREF: StartAddress+30↓j
          push    0 ; dwContext
          push    80000000h ; dwFlags
          push    0 ; dwHeadersLength
          push    0 ; lpszHeaders
          push    offset szUrl ; "http://www.malwareanalysisbook.com"
          push    esi ; hInternet
          call    edi ; InternetOpenUrlA
          jmp     short loc_40116D

StartAddress endp

```

该程序存在 MalService 服务, 并且会链接到 <http://www.malwareanalysisbook.com> 这一网站, 是可以用来分析的迹象。

## Lab 1-3

1. Upload the Lab01-03.exe file to <http://www.VirusTotal.com/>. Does it match any existing antivirus definitions?

7983a582939924c70e3da2da80fd3352ebc90de7b8c4c427d484ff4f050f0aec

61 / 71

61 security vendors and no sandboxes flagged this file as malicious

Size: 4.64 KB | Last Analysis Date: 1 day ago

peexe | fsg | overlay | runtime-modules | detect-debug-environment | long-sleeps | direct-cpu-clock-access | via-tor | checks-user-input

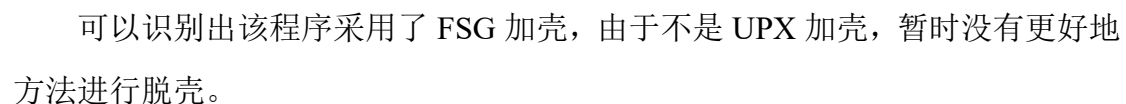
Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: [trojan:graftor/genome](#) | Threat categories: trojan, spyware | Family labels: graftor, genome, agentid

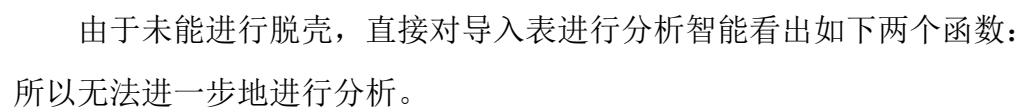
Security vendors' analysis

Vendor	Detection	Family
AhnLab-V3	Trojan:Win32/Generic.R427327	Alibabu
ALYac	Gen.Variant.Graftor/968808	Antiy-AVL
Arcabit	Trojan.Graftor.DEC8a8	Avast
AVG	Win32/Malware-gen	Baidu
BitDefender	Gen.Variant.Graftor/968808	BitDefenderTheta
Bkav Pro	W32.AIDetect/Malware	ClamAV
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	Cyberboss
		Malicious.431f46

**2. Are there any indications that this file is packed or obfuscated? If so, what are these indicators? If the file is packed, unpack it if possible.**



**3. Do any imports hint at this program's functionality? If so, which imports are they and what do they tell you?**

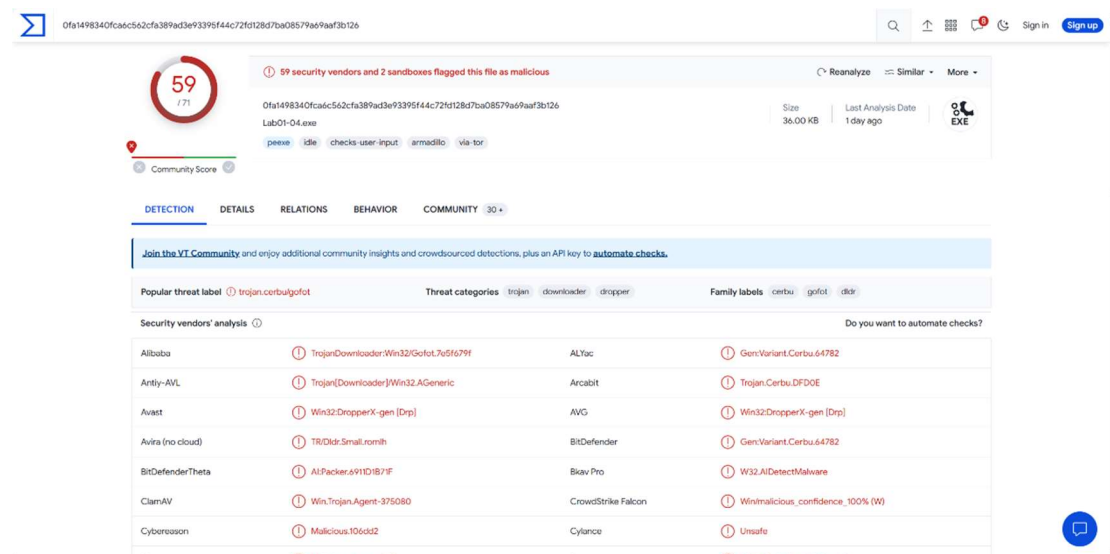


#### 4. What host- or network-based indicators could be used to identify this malware on infected machines?

由于未能进行脱壳，也没有办法分析其中的字符串。

#### Lab 1-4

#### 1. Upload the Lab01-04.exe file to <http://www.VirusTotal.com/>. Does it match any existing antivirus definitions?



59 / 71

59 security vendors and 2 sandboxes flagged this file as malicious

Ofa1498340fca6c562cfa389ad3e93395f44c72fd128d7ba08579a69aaf3b126

Lab01-04.exe

Size: 36.00 KB | Last Analysis Date: 1 day ago

peexe | ide | checks-user-input | armadillo | via-tor

Community Score

DETECTION | DETAILS | RELATIONS | BEHAVIOR | COMMUNITY 30+

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

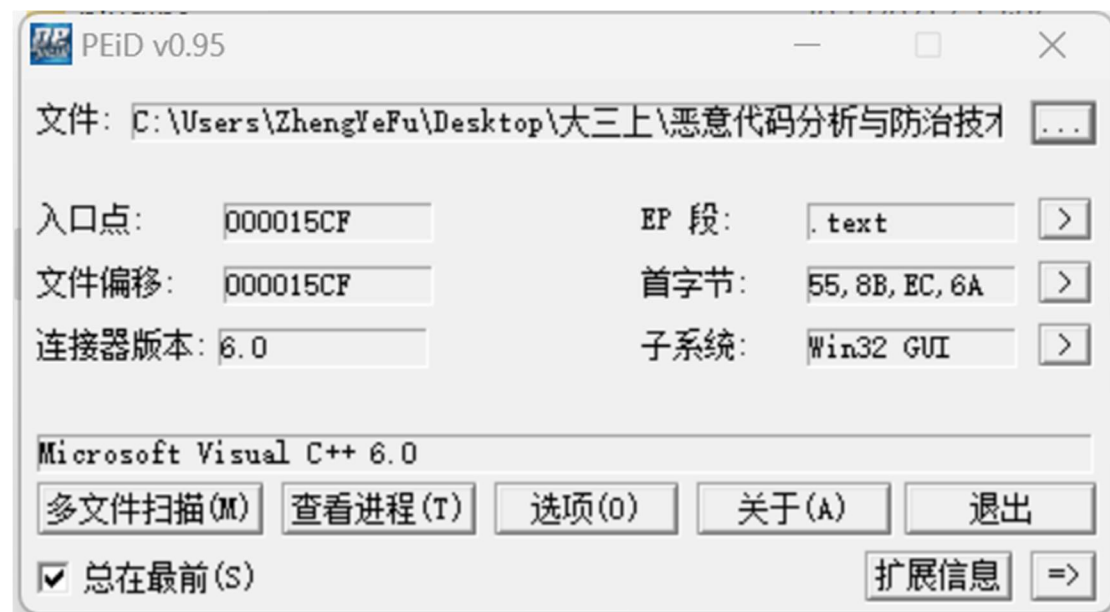
Popular threat label: trojan.cerbu.gofot | Threat categories: trojan, downloader, dropper | Family labels: cerbu, gofot, disk

Security vendors' analysis

Vendor	Detection	Category	Label
Alibaba	TrojanDownloader.Win32/Gofot.7c5f679f	ALYac	Gen-Variant.Cerbu.64782
Antiy-AVL	Trojan(Downloader)Win32.AGeneric	Arcabit	Trojan.Cerbu.DFDOE
Avast	Win32-DropperX-gen [Dps]	AVG	Win32-DropperX-gen [Dps]
Avira (no cloud)	TROJDR.Small.romh	BitDefender	Gen-Variant.Cerbu.64782
BitDefender/Theta	AI.Packer.691D1B7F	Bkav Pro	W32.AI.DetectMalware
ClamAV	Win.Trojan.Agent-375080	CrowdStrike Falcon	Win/malicious_confidence_100% (W)
Cybereason	Malicious.106d42	Cylance	Unsafe

59 家安全供应商和 2 个沙箱检测此文件为恶意文件。

#### 2. Are there any indications that this file is packed or obfuscated? If so, what are these indicators? If the file is packed, unpack it if possible.



PEiD v0.95

文件: C:\Users\ZhengYeFu\Desktop\大三上\恶意代码分析与防治技术

入口点: 000015CF | EP 段: .text

文件偏移: 000015CF | 首字节: 55, 8B, EC, 6A

连接器版本: 6.0 | 子系统: Win32 GUI

Microsoft Visual C++ 6.0

多文件扫描(M) | 查看进程(T) | 选项(O) | 关于(A) | 退出

☒ 总在最前(S) | 扩展信息(=>)

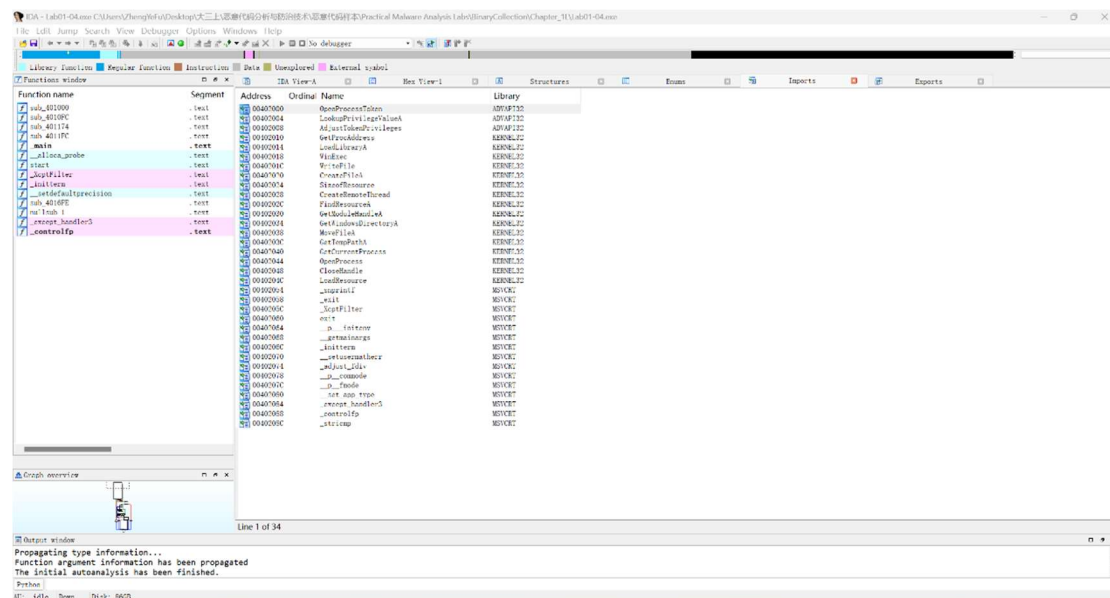
经过 PEiD 的扫描可以得知没有被加过壳

### 3. When was this program compiled?

History ⓘ	
Creation Time	2019-08-30 22:26:59 UTC
First Seen In The Wild	2011-07-05 18:16:16 UTC
First Submission	2011-07-06 00:05:42 UTC
Last Submission	2023-09-17 11:53:04 UTC
Last Analysis	2023-09-16 01:07:06 UTC

该程序的编译时间为：2019 年 8 月 30 日

**4. Do any imports hint at this program's functionality? If so, which imports are they and what do they tell you?**



该程序的导入表如下，看到了这里存在 WinExec,CreateFile,WriteFile 等函数，可以推断该程序会往磁盘上写文件，还有对文件进行移动等一系列操作。

**5. What host- or network-based indicators could be used to identify this malware on infected machines?**

```
push    offset aSystem32Wupdmgr ; "\\system32\\wupdmgr.exe"
lea     eax, [ebp+Buffer]
push    eax
```



.text:004016FE	00000018	C	3烂锰烫烫烫烫烫鮮xFF%@
.rdata:0040228E	0000000D	C	KERNEL32.dll
.rdata:004022E0	0000000D	C	ADVAPI32.dll
.rdata:004022FA	0000000B	C	MSVCRT.dll
.data:0040302C	00000011	C	SeDebugPrivilege
.data:00403040	0000000B	C	sfc os.dll
.data:0040304C	00000016	C	\\system32\\wupdmgr.exe
.data:00403064	00000005	C	%s%s
.data:00403070	00000005	C	#101
.data:00403078	00000013	C	EnumProcessModules
.data:0040308C	0000000A	C	psapi.dll
.data:00403098	00000013	C	GetModuleBaseNameA
.data:004030AC	0000000A	C	psapi.dll
.data:004030B8	0000000E	C	EnumProcesses
.data:004030C8	0000000A	C	psapi.dll
.data:004030D4	00000016	C	\\system32\\wupdmgr.exe
.data:004030EC	00000005	C	%s%s
.data:004030F4	0000000B	C	\\winup.exe
.data:00403100	00000005	C	%s%s

这一段字符串可以看出，本程序可能会往此处路径创建或修改文件。

- 6. This file has one resource in the resource section. Use Resource Hacker to examine that resource, and then use it to extract the resource. What can you learn from the resource?**

使用了 Resource Hacker 工具来存储资源文件，生成了 bin 文件，再次使用 IDA Pro 来分析该文件，查看导入表如下：

Address	Ordinal	Name	Library
00402000		WinExec	KERNEL32
00402004		GetTempPathA	KERNEL32
00402008		GetWindowsDirectoryA	KERNEL32
00402010		controlfp	MSVCRT
00402014		snprintf	MSVCRT
00402018		exit	MSVCRT
0040201C		XcptFilter	MSVCRT
00402020		exit	MSVCRT
00402024		p_initenv	MSVCRT
00402028		getmainargs	MSVCRT
0040202C		initterm	MSVCRT
00402030		setusermatherr	MSVCRT
00402034		adjust_fdiv	MSVCRT
00402038		p_commode	MSVCRT
0040203C		p_fmode	MSVCRT
00402040		set_app_type	MSVCRT
00402044		except_handler3	MSVCRT
0040204C		URLDownloadToFileA	urlmon

这里存在 URLDownloadToFile 函数，表明可能会从网络下载文件，可能会得到额外的恶意程序，并且有调用 WinExec 函数，说明还可能会执行该文件。

## Chapter\_2L

编写 Lab1 样本的 Yara 引擎规则（相关图片信息已经在 lab1 中给出，这里为了节省篇幅，不再赘述）

### lab01-01.exe

首先，在对 lab01-01.exe 中的分析结果进行分析时，我寻找了制定规则的关键要点。分析结果表明，所涉及的程序引入了来自 kernel32 库的函数，如 FindFirstFile、FindNextFile 和 CopyFile 等，这暗示了该程序可能执行文件搜索、修改等操作，具有潜在的威胁性质。不仅如此，该程序还尝试创建一个名为 kernel132.dll 的文件，并将其放置在与系统文件 kernel32.dll 位于同一路径下。这两个文件的命名方式非常相似，唯一的区别是将字母'l'替换为数字'1'，这可以被视为恶意代码试图混淆的迹象。在编写 YARA 规则时，我主要聚焦在与这些相关字符串有关的特征，其中关键点在于检测到对“kernel32.dll”的调用以及用于混淆的“kerne132.dll”的创建。

在对该 exe 文件进行进一步分析时，我发现了反汇编代码中的以下片段。根据这些发现，可以确定该 exe 文件的主要目的是运行一个动态链接库(dll)文件。这个 dll 文件导入了一些关键的系统函数，例如 sleep 和 CreateProcess。在先前的实验(lab1)中，我们已经对这个 dll 文件进行了详细分析，这部分内容包含在 1-2 部分。从这些信息可以推断，该 exe 文件可能是一个后门程序，旨在在目标系统上执行某些不正当的操作。为了有效地防止该 exe 文件的恶意行为，我们制定了规则，限制其对该 dll 文件的调用。具体而言，我定义了相关的匹配规则，以确保该 exe 文件不会执行与该 dll 文件相关的恶意操作。这些规则有助于提高系统的安全性，防止潜在的威胁。

在接下来的代码部分，即所谓的“condition”部分，我对文件的特征进行了详细的筛选和匹配。这项工作是为了确保程序在处理大文件时能够保持高效性，因为处理过大的文件可能会导致性能下降。因此，我引入了文件大小的限制，以确保只有文件大小满足一定条件时才会执行后续操作。此外，还需要验证文件是否为 PE 文件，以确保程序只处理符合特定文件格式的文件。另外，为了进一步筛选文件，我使用了字符串匹配。具体而言，程序会检查文件中是



否包含特定字符串，例如"x1"或"x2"中的任何一个。只要文件中包含其中任意一个字符串，就会触发后续的处理逻辑。由此可以得到完整的规则代码：

```
rule Lab01_01exe_Rule {  
    meta:  
        description = "Identify Lab01-01.exe"  
    strings:  
        $str1 = "C:\\Windows\\System32\\Kernel32.dll" fullword ascii  
        $str2 = "C:\\windows\\system32\\kerne132.dll" fullword ascii  
        $str3 = "Kernel32." fullword ascii  
        $str4 = "kerne132.dll" fullword ascii  
        $str5 = "Lab01-01.dll" fullword ascii  
    condition:  
        uint16(0) == 0x5a4d and  
        uint32(uint32(0x3c))==0x00004550 and  
        filesize < 50KB and 1 of ($str*) and all of them  
}
```

## lab01-01.dll

在逆向分析过程中，我使用 IDA Pro 工具对名为 lab01-01.dll 的动态链接库文件进行了分析，并寻找了相关字符串。分析结果如下：首先，在字符串搜索结果中，我发现了一个 IP 地址"127.26.152.13"的存在。然而，这个 IP 地址并非公网 IP 地址，因此可以推断出该程序不是用于网络传播攻击。因此，这个 IP 地址可以作为制定规则的一个关键信息。此外，lab01-01.dll 文件还调用了 kernel32.dll，这是 Windows 9x/Me 中非常重要的 32 位动态链接库文件，属于内核级文件。它负责系统的内存管理、数据输入输出操作以及中断处理。在逆向分析中，kernel32.dll 的使用也是一个关键的分析点。进一步分析导入表中的内容，我发现涉及到了一些与进程和 socket 相关的函数。这些函数也可以作为制定规则的关键信息。对于 condition 字段，与之前的规则相似，我可以添加文件大小匹配（文件大小为 160KB，限制在小于 500KB 内）、PE 文件判定等规则。最终，我得到了以下第二条规则：

```
rule Lab01_01dll_Rule {  
    meta:  
        description = "Lab01-01.dll"  
    strings:  
        $str1 = "SADFHUHF" fullword ascii  
        $str2 = "MSVCRT" fullword ascii  
        $str3 = "127.26.152.13" fullword ascii  
        $str4 = "WS2" fullword ascii  
    condition:  
        uint16(0) == 0x5a4d and  
        uint32(uint32(0x3c)) == 0x00004550 and  
        filesize < 500KB and all of them  
}
```

## lab01-02.exe

首先，lab01-02.exe 经过了壳层的加工。在不进行去壳处理的情况下，您所能获取的字符串信息相对有限。然而，在 Lab1 中已经完成了对该文档的去壳处理，因此现在可以着手对去壳后的文件进行详细分析。在获取了字符串相关信息后，可以进一步分析该文件的函数导入表以及其中关键的汇编代码片段。从导入的函数名称中可以观察到一些特定函数，如 `CreateService` 和 `InternetOpen` 等。结合之前字符串分析中获取的网址信息，初步推测该程序可能涉及网络操作。进一步观察与网址相关的汇编代码，我们发现该程序似乎包含了一个名为 `MalService` 的服务，并且与网站 <http://www.malwareanalysisbook.com> 建立了连接，这可能表明该程序具有分析性质。因此，这些元素构成了用于字符串分析的关键要点。值得一提的是，在字符串表中存在 `MalService` 和 `Malservice` 两种不同的写法，因此在处理 `s1` 条件时，需要添加 `nocase` 条件以表示大小写不敏感。对于 `condition` 字段，与之前的规则类似，您可以考虑加入文件大小匹配条件（例如，限制在小于 5KB 之内）以及 PE 文件的判定等规则。结合上述所有分析，我总结并编写了如下的规则：

```
rule Lab01_02exe_Rule {
```

```
meta:
    description = "Custom Lab01-02 Rule"
strings:
    $str1 = "http://w" fullword ascii
    $str2 = "CreateService" fullword ascii
    $str3 = "ADVAPI" fullword ascii
    $str4 = "MSVCRT" fullword ascii
    $str5 = "MalService" fullword nocase
    $str6 = "SystemTimeToFile" fullword ascii
    $str7 = "StartService" fullword ascii
    $str8 = "OpenSCManager" fullword ascii
    $str9 = "Internet" fullword ascii
    $str10 = "HGL345" fullword ascii
    $str11 = "Process" fullword ascii
    $str12 = "CreateWaitable" fullword ascii
condition:
    uint16(0) == 0x5a4d and
    uint32(uint32(0x3c))==0x00004550 and
    filesize < 5KB and 5 of them
}
```

### lab01-03.exe

在上个实验中已经对此程序进行了详细分析。该程序使用了 FSG 壳技术，与 UPX 壳不同，因此目前尚无更为有效的方法来去除该壳。然而，在未经去壳的情况下进行程序分析相对较为复杂，我们只能识别出部分导入表和字符串信息。所识别的字符串呈现一种相对混乱的状态，其中一部分字符串可以通过联想猜测来理解其含义，但也有一些字符串非常难以直接辨认。因此，我们在此基础上制定了一系列规则，以期能够更深入地探究恶意代码的具体目的和行为。需要注意的是，由于程序受到 FSG 壳的保护，我们在分析中可能会受到一些限制，而这些限制可能会影响我们对程序的全面理解。这也提醒我们在应对

更复杂的壳保护时，需要采用更加创新和高级的技术来解析程序，以揭示其中的潜在威胁和风险。因此，这里就按照可以被识别的字符串编写了如下规则：

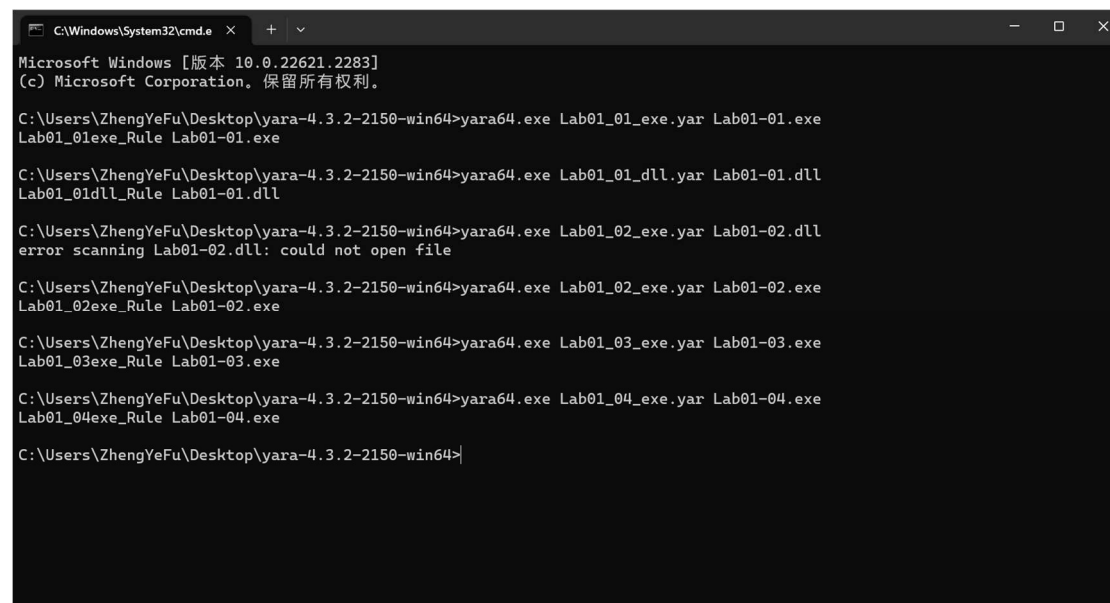
```
rule Lab01_03exe_Rule {  
    meta:  
        description = "Detect Lab01-03.exe"  
    strings:  
        $str1 = "IMSVCR71" fullword ascii  
        $str2 = "p|vuy" fullword ascii  
        $str3 = "}OLEAUTLA" fullword ascii  
        $str4 = "ole32.vd" fullword ascii  
        $str5 = "_getmas" fullword ascii  
        $str6 = "|P2r3Us" fullword ascii  
    condition:  
        uint16(0) == 0x5a4d and  
        uint32(uint32(0x3c)) == 0x00004550 and  
        filesize < 10KB and all of them  
}
```

## lab01-04.exe

在 lab1 中已经对此程序进行了初步分析，并确认它没有经过加壳处理，因此我们可以直接进行进一步分析。我们将利用 IDA Pro 工具来研究该程序的字符串和导入表。通过分析导入表，我们可以观察到程序中存在诸如 WinExec、CreateFile、WriteFile 等函数的引用。从中我们可以推断，该程序的主要功能涉及将数据写入磁盘，并可能进行文件的移动等操作。此外，通过查看程序中的字符串，我们还可以推测该程序可能会在指定路径上创建或修改文件。进一步分析时，我们使用了 Resource Hacker 工具来查看资源文件，并生成了一个二进制文件。随后，我们再次借助 IDA Pro 来分析这个生成的二进制文件，以进一步查看导入表。值得注意的是，在这个二进制文件中存在 URLDownloadToFile 函数的引用，这表明该程序可能会通过网络下载文件，从而可能获取额外的恶意程序。此外，我们还注意到程序调用了 WinExec 函数，这可能意味着它会执

行下载的文件或其他相关操作。综合上述分析，我们可以制定以下规则，以帮助识别程序的行为特征：

```
rule Lab01_04exe_Rule {  
  
    meta:  
  
        description = "Identify Lab01-04 Malicious Indicators"  
  
    strings:  
  
        $str1 = "\\system32\\wupdmgr.exe" fullword ascii  
        $str2 = "\\system32\\wupdmgrd.exe" fullword ascii  
        $str3 = "http://www.practicalmalwareanalysis.com/updater.exe" fullword ascii  
        $str4 = "\\winup.exe" fullword ascii  
        $str5 = "URLDownloadToFile" fullword ascii  
        $str6 = "WinExec" fullword ascii  
  
    condition:  
  
        uint16(0) == 0x5a4d and  
        uint32(uint32(0x3c)) == 0x00004550 and  
        filesize < 50KB and  
        5 of them  
  
}
```



```
C:\Windows\System32\cmd.e  
Microsoft Windows [版本 10.0.22621.2283]  
(c) Microsoft Corporation. 保留所有权利。  
  
C:\Users\ZhengYeFu\Desktop\yara-4.3.2-2150-win64>yara64.exe Lab01_01_exe.yar Lab01-01.exe  
Lab01_01exe_Rule Lab01-01.exe  
  
C:\Users\ZhengYeFu\Desktop\yara-4.3.2-2150-win64>yara64.exe Lab01_01_dll.yar Lab01-01.dll  
Lab01_01dll_Rule Lab01-01.dll  
  
C:\Users\ZhengYeFu\Desktop\yara-4.3.2-2150-win64>yara64.exe Lab01_02_exe.yar Lab01-02.dll  
error scanning Lab01-02.dll: could not open file  
  
C:\Users\ZhengYeFu\Desktop\yara-4.3.2-2150-win64>yara64.exe Lab01_02_exe.yar Lab01-02.exe  
Lab01_02exe_Rule Lab01-02.exe  
  
C:\Users\ZhengYeFu\Desktop\yara-4.3.2-2150-win64>yara64.exe Lab01_03_exe.yar Lab01-03.exe  
Lab01_03exe_Rule Lab01-03.exe  
  
C:\Users\ZhengYeFu\Desktop\yara-4.3.2-2150-win64>yara64.exe Lab01_04_exe.yar Lab01-04.exe  
Lab01_04exe_Rule Lab01-04.exe  
  
C:\Users\ZhengYeFu\Desktop\yara-4.3.2-2150-win64>
```

## 四、实验结论及心得体会

在本次实验中，广泛应用了多项工具，用以深入分析恶意软件代码。这些工具的选择涵盖了我们之前在相关先修课程（如汇编语言与逆向技术以及软件安全课程）中已经接触过的，例如 IDA Pro 和 OllyDbg 等工具，以及一些全新的工具，例如 PE Explorer 和 Resource Hacker 等。我认为对这些工具的熟练掌握对于深入学习病毒分析等领域至关重要。此外，分析过程中，我也着重学习了加壳和脱壳技术，以及对导入表和字符串的深入分析。这些技术的掌握不仅有助于更好地理解恶意代码的运行机制，还有助于提高对恶意软件攻击的识别和应对能力。因此，这些学习过程在我们的实验中具有重要的地位，为我提供了深入了解恶意软件的机会，同时也有助于提升我在软件安全领域的专业技能。同时，在这次 YARA 规则编写和恶意软件检测实验中，我深入了解了 YARA 工具的应用。我学习了如何编写自定义的 YARA 规则，掌握了规则测试的过程，并认识到 YARA 在恶意软件分析和威胁情报领域的重要性。通过这次实验，我提高了问题排除和调试的能力，我希望将来能进一步研究和应用 YARA 来提高网络和系统的安全性。