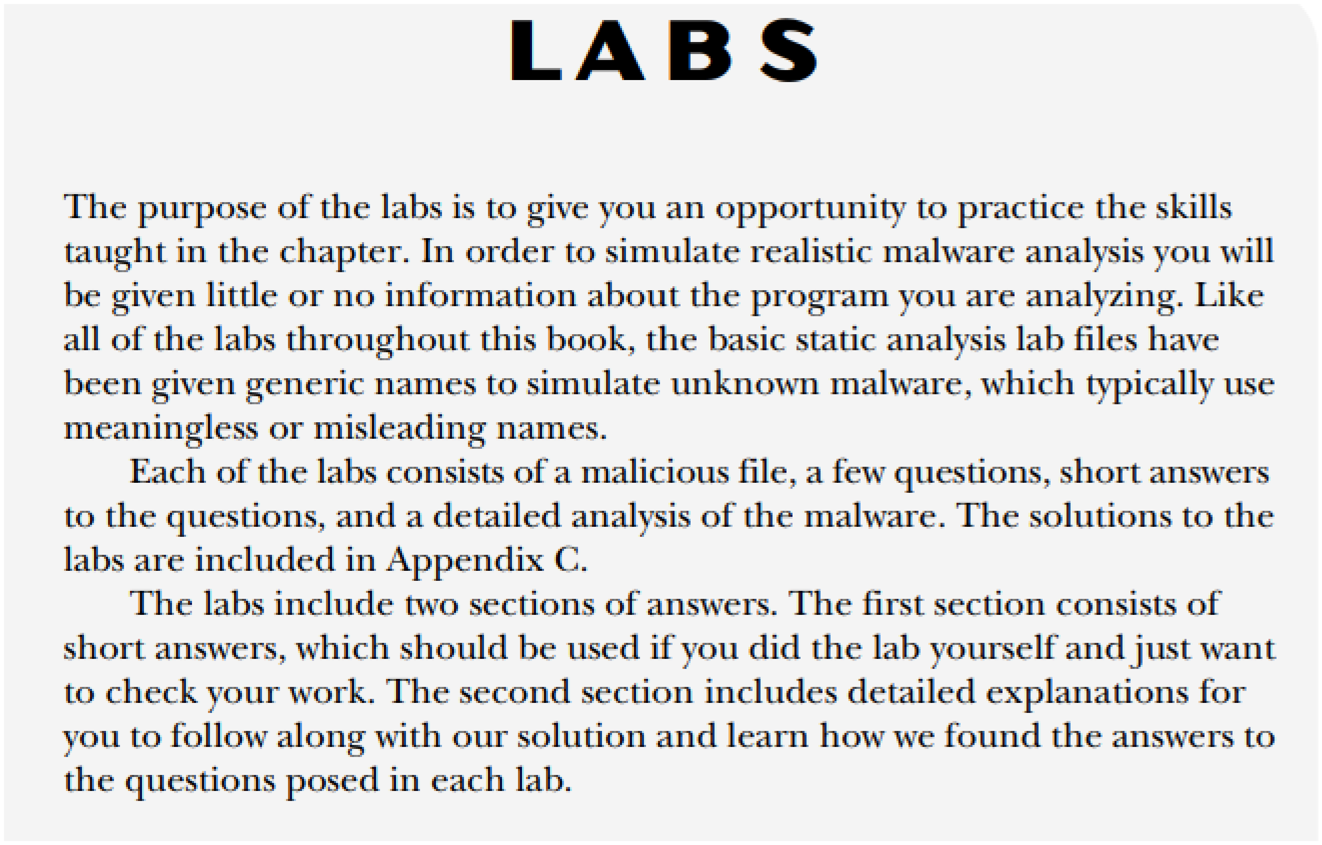
Open in app Get started

Kamran Saifullah Follow

Aug 29, 2019 · 4 min read · Listen

Save


LABS

The purpose of the labs is to give you an opportunity to practice the skills taught in the chapter. In order to simulate realistic malware analysis you will be given little or no information about the program you are analyzing. Like all of the labs throughout this book, the basic static analysis lab files have been given generic names to simulate unknown malware, which typically use meaningless or misleading names.

Each of the labs consists of a malicious file, a few questions, short answers to the questions, and a detailed analysis of the malware. The solutions to the labs are included in Appendix C.

The labs include two sections of answers. The first section consists of short answers, which should be used if you did the lab yourself and just want to check your work. The second section includes detailed explanations for you to follow along with our solution and learn how we found the answers to the questions posed in each lab.

Practical Malware Analysis — Book

Practical Malware Analysis — Chapter 1 — Labs 1–1 — Solution

As we are done with the Chapter-1. It's time to work on the labs to get most out of our learning. So let's begin.

> *Note:* I have copied the Labs Details (text) from the book as it is.

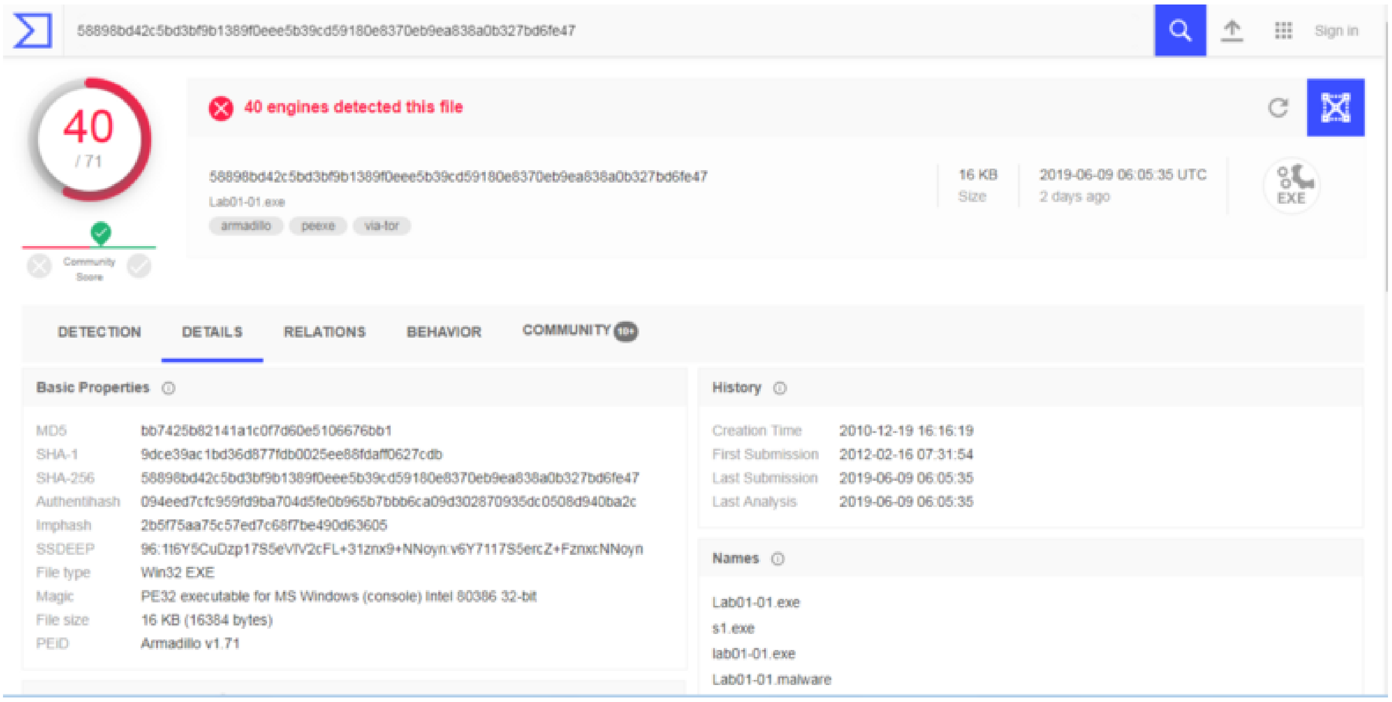
Open in app Get started

answer the questions below.

Questions

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

Let's upload both of the files on VirusTotal and tally the result!



58898bd42c5bd3bf9b1389f0eee5b39cd59180e8370eb9ea838a0b327bd6fe47

40 engines detected this file

58898bd42c5bd3bf9b1389f0eee5b39cd59180e8370eb9ea838a0b327bd6fe47
Lab01-01.exe
armadillo peexe via-tor

16 KB Size
2019-06-09 06:05:35 UTC 2 days ago
EXE

Basic Properties

MD5	bb7425b82141a1c0f7d60e5106676bb1
SHA-1	9dce39ac1bd36d877fdb0025ee88fdaff0627cdb
SHA-256	58898bd42c5bd3bf9b1389f0eee5b39cd59180e8370eb9ea838a0b327bd6fe47
Authentihash	094eed7cfc959fd9ba704d5fe0b965b7bbb6ca09d302870935dc0508d940ba2c
Imphash	2b5f75aa75c57ed7c68f7be490d63605
SSDEEP	96:1t6Y5CuDzp17S5eVlV2cFL+31znx9+NNoyn:v6Y7117S5ercZ+FznxcNNoyn
File type	Win32 EXE
Magic	PE32 executable for MS Windows (console) Intel 80386 32-bit
File size	16 KB (16384 bytes)
PEiD	Armadillo v1.71

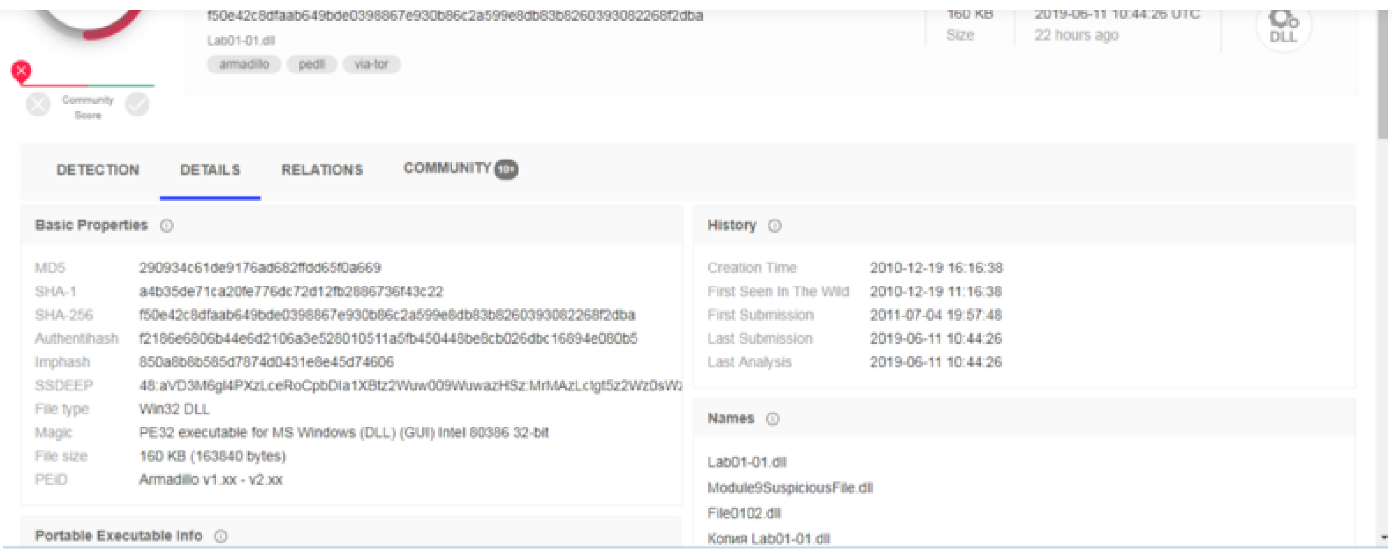
History

Creation Time	2010-12-19 16:16:19
First Submission	2012-02-16 07:31:54
Last Submission	2019-06-09 06:05:35
Last Analysis	2019-06-09 06:05:35

Names

Lab01-01.exe
s1.exe
lab01-01.exe
Lab01-01.malware

Result of Lab01–01.exe



f50e42c8dfaab649bde0398867e930b86c2a599e8db83b8260393082268f2dba
Lab01-01.dll
armadillo pedll via-tor

160 KB
Size

2019-06-11 10:44:26 UTC
22 hours ago

Basic Properties

MD5	290934c61de9176ad682ffdd65f0a669
SHA-1	a4b35de71ca20fe776dc72d12fb2886736f43c22
SHA-256	f50e42c8dfaab649bde0398867e930b86c2a599e8db83b8260393082268f2dba
Authentihash	f2186e6806b44e6d2106a3e528010511a5fb450448be8cb026dbc16894e080b5
Imphash	850a8b8b585d7874d0431e8e45d74606
SSDEEP	48:aVD3M6gl4PXzLceRoCpbDIa1XBtz2Wuw009WuwazHSz:MrMAzLctgt5z2Wz0sW
File type	Win32 DLL
Magic	PE32 executable for MS Windows (DLL) (GUI) Intel 80386 32-bit
File size	160 KB (163840 bytes)
PEiD	Armadillo v1.xx - v2.xx

History

Creation Time	2010-12-19 16:16:38
First Seen In The Wild	2010-12-19 11:16:38
First Submission	2011-07-04 19:57:48
Last Submission	2019-06-11 10:44:26
Last Analysis	2019-06-11 10:44:26

Names

Lab01-01.dll
Module9SuspiciousFile.dll
File0102.dll
Копия Lab01-01.dll

Portable Executable Info

Result of — Lab01–01.dll

We can clearly see that these files have been matched with the previously known signatures and have also been detected as malicious.

2. When were these files compiled?

The compilation time of both file as per the report of VirusTotal is.

Lab01–01.exe → 2010–12–19 16:16:19

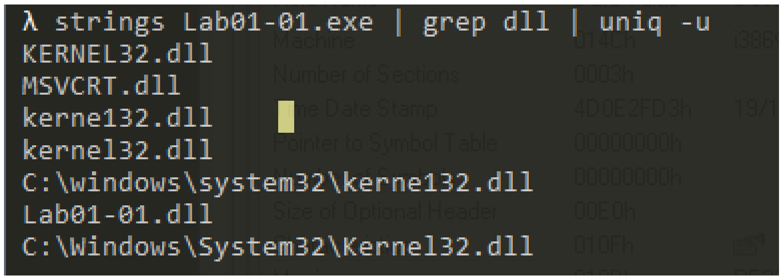
Lab01–01.dll → 2010–12–19 16:16:38

We can also find the compilation time using PEview and checking the IMAGE_FILE_HEADER details.

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

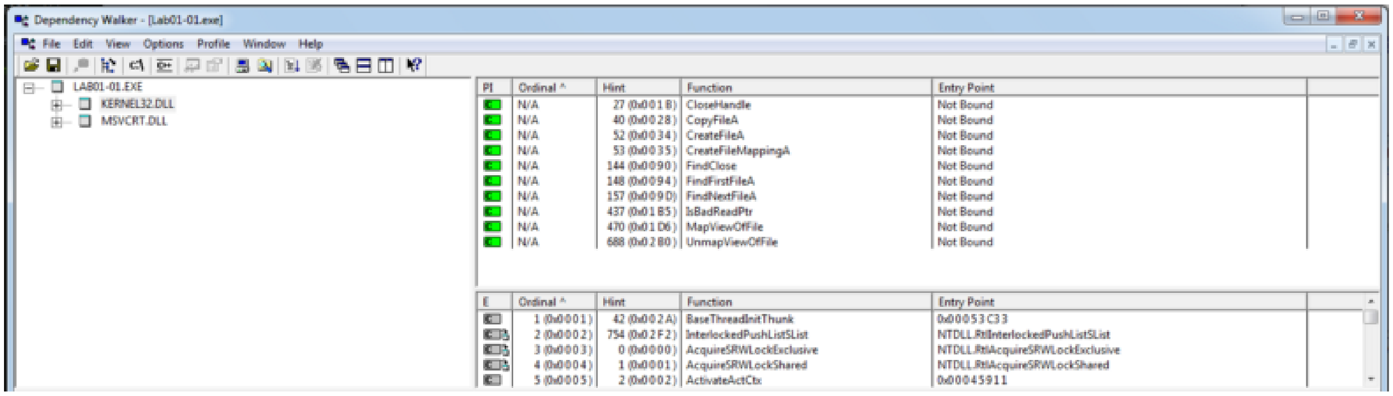
PEiD can be used to find the packed or obfuscated file although we were able to find all the necessary details and the strings. So we conclude that both of these files were not been packed or obfuscated.

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



```
λ strings Lab01-01.exe | grep dll | uniq -u
KERNEL32.dll
MSVCRT.dll
kerne132.dll
kernel32.dll
C:\windows\system32\kerne132.dll
Lab01-01.dll
C:\Windows\System32\Kernel32.dll
```

a. KERNEL32.dll — Following functions from this library were called.



Dependency Walker - [Lab01-01.exe]

```
LAB01-01.EXE
  KERNEL32.DLL
  MSVCRT.DLL
```

PI	Ordinal ^	Hint	Function	Entry Point
🟩	N/A	27 (0x001B)	CloseHandle	Not Bound
🟩	N/A	40 (0x0028)	CopyFileA	Not Bound
🟩	N/A	52 (0x0034)	CreateFileA	Not Bound
🟩	N/A	53 (0x0035)	CreateFileMappingA	Not Bound
🟩	N/A	144 (0x0090)	FindClose	Not Bound
🟩	N/A	148 (0x0094)	FindFirstFileA	Not Bound
🟩	N/A	157 (0x009D)	FindNextFileA	Not Bound
🟩	N/A	437 (0x01B5)	IsBadReadPtr	Not Bound
🟩	N/A	470 (0x01D6)	MapViewOfFile	Not Bound
🟩	N/A	688 (0x02B0)	UnmapViewOfFile	Not Bound

E	Ordinal ^	Hint	Function	Entry Point
	1 (0x0001)	42 (0x002A)	BaseThreadInitThunk	0x00053C33
	2 (0x0002)	754 (0x02F2)	InterlockedPushListSList	NTDLL.RtlInterlockedPushListSList
	3 (0x0003)	0 (0x0000)	AcquireSRWLockExclusive	NTDLL.RtlAcquireSRWLockExclusive
	4 (0x0004)	1 (0x0001)	AcquireSRWLockShared	NTDLL.RtlAcquireSRWLockShared
	5 (0x0005)	2 (0x0002)	ActivateActCtx	0x00045911

CreateFileA → Creates or opens a file or I/O device.

CopyFileA → Copies an existing file to a new file.

CreateFileMappingA → Creates or opens a named or unnamed file mapping object for a specified file.

FindFirstFileA → Searches a directory for a file or subdirectory with a name that matches a specific name (or partial name if wildcards are used).

FindNextFileA → Continues a file search from a previous call.

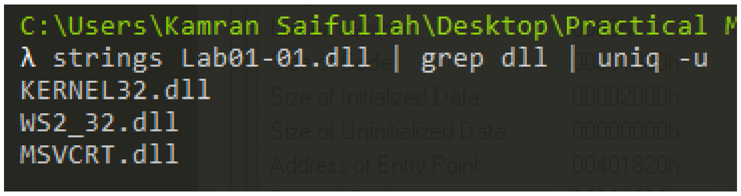
MapViewOfFile → Maps a view of a file mapping into the address space of a calling process. Malware can make changes to the actual file once it is mapped.

b. MSVCRT.dll → A module containing standard C library functions such as printf,

c. kernel132.dll → Disguised version of original KERNEL32.DLL.

d. Lab01–01.dll → Additional DLL file created for the successful working of Lab01–01.exe executable.

The second file Lab01–01.dll do the following imports.



```
C:\Users\Kamran Saifullah\Desktop\Practical M
λ strings Lab01-01.dll | grep dll | uniq -u
KERNEL32.dll
WS2_32.dll
MSVCRT.dll
```

KERNEL32.dll → Kernel32.dll is the 32-bit dynamic link library found in the Windows operating system kernel. It handles memory management, input/output operations, and interrupts. When Windows boots up, kernel32.dll is loaded into a protected memory space so other applications do not take that space over.

MSVCRT → A module containing standard C library functions such as printf, memcpy, and cos. It is a part of the Microsoft C Runtime Library. Non-system processes like msvcrt.dll originate from software you installed on your system.

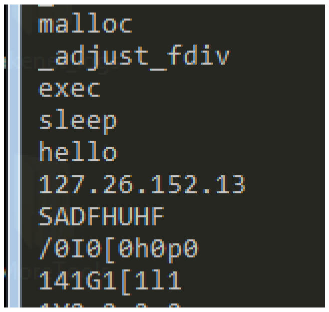
WS2_32.dll → The Windows Sockets Library ws2_32.dll, is required by windows and applications to handle network connections.

5. Are there any other files or host-based indicators that you could look for on infected systems?

While finding the strings we found that there is another file named as “Kerne132.dll” which is supposed to be disguised as the “Kernel32.dll”. Also there is another “Lab01–01.DLL” which is not a common OS DLL. So we can look for these files on the system.

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

over this IP address.



```
malloc
_adjust_fdiv
exec
sleep
hello
127.26.152.13
SADFHUHF
/0I0[0h0p0
141G1[1l1
```

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

On bringing up all the pieces together we can assume that Lab01–01.exe along with the extension Lab01–01.dll is a malware which creates a backdoor and connects to a C&C server and transfer the critical information. Secondly both of the files are not packed and Lab01–01.exe searches in and from directories and look for a particular files and replaces them with disguised files. Also it imports functions from core KERNEL32.DLL and network based imports to establish the connections. Also uses the exec function which means that it would be executing some other programs/files along with sleep function which waits until a particular statement or piece of code gets executed. This is mostly used in backdoors.