Malware Analysis

SIEM XPERT
Get trained today !

Welcome!

Who is this course for?

- Anyone willing and dedicated to learning malware analysis.
- You do not need to have prior knowledge, if you do, it will be helpful.

What this course will require?

- Dedication.

What we will be covering in this course.

1. Introduction to malware analysis.
2. Setting up our environment.
3. Static analysis.
4. Introduction to Assembly language.
5. Code Analysis.
6. Dynamic Analysis.
7. Behavioural Analysis.

SIEM XPERT

Get trained today !

Let’s get started.

What is Malware?

- Malware is an executable or a binary that is malicious in nature.
- Malware is used by attackers to perform a variety of malicious actions like:
  - Spying on the target through:
    - RAT's
    - Keyloggers
  - Data Exfiltration
  - Data encryption and destruction
    - Ransomware

Types Of Malware

Malware refers to any binary or executable that is malicious, however Malware is sorted in to further denominations based on its functionality. Here are the various types of Malware:

- Trojans - Type of malware that disguises itself as a legitimate program for social engineering purposes. It can destroy and exfiltrate data and can also be used for spying.
- RAT's - Type of malware that allows the attacker to remotely access and execute commands on the system. It's functionality can be extended with modules like keyloggers.
- Ransomware - Type of malware that encrypts all files on the system and holds the system and its data for ransom.
- Dropper - Type of malware whose purpose is to download/drop additional malware.

What Is Malware Analysis?

Malware analysis is the process of analyzing a malware sample/binary and extracting as much information as possible from it. The information we extract helps us understand the scope of the functionality of the Malware, how the system was infected with the malware and how to defend against similar attacks in the future.

Objectives of malware analysis:

- To understand the type of malware and the entire scope of what it can do (functionality). Is it a Keylogger, RAT or Ransomware.
- How the system was infected with the malware. Is it a targeted attack or a phishing attack?
- How it communicates with the attacker.
- To exfiltrate useful indicators like registry entries/keys and filenames for the purpose generating signatures that can be used to detect future detection.

Types Of Malware Analysis

- Static analysis - Is the process of analyzing malware without executing or running it. The objective is to extract as much metadata from the malware as possible. Example; strings, PE headers.
- Dynamic analysis - Is the process of executing malware and analyzing it's functionality and behaviour. The objective is to understand exactly how and what the malware does during the execution. This is done in a debugger.
- Code Analysis - Is the process of analyzing/reverse engineering assembly code. This can be both statically and dynamically done (Static and dynamic code analysis)
- Behavioural analysis - Is the process of analyzing and monitoring the malware after execution. It involves monitoring the processes, registry entries and network monitoring to determine the workings of the malware.

SIEM XPERT

Get trained today !

Next up: Setting up our environment

Setting up our environment

Tools we will be using:

- Hypervisor - VirtualBox or VMware
- Windows 7 VM 32/64bit - 64 bit preferable.
- FLARE VM - Windows malware analysis distribution
  - Comes prepackaged with all the tools we need for malware analysis.

Note: Ensure you disable Windows Update and Windows Defender on your analysis VM.

Security guidelines

- Keep your Hypervisor updated.
- When executing malware ensure your network configuration is set to host-only.
- Do not plug any USB devices in to the VM.
- Make sure you download compressed and password protected samples to avoid accidental execution.
- Take snapshots!
- Do not store any valuable data on your analysis VM.
- Disable shared folders, before execution or analysis.

Static Analysis

- Static analysis is the process of analyzing malware/binary without executing it.
- The objective is to extract useful information from the malware, this will help us get an idea of the type of malware and what the malware can do. This information is useful for future analysis as it will allows us to efficiently analyze the sample going forward.

Static analysis flow - How we will approach a sample

- Identifying the file type - Target OS, architecture and format (dll, exe)
- Identifying the malware - Generating a hash of the malware, this will give the malware a unique identifier. Using the hash to see if anyone else has analyzed the malware.
- Strings - Strings give us an idea/glimpse of what the malware can do.
- Packing & Obfuscation - Obfuscation & packing are techniques used to prevent detection. Unpacking or deobfuscating can reveal additional information.
- PE headers - The PE header reveals a lot of information on the malware functionality.

Identifying the file type

- Identifying the file type is extremely important as it helps us identify the target OS and the corresponding architecture.
- An example of a Windows executable file is the PE (Portable Executable).
- A PE could be in the form of; .exe, .dll etc.
- To accurately identify a file type we need to analyze the file signature. This is to avoid false positives caused by the use of double extensions.
- The file signature exists on the file header.
- The file signature for PE files are represented by hexadecimal values of 4D 5A or MZ in the first 2 bytes (0-1).
- PE programs also have the notice “This program cannot be run in DOS mode”
- The PE header begins at hex 50 45.

Note: Attackers may use archiving/packing to evade signature based identification. We will cover this in the packing section.

Tools we will be using

- HxD - Hex Editor
- Exeinfo PE - Retrieves the windows PE header information. It also detects if the executable has been packed and detects the paker version and how to unpack it.
- Pestudio
- CFF explorer

Malware hashing

- Malware hashing is the process of generating cryptographic hashes for the file content of the target malware. We are hashing the malware file.
- The hashing algorithms used in malware identification are:
  - MD5
  - SHA-1
  - SHA-256
- The hashing process gives us a unique digest known as a fingerprint.
- This means we can create unique fingerprints for malware samples.

Why should you hash?

- For accurate identification of malware samples, rather than using file names for malware. Hashes are unique.
- Hashes are used to identify malware on malware analysis sites. (Virus Total).
- Hashes can be used to search for any previous detections or for checking online if the sample has been analyzed by other researchers.

Analyzing strings

SIEMXPERT
Get trained today !

- Strings Analysis - This is the process of extracting readable characters and words from the malware.
- Strings can give us valuable information about the malware functionality.
- Malware will usually contain useful strings and other random strings, also known as garbage strings.
- Strings are in ASCII and Unicode format. (We need to specify the type of strings we want to extract during analysis, as some tools only extract ASCII.
- The types of strings we are looking for are:
  - File names
  - URL's (Domains the malware connects to)
  - IP Addresses
  - Registry Keys
- Attackers may also include fake strings to disrupt our analysis.

Note: Strings give us a glimpse of what the malware can do.

Tools we will be using

- Strings command line utility.
- Shell extensions
- Pestudio
- peid

Packers

- A packer is a tool that is used to compress the content of the malware.
- Attackers will use packers to obfuscate the content of the malware, this makes it difficult to analyze strings.
- Packers compress an executable and when executed the packed executable will be decompressed. This allows us to analyze the original unpacked executable.

Tools we will be using

- UPX
- EXEinfo PE

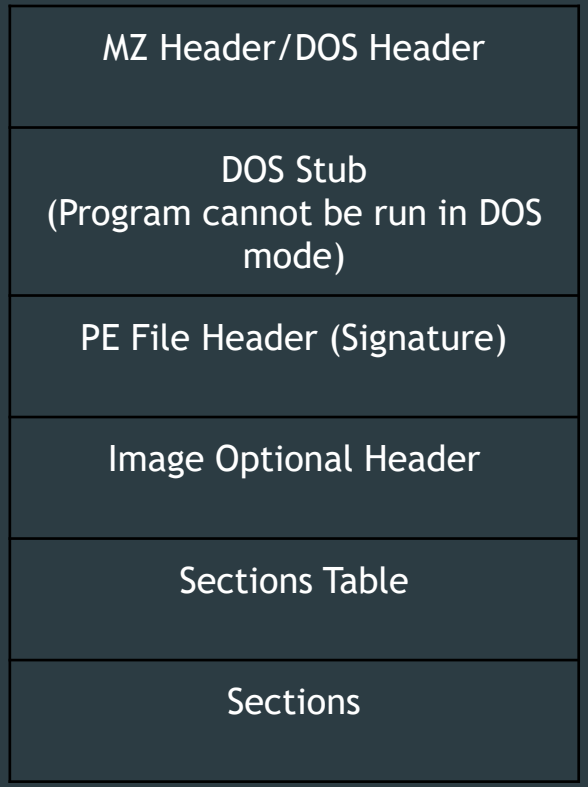
PE Header

- The PE header contains the information the OS requires to run the executable.
- This information is very useful, as it can give us more information about the functionality of the malware and how the malware interacts with the OS.

Why is the PE header important?

1. It contains all of the important and necessary information required by the OS to execute the executable.
2. It contains information that specifies where the executable needs to be loaded in to memory.
3. It contains the libraries that the executable requires to be loaded (dll).
4. It contains information that specifies where the execution begins.

PE Header Structure



Defines the file as an executable binary

Prints a message when run in DOS (Exists for compatibility)

Defines the executable as a PE

Stores important information about the executable: Like the subsystem and the entry point

Instructions on how to load the executable into memory

Executable sections of code and data used by the executable

Sections

PE Sections

Section Name	Function
.code / .text	Executable code
.data	Stores Data (R/W)
.rdata	Stores Data (Read Only)
.idata	Stores The Import Table
.edata	Stores Export Data
.rsrc	Stores Resources (Strings, icons)

Examining The Resources Section (.rsrc)

The resources section contains all the necessary files and information that are used/required by the executable. For example: icons, dialogs

Why is it important?

- Attackers can utilize the resources section to store more malicious files and data like payloads, droppers, configuration info etc.
- The resource section is also useful as it may contain information about the origin of the malware.

Tools we will be using

- Pestudio
- Resource Hacker