Learning Malware Analysis

Explore the concepts, tools, and techniques to analyze and investigate Windows malware

Packt

www.packt.com

By Monnappa K A

Sophos Ex. 1011
Sophos Ltd. and Sophos Inc. v. Webroot Inc.
PGR2023-00031
Page 1 of 9

Learning Malware Analysis

Explore the concepts, tools, and techniques to analyze and investigate Windows malware

Monnappa K A

BIRMINGHAM - MUMBAI

Learning Malware Analysis

Copyright © 2018 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Commissioning Editor: Gebin George
Acquisition Editor: Shrilekha Inani
Content Development Editor: Sharon Raj
Technical Editor: Prashant Chaudhari
Copy Editor: Safis Editing
Project Coordinator: Virginia Dias
Proofreader: Safis Editing
Indexer: Aishwarya Gangawane
Graphics: Tom Scaria
Production Coordinator: Nilesh Mohite

First published: June 2018

Production reference: 2200718

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-78839-250-1

www.packtpub.com

1
Introduction to Malware Analysis

The number of cyber attacks is undoubtedly on the rise, targeting government, military, public and private sectors. These cyber attacks focus on targeting individuals or organizations with an effort to extract valuable information. Sometimes, these cyber attacks are allegedly linked to cybercrime or state-sponsored groups, but may also be carried out by individual groups to achieve their goals. Most of these cyber attacks use malicious software (also called malware) to infect their targets. Knowledge, skills, and tools required to analyze malicious software are essential to detect, investigate and defend against such attacks.

In this chapter, you will learn the following topics:

- What malware means and its role in the cyber-attacks
- Malware analysis and its significance in digital forensics
- Different types of malware analysis
- Setting up the lab environment
- Various sources to obtain malware samples

1. What Is Malware?

Malware is a code that performs malicious actions; it can take the form of an executable, script, code, or any other software. Attackers use malware to steal sensitive information, spy on the infected system, or take control of the system. It typically gets into your system without your consent and can be delivered via various communication channels such as email, web, or USB drives.

The following are some of the malicious actions performed by malware:

- Disrupting computer operations
- Stealing sensitive information, including personal, business, and financial data
- Unauthorized access to the victim's system
- Spying on the victims
- Sending spam emails
- Engaging in distributed-denial-of-service attacks (DDOS)
- Locking up the files on the computer and holding them for ransom

Malware is a broad term that refers to different types of malicious programs such as trojans, viruses, worms, and rootkits. While performing malware analysis, you will often come across various types of malicious programs; some of these malicious programs are categorized based on their functionality and attack vectors as mentioned here:

- **Virus or Worm**: Malware that is capable of copying itself and spreading to other computers. A virus needs user intervention, whereas a worm can spread without user intervention.
- **Trojan**: Malware that disguises itself as a regular program to trick users to install it on their systems. Once installed, it can perform malicious actions such as stealing sensitive data, uploading files to the attacker's server, or monitoring webcams.
- **Backdoor / Remote Access Trojan (RAT)**: This is a type of *Trojan* that enables the attacker to gain access to and execute commands on the compromised system.
- **Adware**: Malware that presents unwanted advertisements (ads) to the user. They usually get delivered via free downloads and can forcibly install software on your system.
- **Botnet**: This is a group of computers infected with the same malware (called *bots*), waiting to receive instructions from the command-and-control server controlled by the attacker. The attacker can then issue a command to these bots, which can perform malicious activities such as DDOS attacks or sending spam emails.
- **Information stealer**: Malware designed to steal sensitive data such as banking credentials or typed keystrokes from the infected system. Some examples of these malicious programs include key loggers, spyware, sniffers, and form grabbers.
- **Ransomware**: Malware that holds the system for ransom by locking users out of their computer or by encrypting their files.
- **Rootkit**: Malware that provides the attacker with privileged access to the infected system and conceals its presence or the presence of other software.

Sophos Ex. 1011
Page 5 of 9

- **Downloader or dropper**: Malware designed to download or install additional malware components.

A handy resource for understanding malware terminologies and definitions is available at `https://blog.malwarebytes.com/glossary/`.

Classifying malware based on their functionalities may not always be possible because a single malware can contain multiple functionalities, which may fall into a variety of categories mentioned previously. For example, malware can include a worm component that scans the network looking for vulnerable systems and can drop another malware component such as a *backdoor* or a *ransomware* upon successful exploitation.

Malware classification can also be undertaken based on the attacker's motive. For example, if the malware is used to steal personal, business, or proprietary information for profit, then the malware can be classified as c*rimeware* or c*ommodity malware*. If the malware is used to target a particular organization or industry to steal information/gather intelligence for espionage, then it can be classified as t*argeted* or e*spionage malware*.

2. What Is Malware Analysis?

Malware analysis is the study of malware's behavior. The objective of malware analysis is to understand the working of malware and how to detect and eliminate it. It involves analyzing the suspect binary in a safe environment to identify its characteristics and functionalities so that better defenses can be built to protect an organization's network.

3. Why Malware Analysis?

The primary motive behind performing malware analysis is to extract information from the malware sample, which can help in responding to a malware incident. The goal of malware analysis is to determine the capability of malware, detect it, and contain it. It also helps in determining identifiable patterns that can be used to cure and prevent future infections. The following are some of the reasons why you will perform malware analysis:

- To determine the nature and purpose of the malware. For example, it can help you determine whether malware is an information stealer, HTTP bot, spam bot, rootkit, keylogger, or RAT, and so on.
- To gain an understanding of how the system was compromised and its impact.

Sophos Ex. 1011
Page 6 of 9

- To identify the network indicators associated with the malware, which can then be used to detect similar infections using network monitoring. For example, during your analysis, if you determine that a malware contacts a particular *domain/IP address*, then you can use this domain/IP address to create a signature and monitor the network traffic to identify all the hosts contacting that domain/IP address.
- To extract host-based indicators such as filenames, and registry keys, which, in turn, can be used to determine similar infection using host-based monitoring. For instance, if you learn that a malware creates a registry key, you can use this registry key as an indicator to create a signature, or scan your network to identify the hosts that have the same registry key.
- To determine the attacker's intention and motive. For instance, during your analysis, if you find that the malware is stealing banking credentials, then you can deduce that the motive of the attacker is monetary gain.

Threat intelligence teams very often use the indicators determined from a malware analysis to classify the attack and attribute them to known threats. Malware analysis can help you get information about who could be behind the attack (competitor, state-sponsored attack group, and so on).

4. Types Of Malware Analysis

To understand the working and the characteristics of malware and to assess its impact on the system, you will often use different analysis techniques. The following is the classification of these analysis techniques:

- **Static analysis**: This is the process of analyzing a binary without executing it. It is easiest to perform and allows you to extract the metadata associated with the suspect binary. Static analysis might not reveal all the required information, but it can sometimes provide interesting information that helps in determining where to focus your subsequent analysis efforts. `Chapter 2`, *Static Analysis*, covers the tools and techniques to extract useful information from the malware binary using static analysis.

Sophos Ex. 1011
Page 7 of 9

- **Dynamic analysis (Behavioral Analysis)**: This is the process of executing the suspect binary in an isolated environment and monitoring its behavior. This analysis technique is easy to perform and gives valuable insights into the activity of the binary during its execution. This analysis technique is useful but does not reveal all the functionalities of the hostile program. `Chapter 3`, *Dynamic Analysis*, covers the tools and techniques to determine the behavior of the malware using dynamic analysis.
- **Code analysis**: It is an advanced technique that focuses on analyzing the code to understand the inner workings of the binary. This technique reveals information that is not possible to determine just from static and dynamic analysis. Code analysis is further divided into *Static code analysis* and *Dynamic code analysis*. *Static code analysis* involves disassembling the suspect binary and looking at the code to understand the program's behavior, whereas *Dynamic code analysis* involves debugging the suspect binary in a controlled manner to understand its functionality. Code analysis requires an understanding of the programming language and operating system concepts. The upcoming chapters (*Chapters 4 to 9*) will cover the knowledge, tools, and techniques required to perform code analysis.
- **Memory analysis (Memory forensics)**: This is the technique of analyzing the computer's RAM for forensic artifacts. It is typically a forensic technique, but integrating it into your malware analysis will assist in gaining an understanding of the malware's behavior after infection. Memory analysis is especially useful to determine the stealth and evasive capabilities of the malware. You will learn how to perform memory analysis in subsequent chapters (*Chapters 10 and 11*).

Integrating different analysis techniques while performing malware analysis can reveal a wealth of contextual information, which will prove to be valuable in your malware investigation.

5. Setting Up The Lab Environment

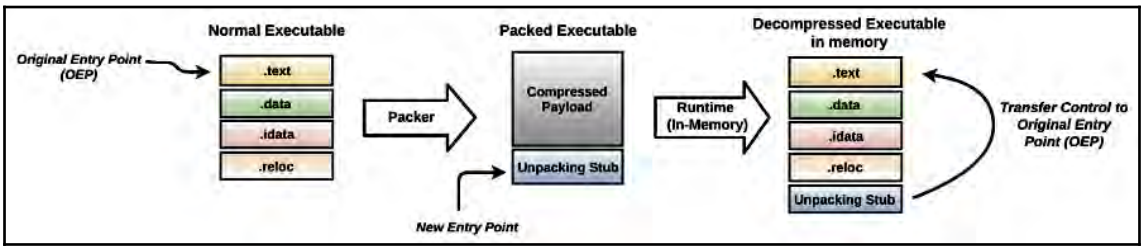
Analysis of a hostile program requires a safe and secure lab environment, as you do not want to infect your system or the production system. A malware lab can be very simple or complex depending on the resources available to you (hardware, virtualization software, Windows license, and so on). This section will guide you to set up a simple personal lab on a single physical system consisting of *virtual machines (VMs)*. If you wish to set up a similar lab environment, feel free to follow along or skip to the next section (*Section 6: Malware Sources*).

Sophos Ex. 1011
Page 8 of 9

4. Malware Unpacking

Attackers go to great lengths to protect their binary from anti-virus detection and to make it difficult for a malware analyst to perform static analysis and reverse engineering. Malware authors often use *packers* and *cryptors* (*see* `Chapter 2`, *Static Analysis, for a basic introduction to packers and how to detect them*) to obfuscate the executable content. A *packer* is a program that takes a normal executable, compresses its contents, and generates a new obfuscated executable. A *cryptor* is like a packer instead of compressing the binary; it encrypts it. In other words, a packer or cryptor transforms an executable into a form that is difficult to analyze. When a binary is packed, it reveals very less information; you will not find strings revealing any valuable information, the number of imported functions will be lower, and the program instructions will be obscured. To make sense of a packed binary, you need to remove the obfuscation layer (unpack) applied to the program; to do this, it is important to first understand the workings of a packer.

When a normal executable is passed through a packer, the executable content is compressed, and it adds an *unpacking stub* (*decompression routine*). The packer then modifies the executable's entry point to the location of the stub and generates a new packed executable. When the packed binary is executed, the unpacking stub extracts the original binary (during runtime) and then triggers the execution of the original binary by transferring the control to the *original entry point (OEP)* as depicted in the following diagram:



To unpack a packed binary, you can either use automated tools or do it manually. The automated approach saves time, but it's not completely reliable (sometimes it works and sometimes it doesn't), whereas the manual method is time-consuming, but once you acquire the skills, it is the most reliable method.

Sophos Ex. 1011
Page 9 of 9