**CSE 434S – Reverse Engineering and Malware Analysis**

**HW1 - Basic Static Analysis**

Download the archive file for this assignment from Canvas and extract the archive to a virtual machine (the archive password is “infected”). The archive contains Sample1 and Sample2, which are malware samples. **Do NOT run them!**

The lecture slides and Chapter 1 and Appendix A in the textbook, as well as the slides on Windows API calls and the Resource Hacker program posted to the assignment Canvas page, will be helpful for completing the assignment.

**Important instructions for your writeup:**

* Use this document to answer the questions.
* Explain your answers and add repeatable notes as you did in the similar lab exercise.
* Take screenshots when needed, but note that screenshots cannot replace your words.
* A good explanation doesn’t necessarily mean a lengthy one. Be concise!
* Assume that the reader doesn’t have the book or our slides. It doesn’t mean that you need to explain the theory, but it means that you cannot answer a question by saying “I followed instructions on page 3 and slide 15”.
* Whenever possible, describe your steps in first person.

**Part 1: Sample1**

1) What is the MD5 of Sample1? (2 pts)

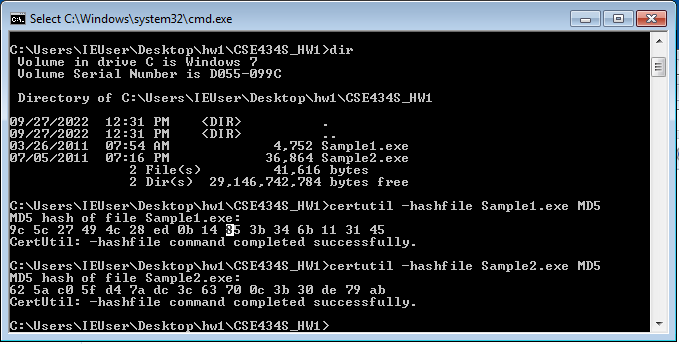
--------------------------------------------------------------------------------------------

C:\Users\IEUser\Desktop\hw1\CSE434S\_HW1>certutil -hashfile Sample1.exe MD5

MD5 hash of file Sample1.exe:

9c 5c 27 49 4c 28 ed 0b 14 85 3b 34 6b 11 31 45

CertUtil: -hashfile command completed successfully.



--------------------------------------------------------------------------------------------

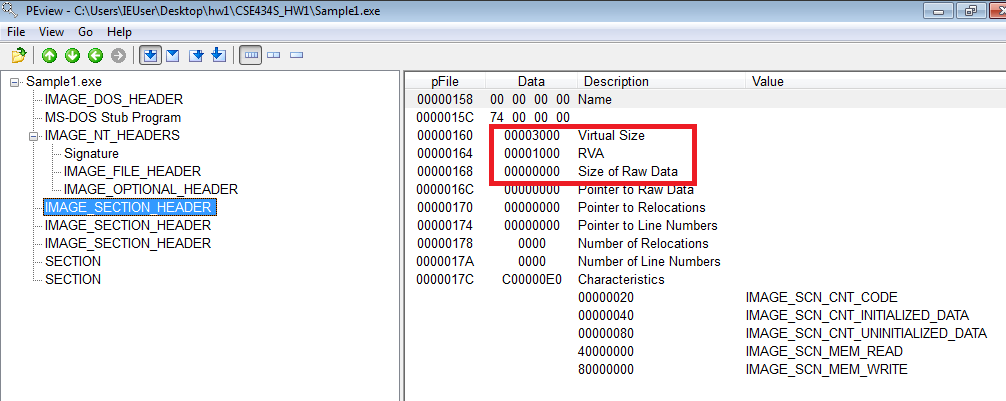
2) Is Sample1 packed? List three indicators to justify your answer. If packed, what packer was used to pack Sample2? (5 pts)

--------------------------------------------------------------------------------------------

Yes, it is packed! And the packer is FSG 1.0 -> dulek/xt.

I found 3 indicators by using PeiD and Peview.

The first indicator is that Virtual Size is much larger than Size of Raw data. In fact Size of Raw data is zero!

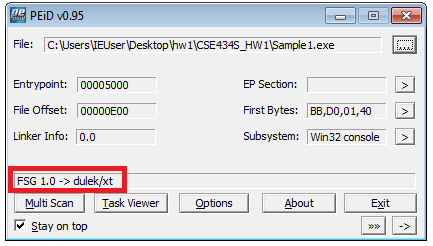


Second indicator is that there was no common PE sections for .text, .rdata, .data, .rsrc.

텍스트이(가) 표시된 사진

자동 생성된 설명

And finally, I can see the packer’s name when I opened the file with PeiD. The pakcer is FSG 1.0 -> dulek/xt.



--------------------------------------------------------------------------------------------

3) If needed, unpack Sample1 and describe how you unpacked it. What is the md5 of the unpacked file? (3 pts)

--------------------------------------------------------------------------------------------

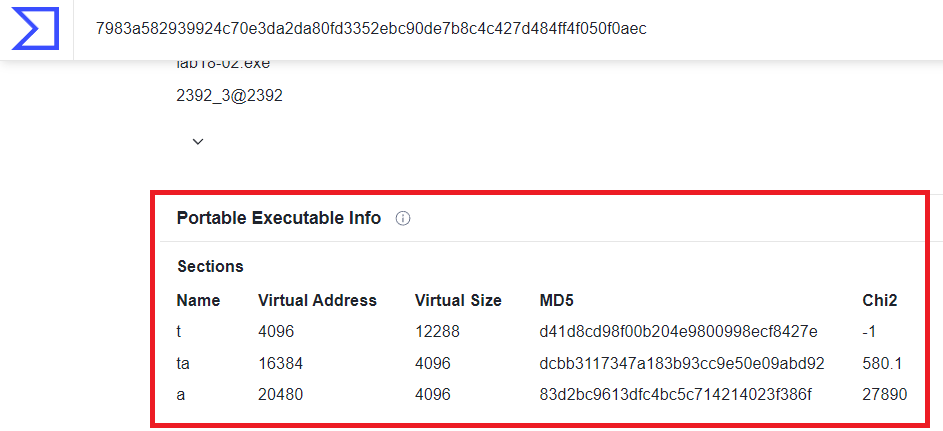
The file is packed using FSG 1.0 -> dulek/xt and I haven’t learn how to unpack it. Of course, I don’t know the md5 checksum of the file.

--------------------------------------------------------------------------------------------

4) What is the exact date and time (to the second) that Sample1 was compiled? (2 pts)

-------------------------------------------------------------------------------------------

I cannot find out the compile time because the file is packed. And I also analyze the file using Virus total, but there was no compilation time.



--------------------------------------------------------------------------------------------

5) Investigate the Windows API functions that Sample1 imports. List two functions that suggest that Sample1 can check whether it is being debugged. How can a malware Sample use each of these functions to check whether it is being debugged? (5 pts)

--------------------------------------------------------------------------------------------

First, Sample1 imports two functions. ‘LoadLibraryA’ ‘GetProcAddress’. With these two imports, I can tell that Sample1 uses Runtime Linking which is common in malware.

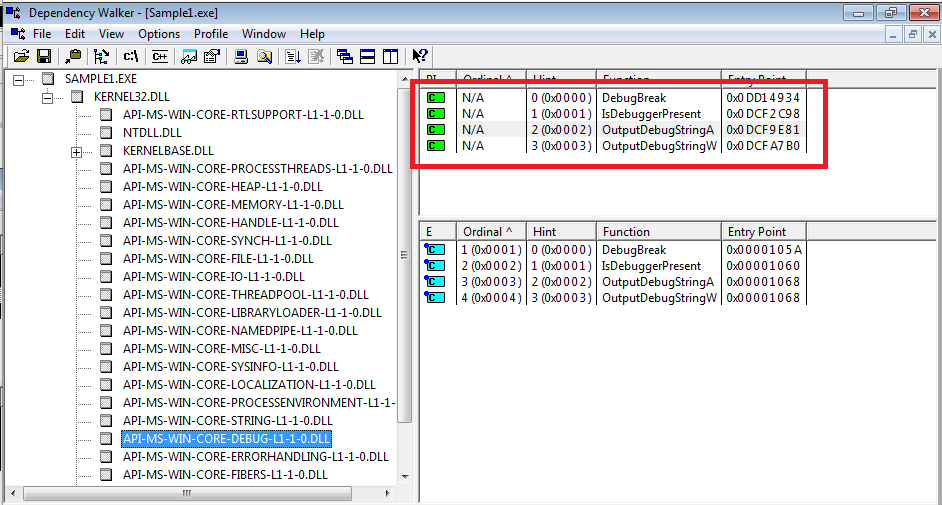
테이블이(가) 표시된 사진

자동 생성된 설명

And it imports several functions about debugging. I can see the list when I looked it up using Dependency walker.

**IsDebuggerPresent, OutputDebugString**

Lastly, I think IsDebuggerPresent is a function which can tell whether the program is being debugged or not. Maybe, Sample1 calls this function to check the status of being debugged or not, so it can act differently if it is in a debug situation.



--------------------------------------------------------------------------------------------

6) Which section in Sample1 contains executable code? What is the virtual size of this section? (5 pts)

--------------------------------------------------------------------------------------------

Usually, the executable code’s location is in Common PE Sections but the file is packed. Unless unpacking the file, it is hard to find out which section is.

--------------------------------------------------------------------------------------------

7) Do any of Sample1’s imports suggest that it is able to connect to the internet? Do any of the strings in Sample1 suggest that it is able to connect to the internet? Why is this suspicious? (10 pts)

--------------------------------------------------------------------------------------------

Unless unpacking the file, it is hard to find out Sample1 uses internet connection or not. And I cannot find strings look like an IP address as well. Since it is packed, I looked through the whole strings in a file, I couldn’t find anything suspicious because all of those strings looked meaningless.

--------------------------------------------------------------------------------------------

8) Can you say what this malware does? You will be graded for providing educated conclusions based on your findings, and not necessarily for the correctness of your conclusions. (3 points)

--------------------------------------------------------------------------------------------

Since it is packed and I don’t know how to unpack this file at the moment, I have to say the I’m not sure about what it does. And also, I can see only two functions imported, LoadLibraryA, GetProcAddress. It is a sign of Runtime Linking and is common in malware program. I only assume the program will call libraries runtime basis.

--------------------------------------------------------------------------------------------

**Part 2: Sample2**

1) What is the MD5 of Sample2? (2 pts)

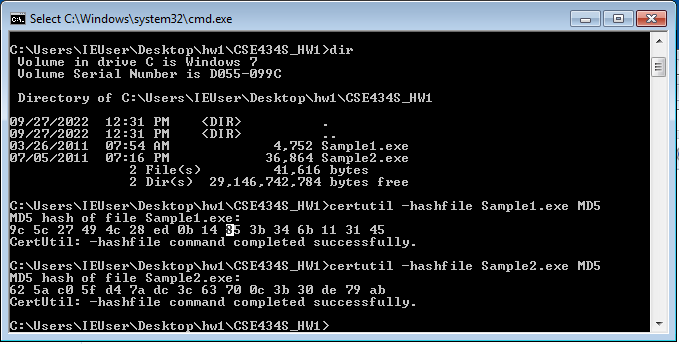
--------------------------------------------------------------------------------------------

C:\Users\IEUser\Desktop\hw1\CSE434S\_HW1>certutil -hashfile Sample2.exe MD5

MD5 hash of file Sample2.exe:

62 5a c0 5f d4 7a dc 3c 63 70 0c 3b 30 de 79 ab

CertUtil: -hashfile command completed successfully.

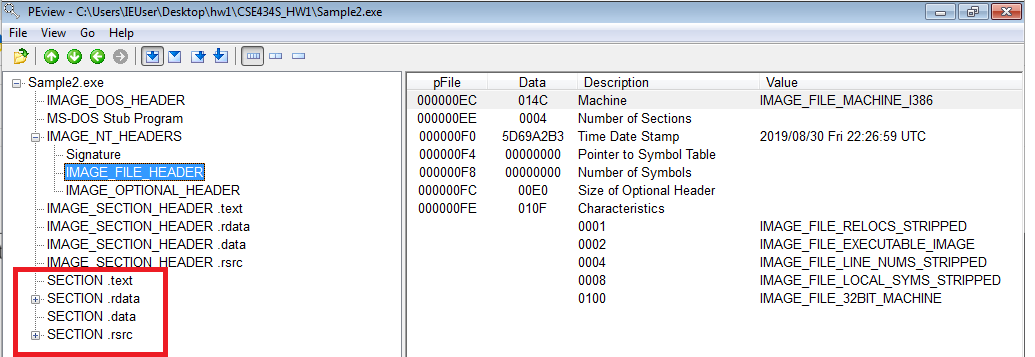


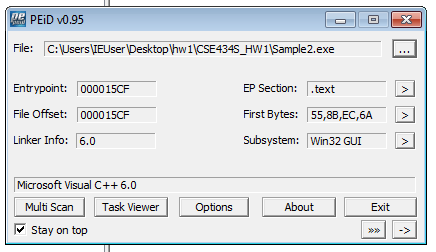
--------------------------------------------------------------------------------------------

2) Is Sample 2 packed? List three indicators to justify your answer. If packed, what packer was used to pack Sample2? (5 pts)

--------------------------------------------------------------------------------------------

I don’t think it’s packed. I can see the SECTION HEADER clearly using PEView and there was no packing information with PEiD.





--------------------------------------------------------------------------------------------

3) If needed, unpack Sample2 and describe how you unpacked it. What is the md5 of the unpacked file? (3 pts)

--------------------------------------------------------------------------------------------

I think the file is not packed with packing tools. Therefore, the md5 is the same with original file’s md5.

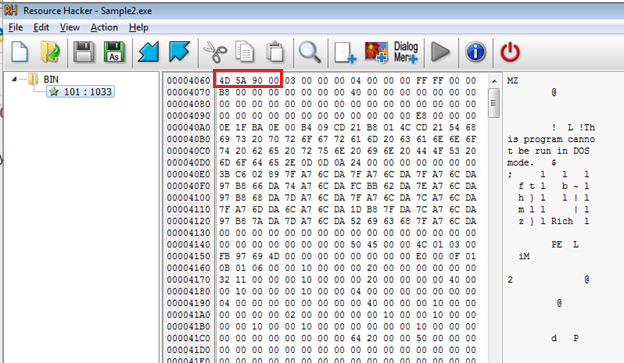
62 5a c0 5f d4 7a dc 3c 63 70 0c 3b 30 de 79 ab

--------------------------------------------------------------------------------------------

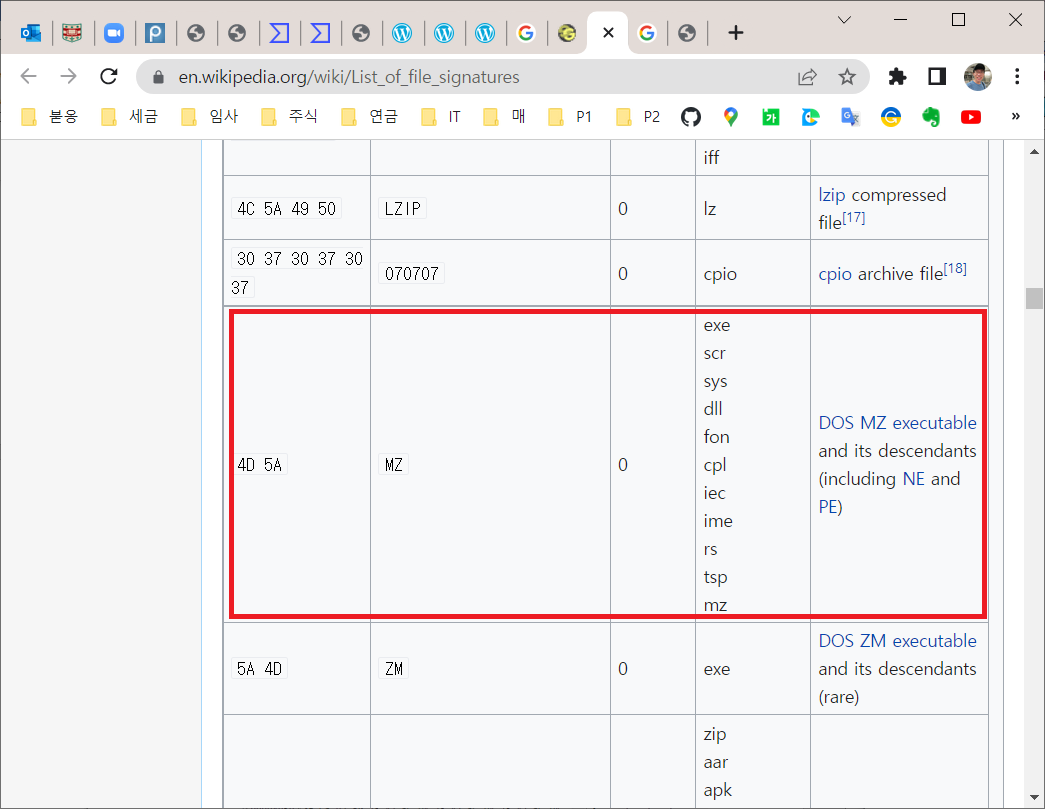
4) Use Resource Hacker to investigate the first resource in the unpacked file. What are the first 4 bytes of the resource (take a screenshot)? Based on these bytes, what is the type of this file? (10 pts)

--------------------------------------------------------------------------------------------

The first 4 bytes of the resource: 4D 5A 90 00



And I found out that it is a DOS executable file.



--------------------------------------------------------------------------------------------

5) Perform additional analysis on the resource file from the previous question. What do you think this file does and why is it malicious? (10 pts)

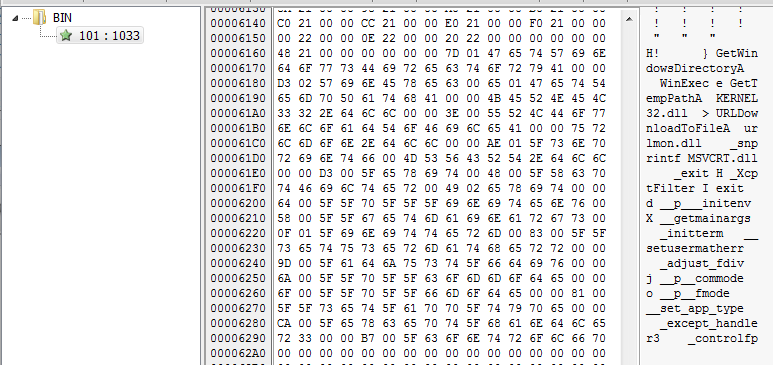
--------------------------------------------------------------------------------------------

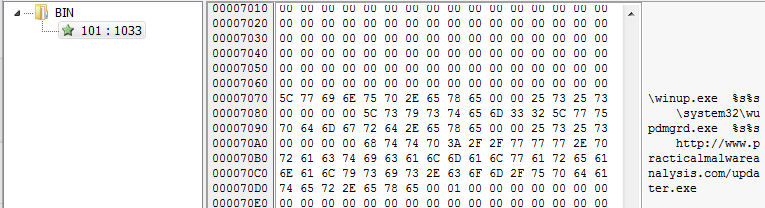
When I checked the strings in Sample2, I found several suspicious strings.

GetWindowsDirectoryA, WinExec, URLDownloadToFileA

http://www.practicalmalwareanalysis.com/updater.exe

From the strings I found, it will download some files from internet and try to execute it. This is typical behavior of a malicious program.





--------------------------------------------------------------------------------------------

6) Can you say what this malware does? You will be graded for providing an educated conclusion based on your findings, and not necessarily for the correctness of your conclusions. (2 points)

--------------------------------------------------------------------------------------------

The file starts with ‘4D 5A 90 00’ which is a sign of an executable file. And from the functions it uses, Sample2 will download a program from the URL ‘http://www.practicalmalwareanalysis.com/updater.exe’ and executes the file for malicious purpose.

--------------------------------------------------------------------------------------------