Identify File Dependencies using Dependency Walker

ILABS
CEH PRACTICAL

Any software program depends on the various inbuilt libraries of an OS that help in performing specified actions in a system. Programs need to work with internal system files to function correctly. Programs store their import and export functions in a kernel32.dll file. File dependencies contain information about the internal system files that the program needs to function properly; this includes the process of registration and location on the machine.

Find the libraries and file dependencies, as they contain information about the run-time requirements of an application. Then, check to find and analyze these files to provide information about the malware in the file. File dependencies include linked libraries, functions, and function calls. Check the dynamically linked list in the malware executable file. Finding out all library functions may allow guessing about what the malware program can do. You should know the various DLLs used to load and run a program.

DLLs

- Some of the standard DLLs are:

DLLs	Description of contents
Kernel32.dll	Core functionality such as access and manipulation of memory, files, and hardware
Advapi32.dll	Provides access to advanced core Windows components such as the Service Manager and Registry
User32.dll	User-interface components such as buttons, scrollbars, and components for controlling and responding to user actions
Gdi32.dll	Functions for displaying and manipulating graphics
Ntdll.dll	Interface to the Windows kernel
WSock32.dll and Ws2_32.dll	Networking DLLs that help to connect to a network or perform network-related tasks
Wininet.dll	Supports higher-level networking functions

Aim

The Dependency Walker tool lists all dependent modules of an executable file and builds hierarchical tree diagrams. It also records all functions that each module exports and calls. Further, it detects many common application problems such as missing and invalid modules, import and export mismatches, circular dependency errors, mismatched machine modules, and module initialization failures.

Here, we will use the Dependency Walker tool to identify the file dependencies of an executable file.

Other Dependency Checking tools

- ✓ Dependency-check (https://jeremylong.github.io)
- ✓ Snyk (https://snyk.io)
- ✓ RetireJS (https://retirejs.github.io)

DEMO

THANKS