

Malware Analysis

Project Archive - Final

Final Report

Ethan Berman

Ryan Bharat

Jenny Chang

Fall 2022

CS467 - Online Capstone Project

Table of Contents

Introduction	3
Research Goals	3
Experimental Approaches/Setup	3
Experimental Results	4
Basic Static Analysis	4
Basic Dynamic Analysis	4
Advanced Static Analysis	5
Advanced Dynamic Analysis	5
Malware Sample 1 - Lab06-01.exe from Practical Malware Analysis	5
Malware Sample 2 - Lab07-03.exe from Practical Malware Analysis	6
Conclusion	8
References	9

Introduction

We have all had the dreaded experience of downloading an unknown file, or following a nefarious link, only to find our computer infected with malware. Some have had the experience of encountering a situation in which a virus scan from an antivirus software does not provide the antidote needed to rectify the issue. In difficult situations like this, rather than focus on solely resolving the malware issue, what if instead we aimed to analyze the malware infection and attempt to understand its intent? In addition to analyzing the procedure which the malware adheres to, what if we were to compare various malware tools with those that are more modern and robust, to find which are best suited for analysis? Given this premise, we sought to set up a secure malware sandbox and observe malware files with various analysis tools - those outdated and unsupported and others which are more modern and robust - in order to study the function of the malicious files and how each affects the operating system.

Research Goals

The goals of this research project are to set up a malware lab, perform static and dynamic analysis on malware samples, and document the process and findings.

Experimental Approaches/Setup

Note: The following are general instructions, as each step may vary depending on software preference and/or operating system type/version.

1. Download relevant VMware/VirtualBox software (pertaining to your system)
2. Create two separate virtual environments
 - a. Download the .iso files of the desired operating system(s) you want to install on each virtual environment
3. At this point, you may want to either download the desired malware analysis tools to both virtual machines, or drag/drop or copy/paste them from your host machine, as this is more difficult once we begin isolating the virtual environments.
4. After the creation of each virtual environment based on their respective .iso files, and adding the desired analysis software, our purpose is to isolate each virtual environment from the host machine.
 - a. If possible, remove any drivers that are not needed for testing your desired malware files (i.e. usb/bluetooth, cd/dvd drive, sound card)
 - b. Disable any antivirus (i.e. windows defender) to be able to test the malware without interference
 - c. Ensure that the network adapters of your virtual machines are set to 'internal network' or whichever setting allows network communication between your virtual machines, but isolated from the host machine.
 - i. To test this, you can ping the ip address from one virtual machine to the other to ensure you get a reply, ensuring proper connection. Additionally, you can ping both of the virtual machine's ip addresses from the host machine to ensure that the requests timeout.
 - d. Ensure that isolation settings such as 'drag and drop' and 'copy and paste' are disabled to further ensure there is no communication between host and virtual machine(s).

At this point, our virtual machines should be isolated from our host machine, and we can begin dissecting malware files with no consequence to our host machine.

Experimental Results

Basic Static Analysis

The first step in analyzing a sample of malware is to examine the properties of the executable without closely examining the code. This step is commonly referred to as basic static analysis. The goal of basic static analysis is to determine whether a file is truly malicious and gain insight into its functionality without executing it. This includes but is not limited to looking at a program's strings, checking if the program's code has been obfuscated, and examining the file's function imports and exports.

PeStudio and VirusTotal are two tools that can be used to perform basic static analysis. PeStudio is a free analysis program that provides a wealth of information about an executable's properties. PeStudio will extract a file's strings, indicate if obfuscation is present, mark functions and strings as suspicious, and much more. An important property when statically analyzing a file is looking at its hash. An executable's hash can be searched online and in malware databases such as VirusTotal for more information. VirusTotal is a useful tool for static analysis as it provides just as much information as PeStudio along with the results for that executable in a variety of different antivirus engines. VirusTotal provides additional insight such as relations to other potentially malicious files and IP addresses as well as sandbox reports.

Typically, the first step when analyzing malware is to look up its file hash in VirusTotal to see if it has already been analyzed. Users can upload files to VirusTotal for analysis, however, all files uploaded to the website can potentially be seen and downloaded by others. Advanced attackers may opt to employ custom malware that is designed for a particular target, therefore, when a malware analyst uploads that sample to VirusTotal, they can potentially alert the attackers to having been discovered. Additionally, there may be instances where files uploaded are not actually malware and could contain sensitive information. If a PDF document potentially contained malware but also had sensitive medical information, it would generally be unwise to upload this to VirusTotal since it would be a breach of certain privacy laws.

Basic Dynamic Analysis

Basic dynamic analysis looks at processes, file system, registry keys, network activities during and after malware execution. The two types of analysis tools we explored are local tools and online sandbox.

Process Monitor, Regshot, and Wireshark are local tools that should only be used on an isolated virtual machine due to the risks of infection from running malware. First, Process Monitor is a tool to capture processes and operations that occur during and after malware execution. Next, Regshot is a tool to compare registry key changes using snapshots taken before and after malware execution. Lastly, Wireshark is a tool to monitor network activities. The advantage of these local tools is that malware stays on the local machine, which is good for companies who may not want malware and private company information to become public or known to third-party vendors. However, the disadvantage is the inconvenience of having to download and operate separate tools in order for a comprehensive analysis.

Any.Run is an online sandbox that can be used on any machine with internet access. A user simply uploads a URL or file and runs the analysis. It can generate a text report listing processes, registry activities, files activities, and network activities. The advantage of Any.Run is that it is an all-in-one tool that can perform a full analysis without separate tools. The process is quick and simple because a malware lab setup is not required. There is no risk to the host machine and network. One disadvantage of Any.Run is that it requires an account using a business or school email. Also, the free edition only offers Windows 7 32 bit for its environment OS, so it may not work for malware designed for another OS. Another

disadvantage is that malware becomes known to the public or third-party once uploaded online, so Any.Run is not a good tool for companies who wish to keep malware or company information private.

Advanced Static Analysis

While basic static and basic dynamic analysis can provide a high level overview of the purpose of a malware file, it fails to provide in depth information about the detailed procedure which the infected file follows. At this point, we need to have a general understanding of assembly and disassembly in order to understand how malware functions at a more precise level. Advanced static analysis is the process of using a disassembler to better view the malware files assembly code. There are a plethora of tools used for advanced static analysis, but I will be highlighting Ollydbg and Malcat with the intention of showcasing the differences between the two tools. Ollydbg has been a popular malware disassembler dating back to the early 2000's and is no longer being developed, while Malcat is in its beta phase, actively contributed to, and is a modern disassembler that is used by IT professionals.

When comparing both Ollydbg and Malcat on installation alone, Ollydbg shows its age. Ollydbg will only work on Windows machines and can work properly on machines back to Windows 95. Malcat on the other hand, is only supported on Windows 7 (64 bit) and above. Malcat also supports various Linux distributions such as Debian and Ubuntu. Overall, if the need is to test operating systems prior to Windows 7, Ollydbg would be the best, while if your systems run on anything newer than Windows 7 or Linux, then Malcat would be the choice.

Advanced Dynamic Analysis

Advanced dynamic analysis refers to the use of a debugger to step through an executable's assembly instructions. Two debuggers to highlight are x32dbg and x64dbg. They are very similar but are intended to debug 32-bit and 64-bit executables, respectively. Debugging malware gives us more control over the behavior. Malware authors typically employ a variety of system checks to prevent certain code from executing. The Russian-speaking hacker group REvil authored their malware to avoid systems that use Russian and related languages [1]. Malware can also perform checks to see if it is being run in a virtual machine or if there is network connectivity. We typically analyze malware in an isolated virtual environment so in order to get the code to execute, we essentially need to trick the malware into executing its malicious code. Debuggers present us with the opportunity to edit register values, patch executables, and understand at a very granular level what the program is doing. By editing the values at runtime and patching executables we can get around many anti-analysis techniques employed by malware authors.

Malware Sample 1 - Lab06-01.exe from *Practical Malware Analysis*

The first malware sample is Lab06-01.exe from *Practical Malware Analysis* [2]. To begin, Pestudio shows the malware is a 32-bit Windows console program compiled on January 31, 2011. The strings "Error 1.1: No internet" and "Success: Internet Connection" suggest that the malware checks for internet connection. In addition, the library section flags wininet.dll as suspicious since this library is used to make internet connections. The imports section flags 6 functions as suspicious, including functions that check for internet connected state, write to file, terminate process, and retrieve environment information.

		imports (44)	flag (6)
library (2)	flag (1)	GetEnvironmentVariableA	x
		InternetGetConnectedState	x
wininet.dll	x	WriteFile	x
kernel32.dll	-	TerminateProcess	x
		GetEnvironmentStrings	x
		GetEnvironmentStringsW	x

Figure 1. Pestudio flags libraries and imports of Lab06-01.exe.

Next, Process Monitor shows the malware manipulates the file system and registry. Also, Regshot confirms registry key changes after the malware execution.

Time	Process Name	PID	Operation	Path	Result	Detail
10:09...	Lab06-01.exe	3572	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS	Image Base: 0x...
10:09...	Lab06-01.exe	3572	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS	Image Base: 0x...
10:09...	Lab06-01.exe	3572	CreateFile	C:\Windows\Prefetch\Lab06-01.EXE-1...	SUCCESS	Desired Access:...
10:09...	Lab06-01.exe	3572	QueryStandard...	C:\Windows\Prefetch\Lab06-01.EXE-1...	SUCCESS	AllocationSize:...
10:09...	Lab06-01.exe	3572	ReadFile	C:\Windows\Prefetch\Lab06-01.EXE-1...	SUCCESS	Offset: 0, Lengt...
10:09...	Lab06-01.exe	3572	CloseFile	C:\Windows\Prefetch\Lab06-01.EXE-1...	SUCCESS	
10:09...	Lab06-01.exe	3572	RegOpenKey	HKLM\System\CurrentControlSet\Contr...	REPARSE	Desired Access:...
10:09...	Lab06-01.exe	3572	RegOpenKey	HKLM\System\CurrentControlSet\Contr...	SUCCESS	Desired Access:...
10:09...	Lab06-01.exe	3572	RegQueryValue	HKLM\System\CurrentControlSet\Contr...	NAME NOT FOUND	Length: 80

Figure 2. Process Monitor shows processes and operations from running Lab06-01.exe.

Keys deleted: 1 Keys added: 5 Values added: 6 Values deleted: 10 Values modified: 39 Folders deleted: 0 Folders added: 0 Folders attributes changed: 0 Files deleted: 0 Files added: 0 Files (attributes) modified: 0 Total changes: 61	Keys deleted: 1 ----- HKLM\SOFTWARE\Microsoft\Provisioning\Sessions\F0GeKcdovE++PYo9.0 ----- Keys added: 5 ----- HKLM\SOFTWARE\Microsoft\Provisioning\Sessions\wb5mKLoziUC8x0HF.0
--	---

Figure 3. Regshot shows registry key changes from running Lab06-01.exe.

Lastly, Ollydbg indicates the malware calls a function at 00401000, which then calls WININET.InternetGetConnectedState. The result is stored as 0 or 1 in EAX. If the result is 0, ASCII values “Error 1.1: No Internet” are pushed and printed to the command line. If the result is 1, ASCII values “Success: Internet Connection” are pushed and printed to the command line.

The screenshot shows the Ollydbg interface. The assembly window displays the following code:

```

00401000 55      PUSH EBP
00401001 8BEC    MOV EBP, ESP
00401002 51      PUSH ECX
00401003 6A 00   PUSH 0
00401004 6A 00   PUSH 0
00401005 FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
00401006 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
00401007 837D FC CMP DWORD PTR SS:[EBP-4], 0
00401008 74 14   JE SHORT Lab06-01.0040102B
00401009 68 48704000 PUSH Lab06-01.00407048
0040100A E8 3E000000 CALL Lab06-01.0040105F
0040100B 83C4 04 ADD ESP, 4
0040100C B8 01000000 MOV EAX, 1
0040100D 5E      POP ESI
0040100E 5D      POP EDI
0040100F 5C      POP EBX
00401010 5B      POP EAX
00401011 5A      POP ECX
00401012 59      POP EAX
00401013 58      POP EAX
00401014 57      POP EAX
00401015 56      POP EAX
00401016 55      PUSH EBP
00401017 8BEC    MOV EBP, ESP
00401018 51      PUSH ECX
00401019 6A 00   PUSH 0
0040101A 6A 00   PUSH 0
0040101B FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
0040101C 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
0040101D 837D FC CMP DWORD PTR SS:[EBP-4], 0
0040101E 74 14   JE SHORT Lab06-01.0040102B
0040101F 68 48704000 PUSH Lab06-01.00407048
00401020 E8 3E000000 CALL Lab06-01.0040105F
00401021 83C4 04 ADD ESP, 4
00401022 B8 01000000 MOV EAX, 1
00401023 5E      POP ESI
00401024 5D      POP EDI
00401025 5C      POP EBX
00401026 5B      POP EAX
00401027 5A      POP ECX
00401028 59      POP EAX
00401029 58      POP EAX
0040102A 57      POP EAX
0040102B 56      POP EAX
0040102C 55      PUSH EBP
0040102D 8BEC    MOV EBP, ESP
0040102E 51      PUSH ECX
0040102F 6A 00   PUSH 0
00401030 6A 00   PUSH 0
00401031 FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
00401032 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
00401033 837D FC CMP DWORD PTR SS:[EBP-4], 0
00401034 74 14   JE SHORT Lab06-01.0040102B
00401035 68 48704000 PUSH Lab06-01.00407048
00401036 E8 3E000000 CALL Lab06-01.0040105F
00401037 83C4 04 ADD ESP, 4
00401038 B8 01000000 MOV EAX, 1
00401039 5E      POP ESI
0040103A 5D      POP EDI
0040103B 5C      POP EBX
0040103C 5B      POP EAX
0040103D 5A      POP ECX
0040103E 59      POP EAX
0040103F 58      POP EAX
00401040 57      POP EAX
00401041 56      POP EAX
00401042 55      PUSH EBP
00401043 8BEC    MOV EBP, ESP
00401044 51      PUSH ECX
00401045 6A 00   PUSH 0
00401046 6A 00   PUSH 0
00401047 FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
00401048 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
00401049 837D FC CMP DWORD PTR SS:[EBP-4], 0
0040104A 74 14   JE SHORT Lab06-01.0040102B
0040104B 68 48704000 PUSH Lab06-01.00407048
0040104C E8 3E000000 CALL Lab06-01.0040105F
0040104D 83C4 04 ADD ESP, 4
0040104E B8 01000000 MOV EAX, 1
0040104F 5E      POP ESI
00401050 5D      POP EDI
00401051 5C      POP EBX
00401052 5B      POP EAX
00401053 5A      POP ECX
00401054 59      POP EAX
00401055 58      POP EAX
00401056 57      POP EAX
00401057 56      POP EAX
00401058 55      PUSH EBP
00401059 8BEC    MOV EBP, ESP
0040105A 51      PUSH ECX
0040105B 6A 00   PUSH 0
0040105C 6A 00   PUSH 0
0040105D FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
0040105E 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
0040105F 837D FC CMP DWORD PTR SS:[EBP-4], 0
00401060 74 14   JE SHORT Lab06-01.0040102B
00401061 68 48704000 PUSH Lab06-01.00407048
00401062 E8 3E000000 CALL Lab06-01.0040105F
00401063 83C4 04 ADD ESP, 4
00401064 B8 01000000 MOV EAX, 1
00401065 5E      POP ESI
00401066 5D      POP EDI
00401067 5C      POP EBX
00401068 5B      POP EAX
00401069 5A      POP ECX
0040106A 59      POP EAX
0040106B 58      POP EAX
0040106C 57      POP EAX
0040106D 56      POP EAX
0040106E 55      PUSH EBP
0040106F 8BEC    MOV EBP, ESP
00401070 51      PUSH ECX
00401071 6A 00   PUSH 0
00401072 6A 00   PUSH 0
00401073 FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
00401074 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
00401075 837D FC CMP DWORD PTR SS:[EBP-4], 0
00401076 74 14   JE SHORT Lab06-01.0040102B
00401077 68 48704000 PUSH Lab06-01.00407048
00401078 E8 3E000000 CALL Lab06-01.0040105F
00401079 83C4 04 ADD ESP, 4
0040107A B8 01000000 MOV EAX, 1
0040107B 5E      POP ESI
0040107C 5D      POP EDI
0040107D 5C      POP EBX
0040107E 5B      POP EAX
0040107F 5A      POP ECX
00401080 59      POP EAX
00401081 58      POP EAX
00401082 57      POP EAX
00401083 56      POP EAX
00401084 55      PUSH EBP
00401085 8BEC    MOV EBP, ESP
00401086 51      PUSH ECX
00401087 6A 00   PUSH 0
00401088 6A 00   PUSH 0
00401089 FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
0040108A 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
0040108B 837D FC CMP DWORD PTR SS:[EBP-4], 0
0040108C 74 14   JE SHORT Lab06-01.0040102B
0040108D 68 48704000 PUSH Lab06-01.00407048
0040108E E8 3E000000 CALL Lab06-01.0040105F
0040108F 83C4 04 ADD ESP, 4
00401090 B8 01000000 MOV EAX, 1
00401091 5E      POP ESI
00401092 5D      POP EDI
00401093 5C      POP EBX
00401094 5B      POP EAX
00401095 5A      POP ECX
00401096 59      POP EAX
00401097 58      POP EAX
00401098 57      POP EAX
00401099 56      POP EAX
0040109A 55      PUSH EBP
0040109B 8BEC    MOV EBP, ESP
0040109C 51      PUSH ECX
0040109D 6A 00   PUSH 0
0040109E 6A 00   PUSH 0
0040109F FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
004010A0 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
004010A1 837D FC CMP DWORD PTR SS:[EBP-4], 0
004010A2 74 14   JE SHORT Lab06-01.0040102B
004010A3 68 48704000 PUSH Lab06-01.00407048
004010A4 E8 3E000000 CALL Lab06-01.0040105F
004010A5 83C4 04 ADD ESP, 4
004010A6 B8 01000000 MOV EAX, 1
004010A7 5E      POP ESI
004010A8 5D      POP EDI
004010A9 5C      POP EBX
004010AA 5B      POP EAX
004010AB 5A      POP ECX
004010AC 59      POP EAX
004010AD 58      POP EAX
004010AE 57      POP EAX
004010AF 56      POP EAX
004010B0 55      PUSH EBP
004010B1 8BEC    MOV EBP, ESP
004010B2 51      PUSH ECX
004010B3 6A 00   PUSH 0
004010B4 6A 00   PUSH 0
004010B5 FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
004010B6 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
004010B7 837D FC CMP DWORD PTR SS:[EBP-4], 0
004010B8 74 14   JE SHORT Lab06-01.0040102B
004010B9 68 48704000 PUSH Lab06-01.00407048
004010BA E8 3E000000 CALL Lab06-01.0040105F
004010BB 83C4 04 ADD ESP, 4
004010BC B8 01000000 MOV EAX, 1
004010BD 5E      POP ESI
004010BE 5D      POP EDI
004010BF 5C      POP EBX
004010C0 5B      POP EAX
004010C1 5A      POP ECX
004010C2 59      POP EAX
004010C3 58      POP EAX
004010C4 57      POP EAX
004010C5 56      POP EAX
004010C6 55      PUSH EBP
004010C7 8BEC    MOV EBP, ESP
004010C8 51      PUSH ECX
004010C9 6A 00   PUSH 0
004010CA 6A 00   PUSH 0
004010CB FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
004010CC 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
004010CD 837D FC CMP DWORD PTR SS:[EBP-4], 0
004010CE 74 14   JE SHORT Lab06-01.0040102B
004010CF 68 48704000 PUSH Lab06-01.00407048
004010D0 E8 3E000000 CALL Lab06-01.0040105F
004010D1 83C4 04 ADD ESP, 4
004010D2 B8 01000000 MOV EAX, 1
004010D3 5E      POP ESI
004010D4 5D      POP EDI
004010D5 5C      POP EBX
004010D6 5B      POP EAX
004010D7 5A      POP ECX
004010D8 59      POP EAX
004010D9 58      POP EAX
004010DA 57      POP EAX
004010DB 56      POP EAX
004010DC 55      PUSH EBP
004010DD 8BEC    MOV EBP, ESP
004010DE 51      PUSH ECX
004010DF 6A 00   PUSH 0
004010E0 6A 00   PUSH 0
004010E1 FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
004010E2 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
004010E3 837D FC CMP DWORD PTR SS:[EBP-4], 0
004010E4 74 14   JE SHORT Lab06-01.0040102B
004010E5 68 48704000 PUSH Lab06-01.00407048
004010E6 E8 3E000000 CALL Lab06-01.0040105F
004010E7 83C4 04 ADD ESP, 4
004010E8 B8 01000000 MOV EAX, 1
004010E9 5E      POP ESI
004010EA 5D      POP EDI
004010EB 5C      POP EBX
004010EC 5B      POP EAX
004010ED 5A      POP ECX
004010EE 59      POP EAX
004010EF 58      POP EAX
004010F0 57      POP EAX
004010F1 56      POP EAX
004010F2 55      PUSH EBP
004010F3 8BEC    MOV EBP, ESP
004010F4 51      PUSH ECX
004010F5 6A 00   PUSH 0
004010F6 6A 00   PUSH 0
004010F7 FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
004010F8 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
004010F9 837D FC CMP DWORD PTR SS:[EBP-4], 0
004010FA 74 14   JE SHORT Lab06-01.0040102B
004010FB 68 48704000 PUSH Lab06-01.00407048
004010FC E8 3E000000 CALL Lab06-01.0040105F
004010FD 83C4 04 ADD ESP, 4
004010FE B8 01000000 MOV EAX, 1
004010FF 5E      POP ESI
00401100 5D      POP EDI
00401101 5C      POP EBX
00401102 5B      POP EAX
00401103 5A      POP ECX
00401104 59      POP EAX
00401105 58      POP EAX
00401106 57      POP EAX
00401107 56      POP EAX
00401108 55      PUSH EBP
00401109 8BEC    MOV EBP, ESP
0040110A 51      PUSH ECX
0040110B 6A 00   PUSH 0
0040110C 6A 00   PUSH 0
0040110D FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
0040110E 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
0040110F 837D FC CMP DWORD PTR SS:[EBP-4], 0
00401110 74 14   JE SHORT Lab06-01.0040102B
00401111 68 48704000 PUSH Lab06-01.00407048
00401112 E8 3E000000 CALL Lab06-01.0040105F
00401113 83C4 04 ADD ESP, 4
00401114 B8 01000000 MOV EAX, 1
00401115 5E      POP ESI
00401116 5D      POP EDI
00401117 5C      POP EBX
00401118 5B      POP EAX
00401119 5A      POP ECX
0040111A 59      POP EAX
0040111B 58      POP EAX
0040111C 57      POP EAX
0040111D 56      POP EAX
0040111E 55      PUSH EBP
0040111F 8BEC    MOV EBP, ESP
00401120 51      PUSH ECX
00401121 6A 00   PUSH 0
00401122 6A 00   PUSH 0
00401123 FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
00401124 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
00401125 837D FC CMP DWORD PTR SS:[EBP-4], 0
00401126 74 14   JE SHORT Lab06-01.0040102B
00401127 68 48704000 PUSH Lab06-01.00407048
00401128 E8 3E000000 CALL Lab06-01.0040105F
00401129 83C4 04 ADD ESP, 4
0040112A B8 01000000 MOV EAX, 1
0040112B 5E      POP ESI
0040112C 5D      POP EDI
0040112D 5C      POP EBX
0040112E 5B      POP EAX
0040112F 5A      POP ECX
00401130 59      POP EAX
00401131 58      POP EAX
00401132 57      POP EAX
00401133 56      POP EAX
00401134 55      PUSH EBP
00401135 8BEC    MOV EBP, ESP
00401136 51      PUSH ECX
00401137 6A 00   PUSH 0
00401138 6A 00   PUSH 0
00401139 FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
0040113A 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
0040113B 837D FC CMP DWORD PTR SS:[EBP-4], 0
0040113C 74 14   JE SHORT Lab06-01.0040102B
0040113D 68 48704000 PUSH Lab06-01.00407048
0040113E E8 3E000000 CALL Lab06-01.0040105F
0040113F 83C4 04 ADD ESP, 4
00401140 B8 01000000 MOV EAX, 1
00401141 5E      POP ESI
00401142 5D      POP EDI
00401143 5C      POP EBX
00401144 5B      POP EAX
00401145 5A      POP ECX
00401146 59      POP EAX
00401147 58      POP EAX
00401148 57      POP EAX
00401149 56      POP EAX
0040114A 55      PUSH EBP
0040114B 8BEC    MOV EBP, ESP
0040114C 51      PUSH ECX
0040114D 6A 00   PUSH 0
0040114E 6A 00   PUSH 0
0040114F FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
00401150 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
00401151 837D FC CMP DWORD PTR SS:[EBP-4], 0
00401152 74 14   JE SHORT Lab06-01.0040102B
00401153 68 48704000 PUSH Lab06-01.00407048
00401154 E8 3E000000 CALL Lab06-01.0040105F
00401155 83C4 04 ADD ESP, 4
00401156 B8 01000000 MOV EAX, 1
00401157 5E      POP ESI
00401158 5D      POP EDI
00401159 5C      POP EBX
0040115A 5B      POP EAX
0040115B 5A      POP ECX
0040115C 59      POP EAX
0040115D 58      POP EAX
0040115E 57      POP EAX
0040115F 56      POP EAX
00401160 55      PUSH EBP
00401161 8BEC    MOV EBP, ESP
00401162 51      PUSH ECX
00401163 6A 00   PUSH 0
00401164 6A 00   PUSH 0
00401165 FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
00401166 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
00401167 837D FC CMP DWORD PTR SS:[EBP-4], 0
00401168 74 14   JE SHORT Lab06-01.0040102B
00401169 68 48704000 PUSH Lab06-01.00407048
0040116A E8 3E000000 CALL Lab06-01.0040105F
0040116B 83C4 04 ADD ESP, 4
0040116C B8 01000000 MOV EAX, 1
0040116D 5E      POP ESI
0040116E 5D      POP EDI
0040116F 5C      POP EBX
00401170 5B      POP EAX
00401171 5A      POP ECX
00401172 59      POP EAX
00401173 58      POP EAX
00401174 57      POP EAX
00401175 56      POP EAX
00401176 55      PUSH EBP
00401177 8BEC    MOV EBP, ESP
00401178 51      PUSH ECX
00401179 6A 00   PUSH 0
0040117A 6A 00   PUSH 0
0040117B FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
0040117C 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
0040117D 837D FC CMP DWORD PTR SS:[EBP-4], 0
0040117E 74 14   JE SHORT Lab06-01.0040102B
0040117F 68 48704000 PUSH Lab06-01.00407048
00401180 E8 3E000000 CALL Lab06-01.0040105F
00401181 83C4 04 ADD ESP, 4
00401182 B8 01000000 MOV EAX, 1
00401183 5E      POP ESI
00401184 5D      POP EDI
00401185 5C      POP EBX
00401186 5B      POP EAX
00401187 5A      POP ECX
00401188 59      POP EAX
00401189 58      POP EAX
0040118A 57      POP EAX
0040118B 56      POP EAX
0040118C 55      PUSH EBP
0040118D 8BEC    MOV EBP, ESP
0040118E 51      PUSH ECX
0040118F 6A 00   PUSH 0
00401190 6A 00   PUSH 0
00401191 FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
00401192 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
00401193 837D FC CMP DWORD PTR SS:[EBP-4], 0
00401194 74 14   JE SHORT Lab06-01.0040102B
00401195 68 48704000 PUSH Lab06-01.00407048
00401196 E8 3E000000 CALL Lab06-01.0040105F
00401197 83C4 04 ADD ESP, 4
00401198 B8 01000000 MOV EAX, 1
00401199 5E      POP ESI
0040119A 5D      POP EDI
0040119B 5C      POP EBX
0040119C 5B      POP EAX
0040119D 5A      POP ECX
0040119E 59      POP EAX
0040119F 58      POP EAX
004011A0 57      POP EAX
004011A1 56      POP EAX
004011A2 55      PUSH EBP
004011A3 8BEC    MOV EBP, ESP
004011A4 51      PUSH ECX
004011A5 6A 00   PUSH 0
004011A6 6A 00   PUSH 0
004011A7 FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
004011A8 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
004011A9 837D FC CMP DWORD PTR SS:[EBP-4], 0
004011AA 74 14   JE SHORT Lab06-01.0040102B
004011AB 68 48704000 PUSH Lab06-01.00407048
004011AC E8 3E000000 CALL Lab06-01.0040105F
004011AD 83C4 04 ADD ESP, 4
004011AE B8 01000000 MOV EAX, 1
004011AF 5E      POP ESI
004011B0 5D      POP EDI
004011B1 5C      POP EBX
004011B2 5B      POP EAX
004011B3 5A      POP ECX
004011B4 59      POP EAX
004011B5 58      POP EAX
004011B6 57      POP EAX
004011B7 56      POP EAX
004011B8 55      PUSH EBP
004011B9 8BEC    MOV EBP, ESP
004011BA 51      PUSH ECX
004011BB 6A 00   PUSH 0
004011BC 6A 00   PUSH 0
004011BD FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
004011BE 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
004011BF 837D FC CMP DWORD PTR SS:[EBP-4], 0
004011C0 74 14   JE SHORT Lab06-01.0040102B
004011C1 68 48704000 PUSH Lab06-01.00407048
004011C2 E8 3E000000 CALL Lab06-01.0040105F
004011C3 83C4 04 ADD ESP, 4
004011C4 B8 01000000 MOV EAX, 1
004011C5 5E      POP ESI
004011C6 5D      POP EDI
004011C7 5C      POP EBX
004011C8 5B      POP EAX
004011C9 5A      POP ECX
004011CA 59      POP EAX
004011CB 58      POP EAX
004011CC 57      POP EAX
004011CD 56      POP EAX
004011CE 55      PUSH EBP
004011CF 8BEC    MOV EBP, ESP
004011D0 51      PUSH ECX
004011D1 6A 00   PUSH 0
004011D2 6A 00   PUSH 0
004011D3 FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
004011D4 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
004011D5 837D FC CMP DWORD PTR SS:[EBP-4], 0
004011D6 74 14   JE SHORT Lab06-01.0040102B
004011D7 68 48704000 PUSH Lab06-01.00407048
004011D8 E8 3E000000 CALL Lab06-01.0040105F
004011D9 83C4 04 ADD ESP, 4
004011DA B8 01000000 MOV EAX, 1
004011DB 5E      POP ESI
004011DC 5D      POP EDI
004011DD 5C      POP EBX
004011DE 5B      POP EAX
004011DF 5A      POP ECX
004011E0 59      POP EAX
004011E1 58      POP EAX
004011E2 57      POP EAX
004011E3 56      POP EAX
004011E4 55      PUSH EBP
004011E5 8BEC    MOV EBP, ESP
004011E6 51      PUSH ECX
004011E7 6A 00   PUSH 0
004011E8 6A 00   PUSH 0
004011E9 FF15 80604000 CALL DWORD PTR DS:[<WININET.InternetGet
004011EA 8945 FC MOV DWORD PTR SS:[EBP-4], EAX
004011EB 837D FC CMP DWORD PTR SS:[EBP-4], 0
004011EC 74 14   JE SHORT Lab06-01.0040102B
004011ED 68 48704000 PUSH Lab06-01.00407048
004011EE E8 3E000000 CALL Lab06-01.0040105F
004011EF 83C4 04 ADD ESP, 4
004011F0 B8 01000000 MOV EAX, 1
004011F1 5E      POP ESI
004011F2 5D      POP EDI
004011F3 5C      POP EBX
004011F4 5B      POP EAX
004011F5 5A      POP ECX
004011F6 59      POP EAX
004011F7 58      POP EAX
004011F8 57      POP EAX
004011F9 56      POP EAX
004011FA 55      PUSH EBP
004011FB 8BEC    MOV EBP, ESP
004011FC 51      PUSH ECX
004011FD 6A 00   PUSH 0
004011FE 6A 00   PUSH 0
004011FF FF15
```

This causes some concern, as this can lead the one doing the analysis to believe that the purpose of this executable is to not only create a new, unknown Lab07-03.dll file, but also to replace the typical kernel32.dll file with a nefarious kernel132.dll file in the system32 directory.

When analyzing the function calls and the ASCII values in the CPU pane, it is clear that the intention of the .exe file is to replace the usual kernel32.dll file with kernel132.dll. This is shown via the string compare function `_stricmp` (Figure 6) to find the known kernel32.dll file.

0040116C	75 67	JNZ SHORT Lab07-03.00401105	
0040116E	58 20304000	PUSH Lab07-03.00403020	
00401173	53	PUSH EBX	
00401174	FF15 64204000	CALL DWORD PTR DS:[<&MSVCRT._stricmp>]	<code>s2 = "kernel32.dll"</code>
0040117A	83C4 08	ADD ESP,8	<code>s1</code>
0040117D	5EC0	TEST EAX,EAX	<code>_stricmp</code>
0040117F	75 26	JNZ SHORT Lab07-03.004011A7	
00401181	8BFB	MOV EDI,EBX	
00401183	83C9 FF	OR ECX,FFFFFFFF	
00401186	F21AE	REPNE SCAS BYTE PTR ES:[EDI]	
00401188	F7D1	NOT ECX	
0040118A	8BC1	MOV EAX,ECX	
0040118C	BE 10304000	MOV ESI,Lab07-03.00403010	ASCII "kernel132.dll"
00401191	8BFB	MOV EDI,EBX	
00401193	C1E9 02	SHR ECX,2	
00401196	F31A5	REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:[ESI]	
00401198	8BC8	MOV ECX,EAX	
0040119A	83E1 03	AND ECX,3	
0040119D	F31A4	REP MOVS BYTE PTR ES:[EDI],BYTE PTR DS:[ESI]	
0040119F	8B7424 10	MOV ESI,DWORD PTR SS:[ESP+10]	
004011A3	8B7C24 20	MOV EDI,DWORD PTR SS:[ESP+20]	
004011A7	83C7 14	ADD EDI,14	
004011AA	EB 96	JMP SHORT Lab07-03.00401142	

Figure 6. Disassembly showing the function call to compare strings

Further in the assembly code, we can find the creation of the file and file mapping for the kernel132.dll file. As expected, the file is created with `GENERIC_ALL` access, which allows full access rights (Figure 7).

30401495	8B3D 14204000	MOV EDI,DWORD PTR DS:[<&KERNEL32.CreateFileA>]	<code>kernel32.CreateFileA</code>
30401498	50	PUSH EAX	<code>hTemplateFile</code>
3040149C	50	PUSH EAX	<code>Attributes</code>
3040149D	6A 03	PUSH 3	<code>Mode = OPEN_EXISTING</code>
3040149F	50	PUSH EAX	<code>pSecurity</code>
304014A0	6A 01	PUSH 1	<code>ShareMode = FILE_SHARE_READ</code>
304014A2	68 00000000	PUSH 00000000	<code>Access = GENERIC_READ</code>
304014A7	68 8C304000	PUSH Lab07-03.0040308C	<code>FileName = "C:\Windows\System32\Kernel132.dll"</code>
304014AC	FFD7	CALL EDI	<code>CreateFileA</code>
304014AE	8B1D 10204000	MOV EBX,DWORD PTR DS:[<&KERNEL32.CreateFileMappingA>]	<code>kernel32.CreateFileMappingA</code>
304014B4	6A 00	PUSH 0	<code>hMapFile = NULL</code>
304014B6	6A 00	PUSH 0	<code>MaximumSizeLow = 0</code>
304014B8	6A 00	PUSH 0	<code>MaximumSizeHigh = 0</code>
304014BA	6A 02	PUSH 2	<code>Protection = PAGE_READONLY</code>
304014BC	6A 00	PUSH 0	<code>pSecurity = NULL</code>
304014BE	50	PUSH EAX	<code>hFile</code>
304014BF	894424 64	MOV DWORD PTR SS:[ESP+64],EAX	
304014C3	FFD3	CALL EBX	<code>CreateFileMappingA</code>
304014C5	8B2D 0C204000	MOV EBP,DWORD PTR DS:[<&KERNEL32.MapViewOfFile>]	<code>kernel32.MapViewOfFile</code>
304014CB	6A 00	PUSH 0	<code>MapSize = 0</code>
304014CD	6A 00	PUSH 0	<code>OffsetLow = 0</code>
304014CF	6A 00	PUSH 0	<code>OffsetHigh = 0</code>
304014D1	6A 04	PUSH 4	<code>AccessMode = FILE_MAP_READ</code>
304014D3	50	PUSH EAX	<code>hMapObject</code>
304014D4	FFD5	CALL EBP	<code>MapViewOfFile</code>
304014D6	6A 00	PUSH 0	<code>hTemplateFile = NULL</code>
304014D8	6A 00	PUSH 0	<code>Attributes = 0</code>
304014DA	6A 03	PUSH 3	<code>Mode = OPEN_EXISTING</code>
304014DC	6A 00	PUSH 0	<code>pSecurity = NULL</code>
304014DE	6A 01	PUSH 1	<code>ShareMode = FILE_SHARE_READ</code>
304014E0	8BFA	MOV ESI,EAX	
304014E2	68 00000000	PUSH 00000000	
304014E7	68 7C304000	PUSH Lab07-03.0040307C	<code>Access = GENERIC_ALL</code>
304014EC	897424 74	MOV DWORD PTR SS:[ESP+74],ESI	<code>FileName = "Lab07-03.dll"</code>
304014F0	FFD7	CALL EDI	<code>CreateFileA</code>

Figure 7. Disassembly showing the creation of a file and file mapping for suspicious kernel132.dll

From here, by observing the call tree, there is a reference to the procedure `<Lab07-03.Sleep>` (Figure 8) which gives some insight as to what the nefarious DLL file's function is. Upon inspecting the imports, with a tool like Process Hacker, of the provided Lab07-03.dll file, it is clear that its purpose is to either create a new process via `exec`, or to sleep.

References

- [1] A. Zemlianichenko, "Code in huge ransomware attack written to avoid computers that use Russian, says New report," *NBCNews.com*, 07-Jul-2021. [Online]. Available: <https://www.nbcnews.com/politics/national-security/code-huge-ransomware-attack-written-avoid-computers-use-russian-says-n1273222>. [Accessed: 12-Nov-2022].
- [2] M. Sikorski and A. Honig, *Practical Malware Analysis*. San Francisco, CA: No Starch Press, 2012.