CMSC 491/691
Malware Analysis

Basic Static Analysis

Topics

- Strings
- PE File Metadata
- Packing

Static Analysis

- Learning properties of a file without running it
- For now, just doing basic static analysis
  - Analyzing file properties / metadata
- Will do advanced static analysis later
  - Analyzing disassembled code

Strings

- Sequences of printable characters in a file
- Running strings on a file is usually first step of analysis
- Gives hints about functionality of program
- Example: `strings -n 8 [file path] | more`
  - Gets all strings of length >= 8 from a file and pipes output to more

FLOSS

- Like strings but more powerful

- Extracts:
  - ASCII strings
  - UTF-16 strings
  - Stack strings
  - Some encoded strings

- `floss -n 8 --no-decoded-strings [file path]`

Strings and FLOSS Demo (RJ)

Lab01-01.exe

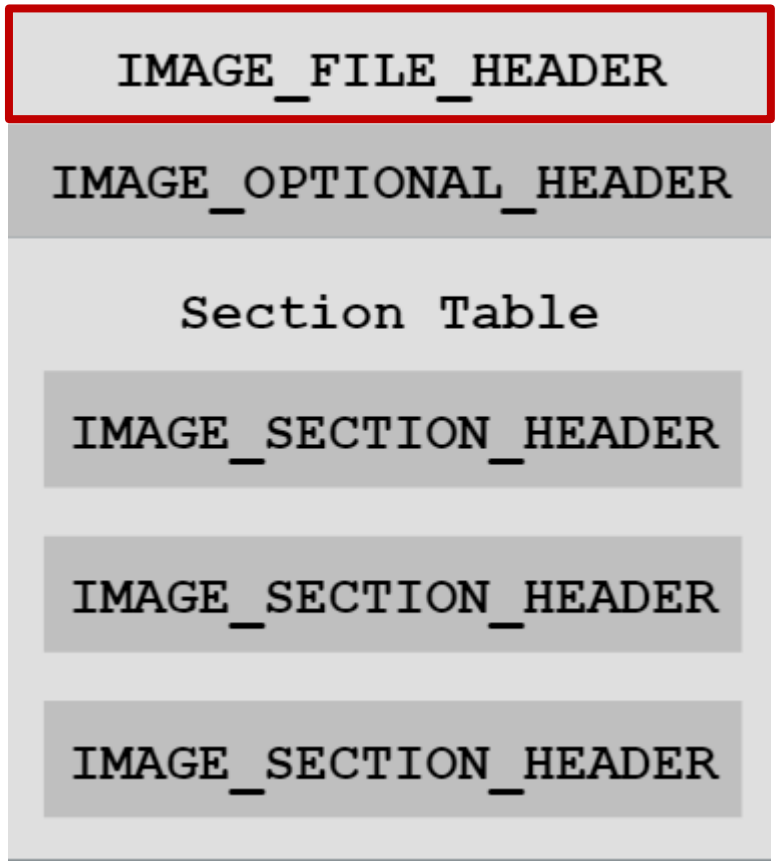
Lab09-02.exe

PE File Format

- File format for Windows executables
- Includes EXE, DLL, SYS, and other file types
- Describes how the executable file is loaded into memory
- Contains lots of metadata that is useful to malware analysts!

The IMAGE_FILE_HEADER

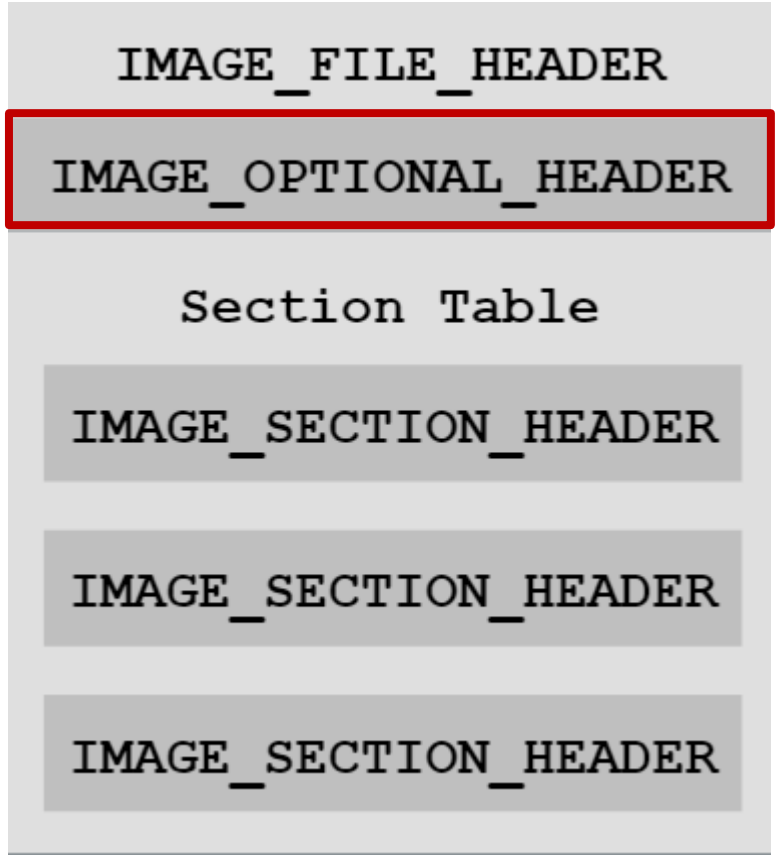
- Contains basic file information
  - NumberOfSections
  - TimeDateStamp
  - Characteristics



The IMAGE_OPTIONAL_HEADER

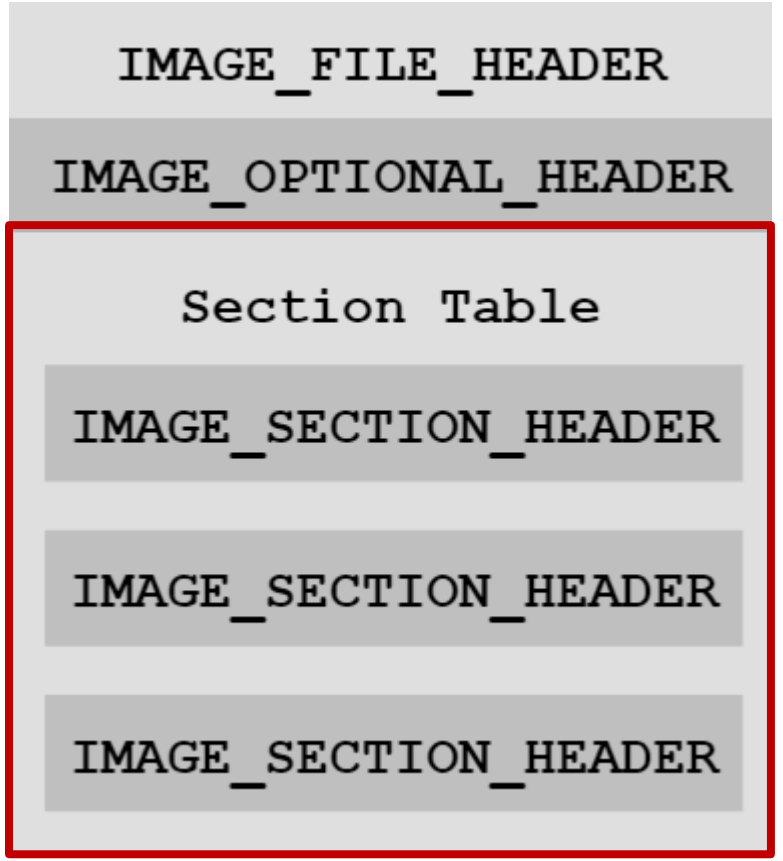
- Not actually optional

- Contains lots of important metadata:
  - AddressOfEntryPoint
  - Sizes of various parts of the file that get loaded into memory
  - Minimum versions of operating system, linker, image, subsystem



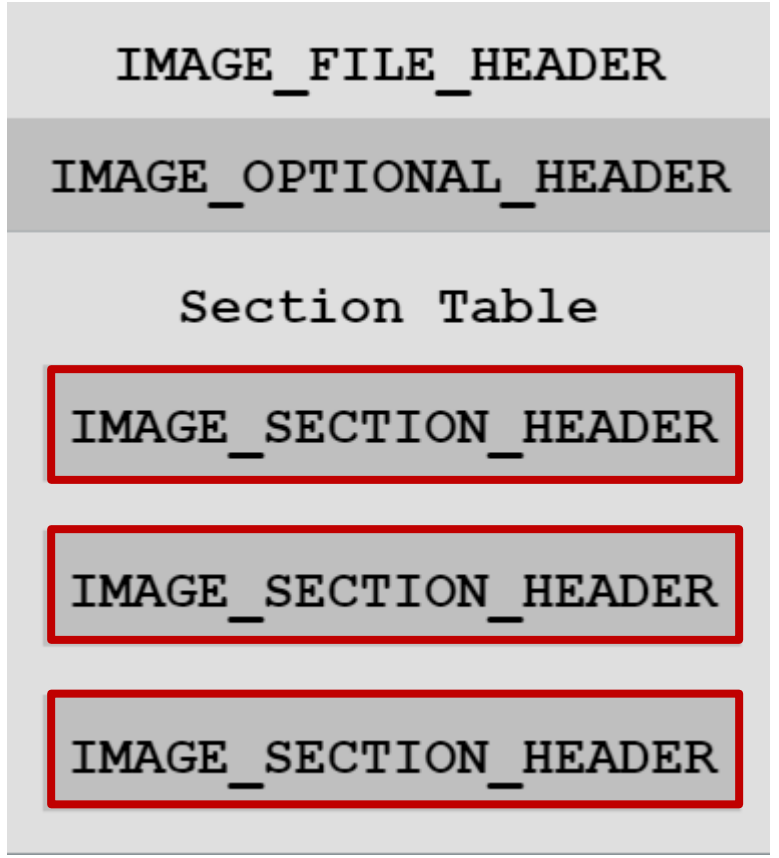
The Section Table

- Each section corresponds to a continuous area of memory in a process
- Section table contains an array of IMAGE_SECTION_HEADERs



IMAGE_SECTION_HEADERs

- Each contains that section's:
  - Name
  - VirtualAddress
  - VirtualSize
  - SizeOfRawData
  - Characteristics



Common PE Sections

Section name	Contents
.text	Executable code
.data	Initialized data
.idata	Import Address Table
.rsrc	Resource Directory Table
.rdata	Read-only initialized data

- Many other common section names
- Unusual section names are a malicious indicator

VirusTotal Demo
PE File Format Demo

Lab -03-04.exe

Imports

- Import Address Table lists which functions a file imports from the Windows API
  - Windows API functions defined in DLL files
- Imports give info about what actions a file can perform
- Commonly second step in basic static analysis, after strings

Resources

- Additional data/file contained within a PE file
- In legitimate files, often icons, application manifest, etc
- Malware often hides things in resources!

Resources and Imports Demo

VT and Imports demo as time permits

ResourceHacker and Lab03-03.exe

Packers

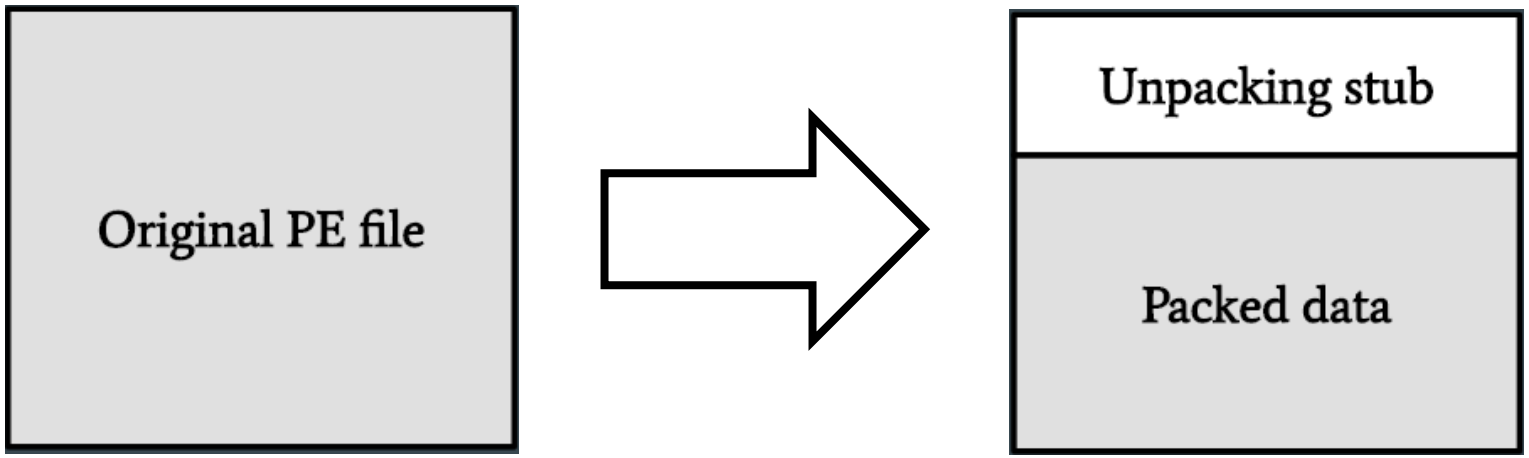
- Malware authors want to make it difficult for you to perform static analysis on their malware

- Use packers to hide:
  - Executable code
  - Strings
  - Imports

How Packers Work

- Compress original program and add an unpacker stub

- When the packed executable is run, the stub unpacks the compressed program into memory and runs it



Indicators that a File is Packed

- File / Section entropy > 7
- Few readable strings
- Unusual section names
- Imports resolved using runtime linking
- Sections with unusual raw / virtual sizes

- PEiD, DIE, VirusTotal are decent at detecting packers
  - Notice lots of some false positives for some packers though

Entropy

- A byte has 2^8 possible values, so a truly random sequence of bytes has an entropy of 8
- Executable code usually has an entropy around 4-6
- Obfuscated / encrypted data usually has an entropy over 7, often near 8

Runtime Linking

- Malware authors don't want you to be able to easily analyze a program's imports

- Can hide a file's imports until it is run by using runtime linking
  - Resolves imports as the file runs
  - Can import functions that are not listed in the Import Address Table (IAT)

How Runtime Linking Works

- LoadLibrary – Gets a handle (expand on this concept, not just a pointer) to any DLL file on a system
- GetProcAddress – Gets address of any function in a DLL
- Together, allows a program to import a function from any DLL

Packing Indicators Demo

Lab01-02.exe

Lab01-03.exe