Dhruv Chandel
2K18/IT/046
Kartik Bansal
2K18/IT/064

PROJECT REPORT

1. INTRODUCTION

- Executables are the programs packed into a executable package as one whole package for the purpose of Installation on the host device.
- Many a times these come from an untrusted source and it is unknown whether if they are genuine executables or are tampered with or are themselves some sort of malware.
- Static Malware Analysis is a technique where malware can be analysed without actually running it.
- Basic Static analysis involves techniques such as Analysing raw strings in the executables that might include some malicious IP addresses , pieces of malicious codes, definite patterns of code, DLL's imports, API calls etc.
- Some DLL imports are specific to some specific type of malware.
  eg : "winsock.DLL" is a DLL which provides functions which help a program(malware) to connect to any remote host via socket.
  Therefore, presence of these types of DLL's in strings can be quite handy in detecting or analysing malicious executables.
- Many-a-time executables are in packed format so very less strings are produced for these types of executables and number of DLL imports for such files are less which might indicate that their Packed nature.

2. PROBLEM DESCRIPTION

- When we analyse an executable and generate strings for it, the number of strings produced are very large.

https://github.com/dhruvchandel/static_malware_analysis

- Also large propotion of the string generated for an executable involves strings which represents memory addresses or other portions of code that are interpreted by the 'string module' as raw string which actually are not useful in any way for us to perform the static analysis of the executable.
- So we aim to design a approach to efficiently extract all the DLL imports and malicious IP addresses from the large number of strings so that we can easily analyses large text files for useful information.

3. PROPOSED APPROACH

We Aim to use '**REGEX module'** to solve this problem since REGEX modules helps in efficient searching of strings and patterns from any given text with ***Cross Platform GUI support.***

Also to assist client in analysing DLL's information extracted from executables we have performed Web scraping which provides a brief information regarding each DLL imported by the exeutable and also provide appropriate URL's for further investigation.

3.1 Assumptions

This Program might not work efficiently in case of compressed/packed executables and also it might not work adequately in case of executables that involve Runtime Linking *(because here DLL imports are runtime)* and can't be analysed statically.

3.2 Proposed Algorithm

Pseudo-code

1. Ask the client to upload the executableat GUI provided.
2. Extract all strings from executable and store them in temp.txt file.

3. Extract all DLL imports and IP addresses using REGEX module and temp.txt file is deleted from client' system.
4. Then using web-scrapping brief information about DLL imports and url links for further reading are provided on GUI.

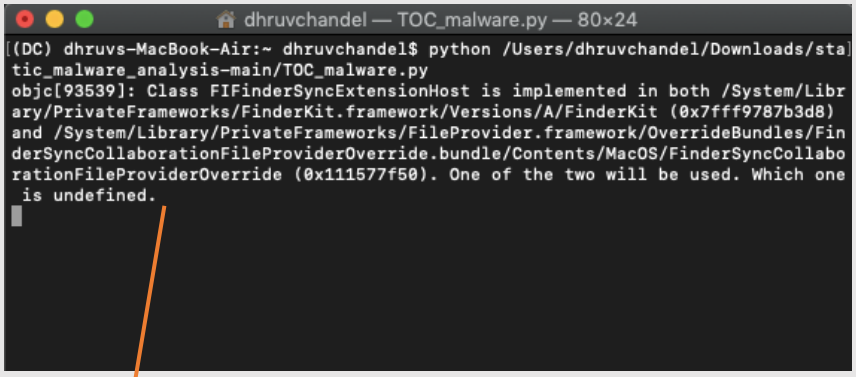
4. Implementation Details or Progress

.

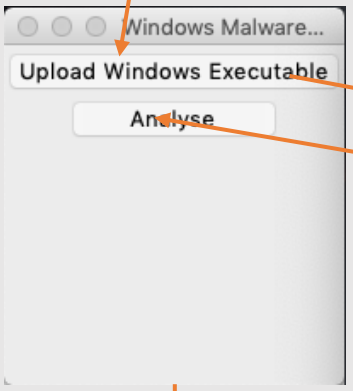
Table I: Implementation Environment, an example

Programming Language(s)	Python 3
Operating System	Cross-Platform (Windows/Mac/Linux)
Library Packages or APIs used (if any)	tkinter, BeautifulSoup, webbrowser, re, os, urllib, subprocess, strings module
Interface Design (GUI / Web / Other)	GUI using tkinter
Database Package (if any)	None
Servers Used (if any)	None

https://github.com/dhruvchandel/static_malware_analysis

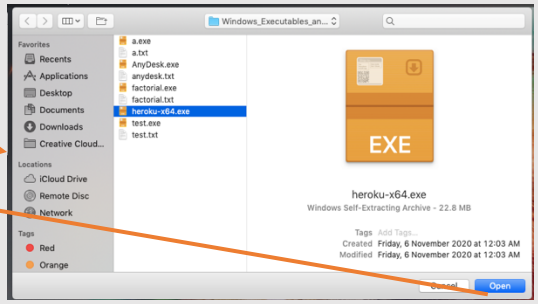

dhruvchandel — TOC_malware.py — 80×24
[(DC) dhruvs-MacBook-Air:~ dhruvchandel$ python /Users/dhruvchandel/Downloads/static_malware_analysis-main/TOC_malware.py
objc[93539]: Class FIFinderSyncExtensionHost is implemented in both /System/Library/PrivateFrameworks/FinderKit.framework/Versions/A/FinderKit (0x7fff9787b3d8) and /System/Library/PrivateFrameworks/FileProvider.framework/OverrideBundles/FinderSyncCollaborationFileProviderOverride.bundle/Contents/MacOS/FinderSyncCollaborationFileProviderOverride (0x111577f50). One of the two will be used. Which one is undefined.

1

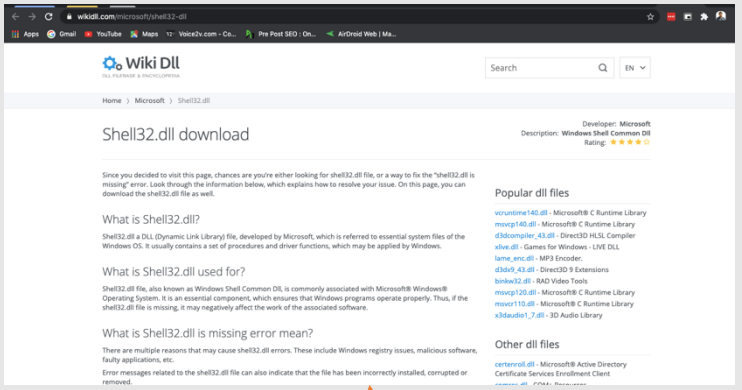

Windows Malware...
Upload Windows Executable
Analyse

2

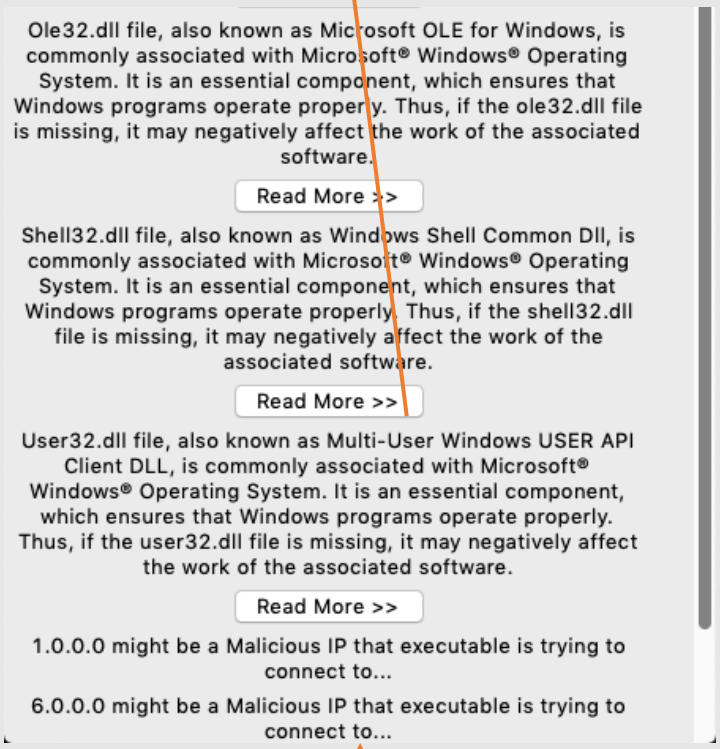
3


Windows_Executables_an...
Favorites
Recents
Applications
Desktop
Documents
Downloads
Creative Cloud...
Locations
iCloud Drive
Remote Disc
Network
Tags
Red
Orange
a.exe
a.txt
AnyDesk.exe
anydesk.txt
factorial.exe
factorial.txt
heroku-x64.exe
test.exe
test.txt
EXE
heroku-x64.exe
Windows Self-Extracting Archive - 22.8 MB
Tags Add Tags...
Created Friday, 6 November 2020 at 12:03 AM
Modified Friday, 6 November 2020 at 12:03 AM
Cancel
Open

4


Wiki Dll
Search
Shell32.dll download
Popular dll files
What is Shell32.dll?
What is Shell32.dll used for?
What is Shell32.dll is missing error mean?
Other dll files

5


Ole32.dll file, also known as Microsoft OLE for Windows, is commonly associated with Microsoft® Windows® Operating System. It is an essential component, which ensures that Windows programs operate properly. Thus, if the ole32.dll file is missing, it may negatively affect the work of the associated software.
Read More >>
Shell32.dll file, also known as Windows Shell Common Dll, is commonly associated with Microsoft® Windows® Operating System. It is an essential component, which ensures that Windows programs operate properly. Thus, if the shell32.dll file is missing, it may negatively affect the work of the associated software.
Read More >>
User32.dll file, also known as Multi-User Windows USER API Client DLL, is commonly associated with Microsoft® Windows® Operating System. It is an essential component, which ensures that Windows programs operate properly. Thus, if the user32.dll file is missing, it may negatively affect the work of the associated software.
Read More >>
1.0.0.0 might be a Malicious IP that executable is trying to connect to...
6.0.0.0 might be a Malicious IP that executable is trying to connect to...

5. CONCLUSION

Project was successfully implemented and problem stated was solved efficiently using Regex.However future scope of work in this project includes:-

- We could include IP tracker program to trace route for malicious Ips detected.
- We could create database of function calls imported.
- Certain executables that are packed can be detected by presence of certain API calls such as GetProcAddress() in Windows that can be used in packer detection programs.

https://github.com/dhruvchandel/static_malware_analysis