

West Chester University

CSC 471

Dr. Si Chen

Spring 2024 Lab 5

Submitted by

Sean Berlin

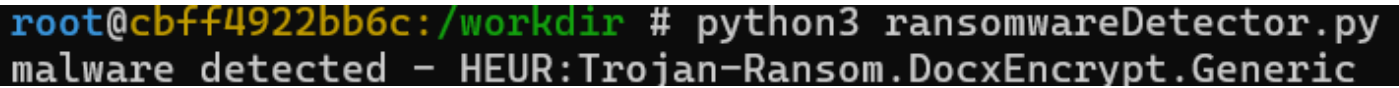
5/10/24

1. Introduction:

The purpose of this lab is to develop a heuristic dynamic analysis tool for detecting unknown ransomware. Traditional antivirus systems often struggle to detect new and obfuscated viruses due to sophisticated obfuscation techniques. Dynamic heuristic analysis offers a promising approach by examining program behavior in a controlled environment to determine malicious intent. By implementing heuristic rules based on log data, the developed program aims to enhance our ability to detect and mitigate emerging ransomware threats.

2. Analysis and Results:

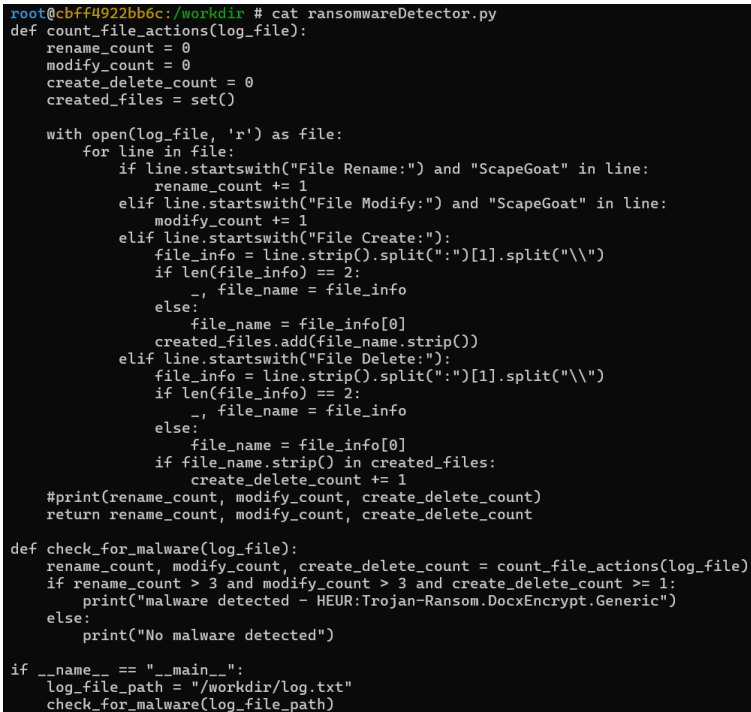
Screenshot of Successful Execution



```
root@cbff4922bb6c:/workdir # python3 ransomwareDetector.py
malware detected - HEUR:Trojan-Ransom.DocxEncrypt.Generic
```

Figure 1

Screenshot of ransomwareDetector.py



```
root@cbff4922bb6c:/workdir # cat ransomwareDetector.py
def count_file_actions(log_file):
    rename_count = 0
    modify_count = 0
    create_delete_count = 0
    created_files = set()

    with open(log_file, 'r') as file:
        for line in file:
            if line.startswith("File Rename:") and "ScapeGoat" in line:
                rename_count += 1
            elif line.startswith("File Modify:") and "ScapeGoat" in line:
                modify_count += 1
            elif line.startswith("File Create:"):
                file_info = line.strip().split(":")[1].split("\\")
                if len(file_info) == 2:
                    _, file_name = file_info
                else:
                    file_name = file_info[0]
                created_files.add(file_name.strip())
            elif line.startswith("File Delete:"):
                file_info = line.strip().split(":")[1].split("\\")
                if len(file_info) == 2:
                    _, file_name = file_info
                else:
                    file_name = file_info[0]
                if file_name.strip() in created_files:
                    create_delete_count += 1
    #print(rename_count, modify_count, create_delete_count)
    return rename_count, modify_count, create_delete_count

def check_for_malware(log_file):
    rename_count, modify_count, create_delete_count = count_file_actions(log_file)
    if rename_count > 3 and modify_count > 3 and create_delete_count >= 1:
        print("malware detected - HEUR:Trojan-Ransom.DocxEncrypt.Generic")
    else:
        print("No malware detected")

if __name__ == "__main__":
    log_file_path = "/workdir/log.txt"
    check_for_malware(log_file_path)
```

Figure 2

Screenshot of logs.txt

```
root@cbff4922bb6c:/workdir # cat log.txt
File Rename: ScapeGoat\ScapeGoat01.docx -> : ScapeGoat\23037713.crc32
File Modify: ScapeGoat
File Modify: ScapeGoat\23037713.crc32
File Rename: ScapeGoat\ScapeGoat02.docx -> : ScapeGoat\9138a4ea.crc32
File Modify: ScapeGoat
File Modify: ScapeGoat\9138a4ea.crc32
File Rename: ScapeGoat\ScapeGoat03.docx -> : ScapeGoat\15f90b94.crc32
File Modify: ScapeGoat
File Modify: ScapeGoat\15f90b94.crc32
File Rename: ScapeGoat\ScapeGoat04.docx -> : ScapeGoat\82a92afd.crc32
File Modify: ScapeGoat
File Modify: ScapeGoat\82a92afd.crc32
File Rename: ScapeGoat\ScapeGoat05.docx -> : ScapeGoat\e7c631db.crc32
File Modify: ScapeGoat
File Modify: ScapeGoat\e7c631db.crc32
File Create: readme.txt
File Modify: readme.txt
File Create: delself.cmd
File Modify: delself.cmd
File Modify: log.txt
File Delete: virus.exe
File Delete: delself.cmd
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
File Modify: log.txt
```

Figure 3

Analysis:

Generated by ChatGPT- To discuss the analysis and the results, the Python code snippet showcased in *Figure 2* elucidates the implementation of heuristic rules for analyzing the log.txt file generated during ransomware execution. In particular,

the `count_file_actions` function employs a meticulous approach to track occurrences of file renaming, modification, and creation-deletion events within the "ScapeGoat" folder. Of notable significance is the handling of the third heuristic rule: file self-deletion activity detection. This is achieved by maintaining a set of created file names and cross-referencing it when encountering file deletion events. If a deleted file was previously created within the same execution session, it signals a potential instance of file self-deletion, thus incrementing the `create_delete_count` variable. Subsequently, the `check_for_malware` function evaluates whether the observed activities, including potential file self-deletion instances, violate predefined heuristic rules, thereby signaling potential ransomware presence. In *Figure 3*, the content of the `log.txt` file is observed, demonstrating recorded activities essential for heuristic analysis. By integrating these components, the Python program enables effective detection and mitigation of ransomware threats through dynamic heuristic analysis, as depicted in *Figure 1*.

3. Discussion and Conclusion:

Generated by ChatGPT- In conclusion, this lab has effectively met its goals of enhancing understanding of antivirus systems and heuristic detection concepts, while also addressing the need for improved detection of new and obfuscated viruses. By developing a heuristic dynamic analysis tool for detecting unknown ransomware, we employed dynamic heuristic analysis techniques to examine program behavior in a controlled environment. Through meticulous parsing of the `log.txt` file generated during ransomware execution, our Python program successfully tracked file activities within the designated "ScapeGoat" folder. Notably, the implementation of heuristic rules enabled us to identify potential instances of ransomware behavior, including file renaming, modification, and self-deletion. This experience has provided valuable insights into the practical application of dynamic heuristic analysis, underscoring its significance in the ongoing battle against evolving cybersecurity threats.