Dynamic Heuristic Analysis Tool for Detection of Unknown Malware

CSC 471 - 02

Tyler Prehl

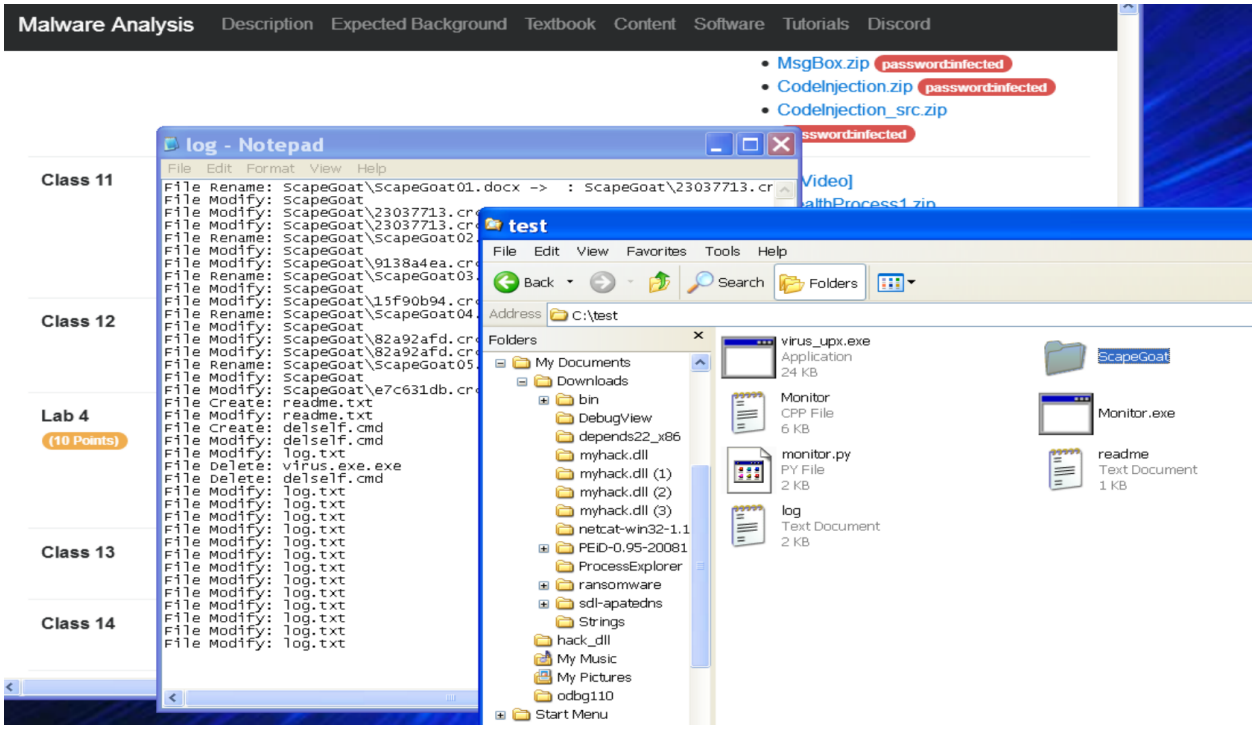
3/31/2022

Introduction

The purpose of this lab is to practice using dynamic heuristic analysis - analyzing behavior instead of static information - of log files to determine whether an executable file is malicious or not. This is a form of antivirus software that works to identify new forms of malware by using heuristic rules to determine what is and what is not malware. In this lab, I will create a Python script that will analyze a log file to determine whether the executable that produced the effects in the log file is malicious or not.

Analysis and Results

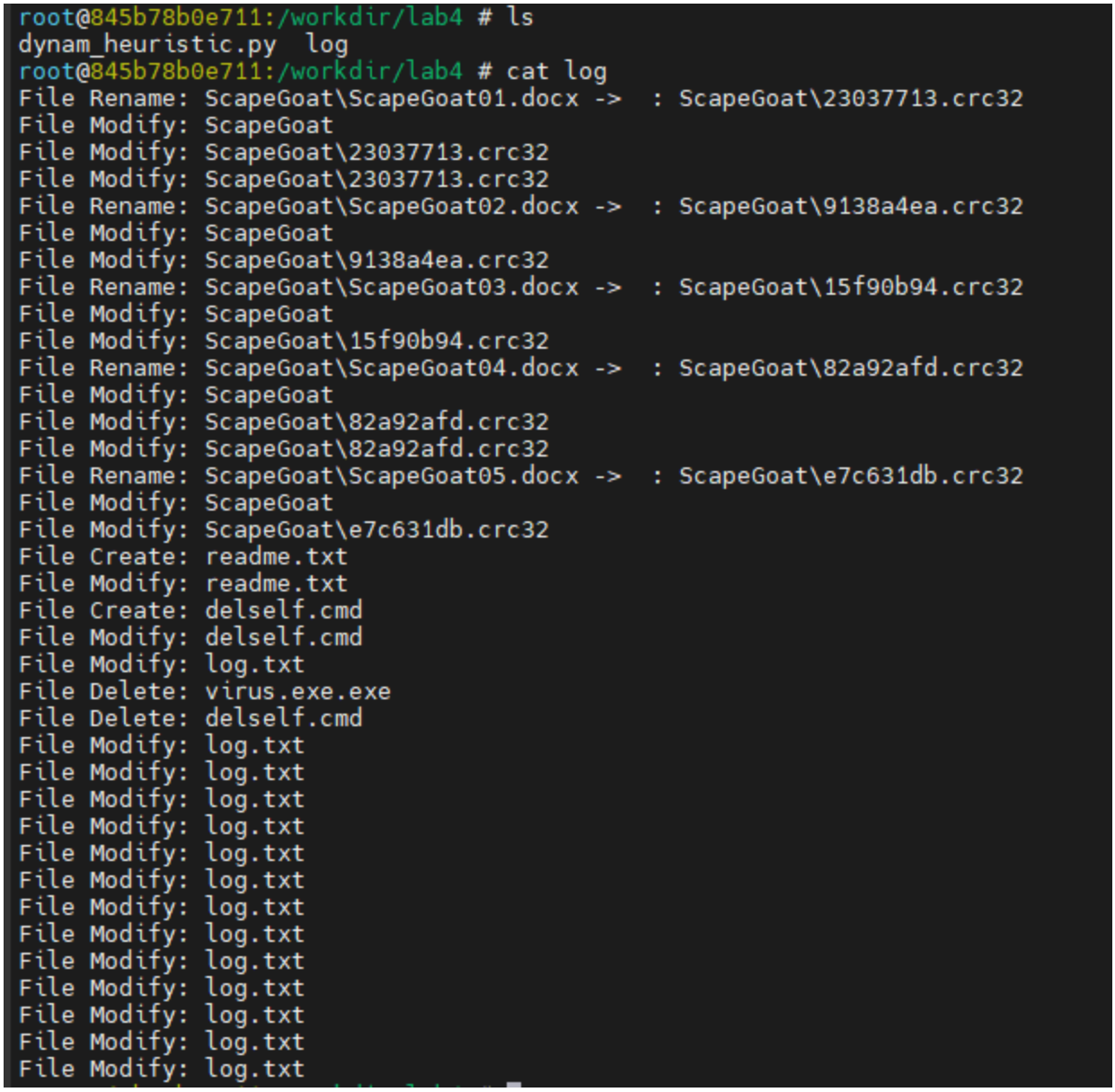
To set up the experiment, I downloaded a malicious executable file onto a Windows XP VM through VirtualBox to create a log file for analysis. In this image, you can see the log file created and the files downloaded to run and monitor the malicious executable file.



To properly identify whether the executable file is malware, we will create three heuristic rules:

1) If there are more than three Word documents (docx) in the ScapeGoat folder have been removed, and
2) If more than 3 files in the ScapeGoat have been modified, and
3) If the number of file self-delete (a file has been created and then immediately deleted) activities is greater than or equal to 1, then the executable file is malicious ransomware.

Because I created my dynamic heuristic analyzer Python file (dynam_heuristic.py) in BadgerCTF, I first moved the log file created in VirtualBox to BadgerCTF via a USB drive.



In dynam_heuristic.py, I decided the best method to analyze the log file would be by retrieving all the lines as strings and storing them in a list variable to be looped over to see which heuristic rules applied to each line, if any. I also initialized three variables (renamedDocx, modifiedFiles, and selfDeletes) to keep track of the number of the occurrences of heuristic rules being broken.

```
 GNU nano 5.4                                          dynam_heuristic.py
1 logFile = open('/workdir/lab4/log')
2 logLines = logFile.readlines()
3
4 renamedDocx = 0
5 modifiedFiles = 0
6 selfDeletes = 0
7 createdFiles = []
8
9 for line in logLines:
```

I then created the heuristic rules within the for loop to analyze each line for signs of malware.

```
for line in logLines:
    if line.find("Rename:")>-1 and  line.find(".docx ->")>-1 and line.find(".crc32")>-1:
        renamedDocx += 1

    if line.find("Modify")>-1:
        modifiedFiles += 1

    if line.find("Create")>-1:
        createdFiles.append(line[13:])

    if line.find("Delete")>-1 and (line[13:] in createdFiles):
        selfDeletes += 1
        createdFiles.remove(line[13:])
```

Lastly, I check to see if the totals of the heuristic rule tracking variables exceed the allowed amounts by the overall heuristic to determine whether I identified a malicious ransomware file through its behavior. Whether malicious ransomware was detected or not, a simple print statement lets the user know what occurred and the number of occurrences of broken rules. And of course, I close the opened log file.

```
if renamedDocx>3 and modifiedFiles>3 and selfDeletes>0:
    print("malware detected - HEUR:Trojan - Ransom.DocxEncrypt.Generic")
    print("renamedDocx: %d\nmodifiedFiles: %d\nselfDeletes: %d" % (renamedDocx, modifiedFiles, selfDeletes) )

else:
    print("no malware was found")
    print("renamedDocx: %d\nmodifiedFiles: %d\nselfDeletes: %d" % (renamedDocx, modifiedFiles, selfDeletes) )

logFile.close()
```

Running dynam_heuristic.py results in the following output, showing that the virus.exe file that's behavior was monitored and captured in the log file analyzed is indeed malicious ransomware.

```
root@845b78b0e711:/workdir/lab4 # python3 dynam_heuristic.py
malware detected - HEUR:Trojan - Ransom.DocxEncrypt.Generic
renamedDocx: 5
modifiedFiles: 28
selfDeletes: 1
root@845b78b0e711:/workdir/lab4 #
```

Discussion and Conclusion

This lab provided great insight into the basic concepts of dynamic analysis for antivirus software. Dynamic analyzing antivirus software is essential for identifying new forms of malware that static analyzing antivirus software cannot capture due to hacker methods such as packing. Going forward, having the knowledge of how dynamic analyzing antivirus software operates provides me with background knowledge and understanding of how to identify new or hidden forms of malware to disinfect infected machines.