Lab # 1 - 4 — Assessment Worksheet

**Course Name and Number: IAM302**

**Student Name and Student Id:**

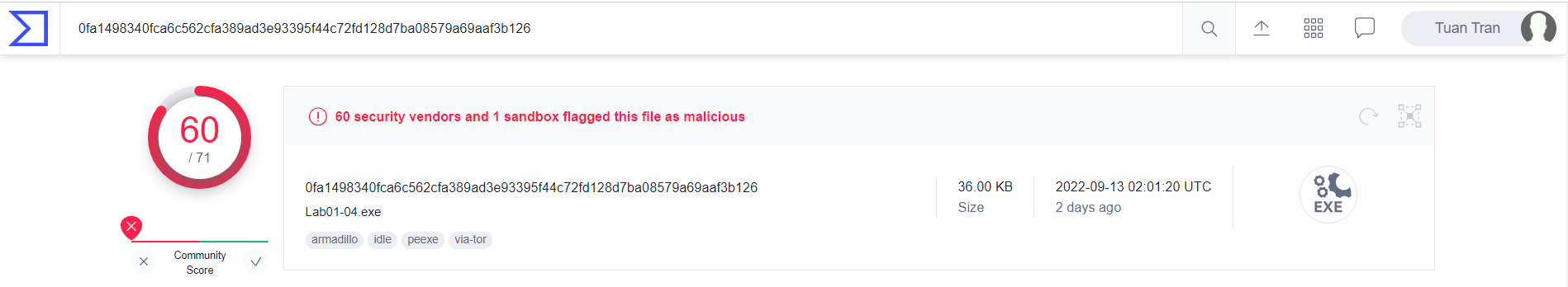
* **Tran Thanh Tuan – SE161095**
* **Diep Anh Vu – SE160365**
* **Shim Jun Woo – SE161064**

**Instructor Name: Vu Duc Ly**

## **Analyze the file Lab01-04.exe**

***Lab Assessment Questions & Answers***

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

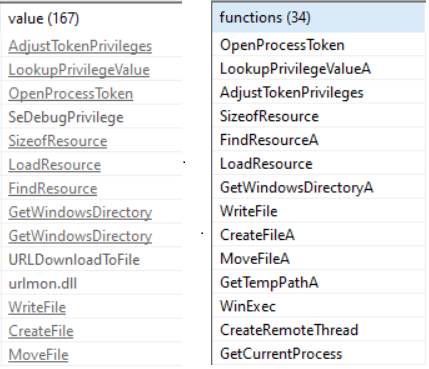
After scanning Lab01-04.exe using 71 different antivirus engines 60 of them indicated this file as malware. This is enough confirmation to treat this executable as malware.

1. **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.**

Three parts make the whole process of analysis if a file is obfuscated or packed. As usual, we focused on the number of imports and string within the file, the difference between its virtual-size and raw-size in the .text section, and the result of work done by petstudio tool.

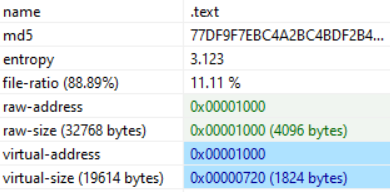
* Imports and strings

167 strings and 34 imports mean that the file isn't packed or obfuscated for now on.



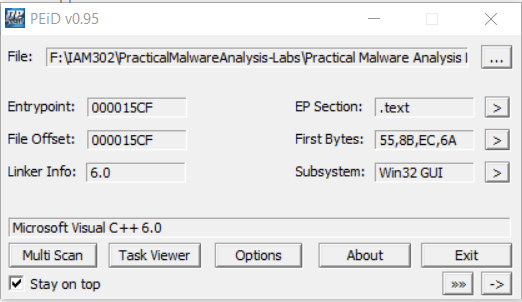
* The difference between virtual-size and raw-size in the .text section

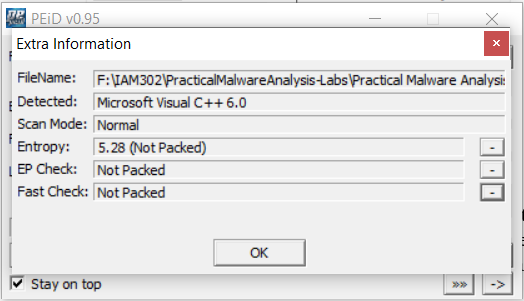
As you can see virtual-size is smaller than raw-size and the difference between them isn't large. It's obviously a normal situation so there is another indicator that this file isn't packed or obfuscated.



* Checking lab01-04.exe with PeiD

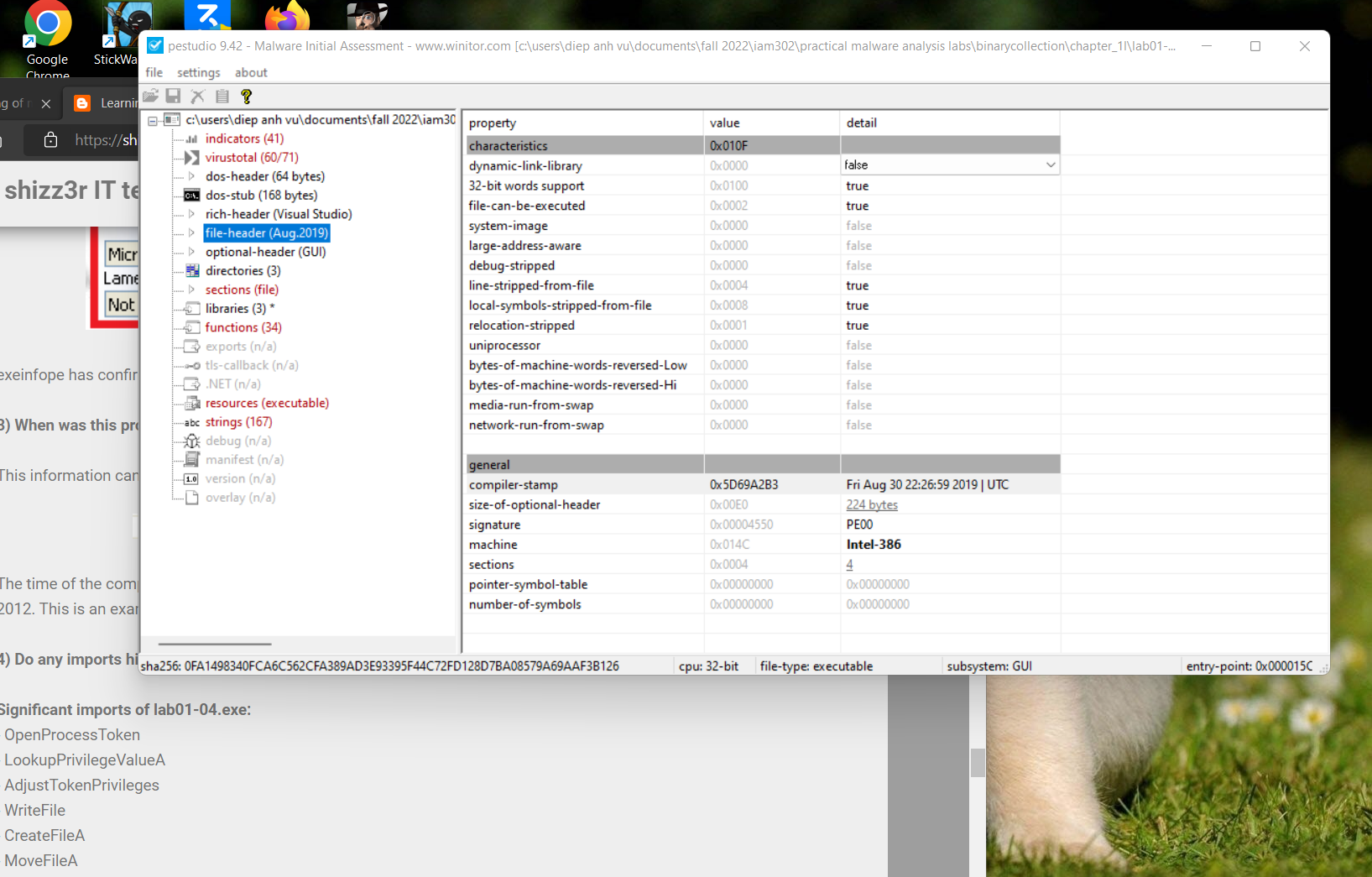
PeiD has confirmed that the analyzed file isn't packed or obfuscated. And also it has shown that Microsoft Visual C++ 6.0 is the compiler.





1. **When was this program compiled?**

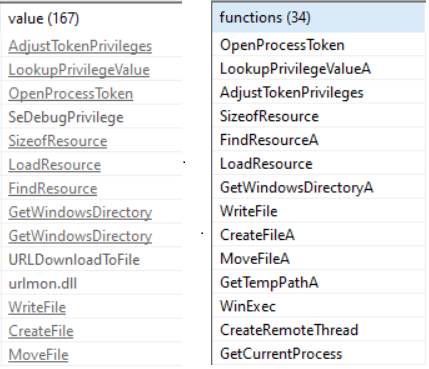
This information can be gathered from File header of lab01-04.exe file using pestudio tool and in the Details part from VirusTotal.





The time of the complication is Fri Aug 30 22:26:59 2019. The date is manipulated for sure since the book was released in 2012. This is an example that we shouldn’t believe compiler-stamps.

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



By using pestudio, we can define ignificant imports of lab01-04.exe

* OpenProcessToken
* LookupPrivilegeValueA
* AdjustTokenPrivileges
* WriteFile
* CreateFileA
* MoveFileA
* WinExec
* CreateRemoteThread

The list of significant imports is quite big compared to earlier examples. From looking at these functions we would tell that this executable probably creates another malicious file but this time within one of the directories with specific privileges. OpenProcessToken, LookupPrivilegeValueA, and AdjustTokenPrivileges might be used for this reason or to simply raise privileges of the created file, then use them to gain more benefits from victim's Windows. The last two functions are also very interesting. We think that they are used for execute some code inside the created process after setting correct privileges. lab01-04.exe can be used to prepare a new, dangerous program.

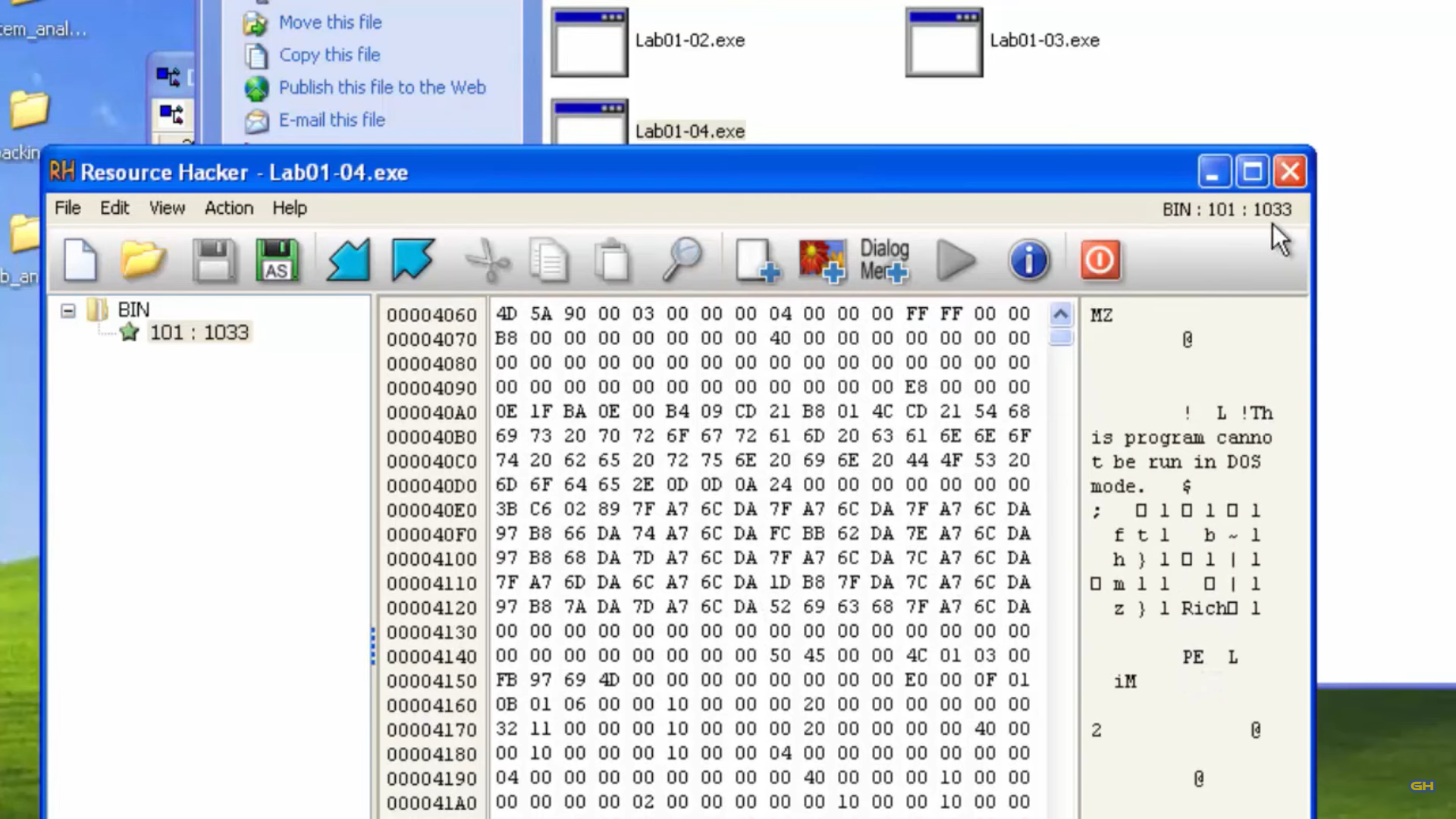
1. **What host- or network-based indicators could be used to identify this malware on infected machines?**

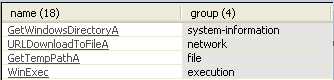
To determine host or network-based indicators that could be used to identify this malware on infected machines we have to look for suspicious strings. First of all, there is a name of the process that might be created by the malware - winup.exe. We would look at this file in further analysis for sure, but we can be wrong that this program is malicious. Next SeDebugPrivilege is one of the best hints at what this malware does. Without any reverse engineering, we are able to tell that the malicious program tries to get this privilege probably on the new file. When the exemplary process owns this privilege then it has access to all processes across the system and can do process replacement for example by placing the shellcode even in System files. Doing this an attacker has a straight path to add itself to administrators and then we are done. In the strings there is URLDownloadToFile so it might be a function that downloads updater.exe from the mentioned site and write the content into winup.exe to hide from the victim. Everything points to that lab01-04.exe is a downloader.

1. **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?**

IMAGE\_DOS\_HEADER is visible in the ascii representation of the sequence of bytes within the examined resource. Therefore we had decided to extract this resource to the binary file and then have tried to look deeper into it using pestudio. To extract a resource into a binary file -> Action -> Save resource to a BIN file.

Imports were very interesting to research within the saved binary file so here they are:





As you can see the executable hidden in the resource section imports the URLDownloadToFileA function and not lab01-04.exe itself. Now we know that networking action is being done by the executable code from resource section.