# CIT 285 - Lab #3: Malware

## Introduction

In this lab, you will analyze malware and legitimate file samples using both static and dynamic techniques. All sample files to be analyzed are found in the encrypted archive file lab03-files.zip to prevent anti-virus software from examining the file and deleting the samples.

1. Open the file folder in the left-hand menu, navigate to Downloads, right-click the encrypted lab03-files.zip folder and select ‘Extract Here’
2. In the new window make sure ‘Archive Manager’ is highlighted, then click the ‘Select’ button in the top right of the window. double-click the second folder.
3. Select the lab05-files folder in the new window, then hit the ‘Extract’ button in the top left corner.
4. A second window opens for saving the extracted lab-05 content to the Downloads folder. Click the ‘Extract’ button in the top right, enter the password **cit480**.
5. A new folder called ‘lab05-files’ is now in the Downloads location.
   * The sample filenames have a suffix of .txt in order to avoid accidental execution of malware by clicking on them or typing their names.

## 1: Static Analysis with VirusTotal

In a browser window, go to [www.virustotal.com](http://www.virustotal.com).

1.1: Upload **sample1.txt** to VirusTotal. Select the various tabs to answer the questions below.

* What is the detection ratio for sample #1?
* What are three of the names that sample #1 is identified as?

**The detection ration was 54/71. Three names identified in sample 1 are Panda, Rising, and Kingsoft.**

1.2: When a sample is uploaded to VirusTotal (www.virustotal.com), the web site reports a detection ratio.

* What does this detection ratio mean? Note: You may need to do a little outside research and searching through VirusTotal documentation.

“VirusTotal inspects items with over 70 antivirus scanners and URL/domain blacklisting services, in addition to a myriad of tools to extract signals from the studied content.” The detection ration means it scans over 70 antivirus scanners and the number that comes back is how many that is detected in this scan.

1.3: Upload sample #2 to VirusTotal in a new browser tab.

* What is the detection ratio for sample #2?
* What are three of the names that sample #2 is identified as?

**The detection ration was 0/69. There was zero detected viruses. Three names that are not detected are AVG, CMC, and DrWeb. Everything went undetected though.**

1.4: Compare the information under the tabs (e.g. ‘Behavior’ and ‘Details’) for both sample1.txt and sample2.txt.

* Discuss the differences for each sample and briefly explain what makes one file legitimate software and the other malware.

**In sample2 we can see under the behavior tab much more information on the source such as trusted sources, file version information, the history of the file specifically first and last submissions, and there are no details located in this file. This file is a legitimate software, and you can tell by the trusted sources and the file version being a Microsoft corporation. Compared to the sample1 which we can see a lack of information as a whole such as no history and no trusted sources. We can see the behavior tab in this sample, and it shows us the files open and registry keys. We can see that this is a malware file due to these characteristics.**

## 

## 2: Strings

The strings command prints strings of printable characters found in files. This command can be used to investigate the purpose of executables, since those strings often include paths to configuration and log files, error messages, configuration items, and even hardcoded passwords.

The strings command takes several important arguments including:

* **-a**, which causes it to list strings from all sections of the file
* **-#** option, which restricts output to strings of the specified # length or greater.

As the output of strings is long, we typically pipe its output to the less command for viewing. **Before issuing the commands below, be sure to move to the extracted directory containing the sample files or supply the full path to the file.**



**# strings -a -6 calc.exe | less**

2.1: How many strings of length 6 or greater are found in the file?

**# strings -a -6 calc.exe | wc -l**

**231 strings are found**

2.2: Using the command below, provide the dynamic library scripts used by calc.exe in the box below.

**# strings -a -6 calc.exe | grep ‘dll’**

A screenshot of a computer

Description automatically generated with medium confidence

2.3: Write one message that calc.exe may return when the program encounters an error or completes. Provide the message and the command used in below. Base your answer on the output of the strings command. *Note: create a command similar to the ones above or of your own design using, but it must include* strings. Hint: Regular expressions and grep may be useful.

**Strings -n 2 calc.exe**

**A screenshot of a computer

Description automatically generated**

2.4: What version of XML is used by calc.exe? Provide the strings command used to find this information and the returned output below.

**Strings -a calc.exe | grep ‘xml’**

**Version 1.0**

A screenshot of a computer

Description automatically generated

2.5: Use strings to search **sample #1** for any messages it may return and dynamic link libraries it uses. How do these results compare to those from **calc.exe**? Would you consider this suspicious? Refer back to the virus total results if needed.

**I would consider this suspicious. Running the commands above and testing it out there was no xml, and lack of information total. When using the grep ‘dll’ there was only the kernel32.dll which is a memory, input/output, and many more for windows OS. It is suspicious only seeing this one and the lack of information when searched.**

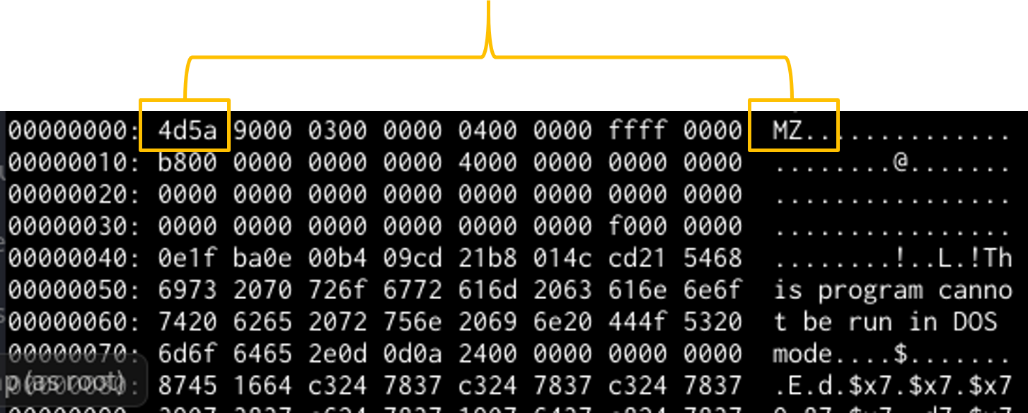
2.6: Write one error message and one dynamic library that is used by sample #2. Compare your results with those from calc.exe. Are these results suspicious?

These results are not suspicious at all. I would say these are normal outputs, when using the operations above. There was plenty of .dll’s and we could see similar data as when searched in the calc.exe

## 3: Executable Format Analysis

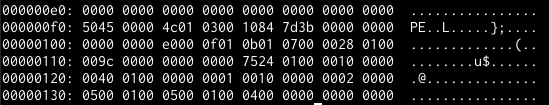
If you examine the beginning of calc.exe with a hex viewer, you will see the standard header of a MS Windows executable file.

**$ xxd calc.exe | head -20**

****

Note that the first two bytes, indicating the filetype, are MZ, which are the initials of one of the creators of MS-DOS. The file begins looking like a DOS MZ executable, but soon after the MZ string is a long message stating “This program cannot be run in DOS mode.” The part of the header from MZ up until the string PE is a small machine language program that will run under MS DOS and display that error message.

The **actual filetype** is seen with the string PE, indicating that this file is a Portable Executable (PE) format file.



PE files first appeared with Windows NT 3.1 and which is still the standard Windows executable format. Complete details of the PE header can be found in diagrams by Ero Carerra at

https://docs.google.com/file/d/0B3\_wGJkuWLytQmc2di0wajB1Xzg/edit?usp=drive\_web

3.1: Examine the headers of sample #1. Does it look like a PE file? If not, how does its header differ from calc.exe's header?

**This does not look like a PE file only because when looking at the results, yes it does have the MZ but it does not follow with the “This program cannot be run in DOS mode.” Before the PE. Also the PE is much longer and is very spaced out compared to the PE files of calc and sample2.**

3.2: Examine the headers of sample #2. Does it look like a PE file? If not, how does its header differ from calc.exe's header?

**This does look like a PE file.**

## Submitting

Upload a completed copy of the lab to Canvas by the due date.