DOI 10.34185/1562-9945-5-136-2021-06
UDC 004.056.5

S. Klishch, A. Guda, Yu. Synytsina, Yu. Mala

A PRACTICAL APPROACH TO MALWARE ANALYSIS

Abstract. Malware analysis takes significant place at the intersection of incident response, forensics, and security monitoring, and system and network administration. The reason behind performing malware analysis is to extract information from a malware sample that can assist in responding to a malware incident. From a business perspective malware analysis is critical for saving valuable data of many organizations since the control of any IT system vulnerability in the wrong hands can have unexpected consequences. In this article we will cover general practical aspects and pre requirements needed for quick start in this area.

Key words: malware, cybersecurity, malware analysis, forensics, malware analysis tools

1. Introduction

Malware Analysis is one of the unique fields in Cybersecurity [1]. To be able start in this area it is recommended to have a basic understanding of: low-level programming, operating system concepts, high-level programming languages such as C, C++, also networking and network protocols, cybersecurity in general.

2. Malware and malware analysis goals

Malware stands for Malicious Software. This can be any software or code that is used with malicious intent. The most common malicious ideas are: stealing critical information, disrupting services or organizations, obtaining unauthorized access, espionage, spam, DDoS, ransom for victim files. According to these ideas malware analysis has corresponding goals: find out how malware works, its possible detection methods, and propose measures to contain or eliminate anticipated threats [2].

Malware analysis toolkit:

- File format analyzers
- System and network monitoring tools
- Behavioral analysis tools
- Code analysis tools
- Additional tools for data manipulation: converters, editors, registry interaction tools

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

- Virtualization, containerization tools

Another one step after finishing analyzing the sample is to write a signature that can help in detection process for other systems. It is called Indicator of Compromise or IOC.

3. Malware types

There are some common malware types [3]:

1. Virus

A type of malicious software that require human action to infect a target and are often spread through email attachments and internet downloads.

2. Worm

Have something in common with viruses but has an ability to spread itself without intervention or help from the user.

3. Scareware

A type of malware that uses social engineering to trick the user into doing something unwanted, such as installing, downloading software that pretends to be antivirus, or cleaning software.

4. Ransomware

A type of malware that denies access to your system and personal information, and usually demands a payment to get your access back.

5. Botnets

A group of compromised computers that are controlled by an attacker through a command and control (C&C) channel. Often used for Distributed Denial of Service (DDoS) attacks or for performing CPU/GPU intensive tasks such as cryptocurrency mining.

6. Trojan

Software that looks and acts like a regular program but contains malicious code within it. Can be combined with a worm to make the process of harm endless.

7. Spyware

Installed on the system without user knowledge. Designed to eavesdrop, gather information.

8. Rootkit

A type of malware that is designed so that it can remain hidden on your computer.

9. KeyLogger

It is malicious software for recording computer user keystrokes to steal passwords and other sensitive information.

10. LogicBomb

A code or a piece of malicious code that is intentionally inserted into software. It is triggered only after certain conditions are met.

11. Backdoors/RAT[4]

A type of malware that attacker installs in the victim's machine without the victim's knowledge. It gives an attacker remote access into the victim's device by opening a port or establishing a remote connection in the background.

12. Dropper

A type of malware that hides another malware inside it that can be executed.

13. Downloader

A type of malware that downloads and install another malware that does the actual actions.

14. Adware

A type of malware that gives unwanted advertisements.

4. Analysis techniques

Malware analysis can be done by: static analysis, dynamic analysis.

Static analysis is done without executing malware. Basic: analyze the file properties, file structure, the functions being used, etc. Advanced: analyst tries to understand the malware by doing reverse engineering.

Dynamic analysis is done by executing malware and monitoring its behavior. There are two approaches such as basis and advanced. Basic: run malware in a virtual environment with preinstalled and configured monitoring tools. Analyze what tools output. Advanced: Run malware by using debugger. This would give more control on the malware.

There are many free resources for training with malware samples:

- https://github.com/ytisf/theZoo
- https://thezoo.morirt.com/
- https://dasmalwerk.eu/
- https://github.com/InQuest/malware-samples
- https://github.com/fabrimagic72/malware-samples
- https://www.malware-traffic-analysis.net/
- https://awesomeopensource.com/project/InQuest/malware-samples
- https://contagiodump.blogspot.com/
- https://objective-see.com/malware.html
- https://zeltser.com/malware-sample-sources/

Additional resources that require registration:

- https://virusshare.com/
- https://bazaar.abuse.ch/
- https://malshare.com/
- https://www.hybrid-analysis.com/
- https://app.any.run/

5. Acquisition tools

Evidence acquisition tools can be categorized into:

- Disk imaging tools
- Memory acquisition tools
- Other tools

There are preconfigured images (virtual machines) with these tools. It can simplify the process and help avoid mistakes such as accidentally overwriting important information when automatically mounting a hard drive.

Virtual machines:

- Windows Forensic Environment (WinFE) https://www.winfe.net/download
- SIFT Workstation https://digital-forensics.sans.org/community/downloads
- CAINE (Computer Aided Investigative Environment) https://www.caine-live.net/page5/page5.html

Disk imagining tools:

- FTK Imager https://accessdata.com/product-download

Used for data acquiring from hard or removable drives, computer memory.

- Guymager – is a free forensic imager for media acquisition.
- Magnet ACQUIRE https://www.magnetforensics.com/resources/magnet-acquire/

Help to acquire forensic images from iOS or Android devices.

Memory tools:

RAM starts fresh every time the computer is turned on. Once the system is shut down, information stored in RAM will be lost. From analyst perspective there is a lot of valuable information such as running processes, open sockets, cached passwords, etc.

1. DumpIt https://github.com/comaeio

This tool is a part of the free Comae Memory Toolkit. This tool supports from Windows XP until Windows 10 64-bits. It provides extra information during the acquisition such as displaying the Directory Table Base and the address of the debugging data structures, which are necessary parameters for memory analysis framework such as Rekall and Volatility.

2. Hibr2Bin https://github.com/comaeio

This tool enables users to uncompress Windows Hibernation file. Also supports from Windows XP up to Windows 10 x64

3. Belkasoft Live RAM Capturer https://belkasoft.com/ram-capturer

It is a free forensic tool that allows to reliably extract the entire contents of computer's volatile memory – even if protected by an active anti-debugging or anti-dumping system. Belkasoft Live RAM Capturer is compatible with all versions and editions of Windows including XP, Vista, Windows 7, 8 and 10, 2003 and 2008 Server [5].

Additional tools:

1. Kroll Artifact Parser and Extract (KAPE) https://www.kroll.com/en/services/cyber-risk/incident-response-litigation-support/kroll-artifact-parser-extractor-kape

KAPE is a free-software program that will target a device or storage location, find the most forensically important artifacts (based on your needs), and parse them within a few minutes. Because of its speed, KAPE allows investigators to find and prioritize the systems most critical for their case [6].

2. Rawcopy https://github.com/jschicht/RawCopy

It is a command line application that copy files off NTFS volumes by using low level disk reading method.

3. GRR Rapid Response https://grr-doc.readthedocs.io/en/latest/

It is an incident response framework focused on remote live forensics. It consists of a python client (agent) that is installed on target systems, and python server infrastructure that can manage and talk to clients. The goal of GRR is to support forensics and investigations in a fast, scalable manner to allow analysts to quickly triage attacks and perform analysis remotely [7].

4. Osquery https://osquery.io/

It is open source. It is used to record network connections and process executions. Windows, macOS, CentOS, FreeBSD, and almost every Linux OS released since 2011 are supported with no dependencies. Frequently, attackers will leave a malicious process running but delete the original binary on disk. This query returns any process whose original binary has been deleted, which could be an indicator of a suspicious process.

5. Static Analysis

The idea is to study as much as you can about the sample without running it. It can be done in four stages:

1. File identification and classification

File identification is the process of identifying the file type and obtaining a unique signature of the sample being analyzed. Classifying the sample can help to understand if it belongs to a well-known malware, which has been previously analyzed.

Identification based on the file types: Portable Executable (PE), PDF, DOCX, MP3. Based on file hashes: known hashes – md5, sha1, sha256 or fuzzy hash or ImpHash Based on strings embedded: strings can be encoded ASCII or Unicode.

All files can be divided in two categories such as ASCII (plain text files HTML, XML, etc.) and binary (structured files PDF, EXE).

Hashing used to classify malware as well. Hashing generates a fixed length string that referred to as a hash. The hash of the sample can be used to search for its existence within a database of known malware hashes. For example such service as VirusTotal. Two different files should not generate the same hash value. If this happened we call it a "hashing collision". If we change only one single bit of a file, it will lead to a completely different hash value. Malware that can modify itself is known as malware that is applying polymorphism. To deal with this type of malware we should use "Fuzzy hashing".

2. Scanning

A scanner makes a decision is file malicious or not based on scanning results. Offline scanners: ClamAV, Malwarebytes. Online scanners: VirusTotal, Hybrid-Analysis. From investigation perspective it is best to keep samples as much confidential as possible. For example, information disclosure actions can give attackers a signal that their activity has been detected. In this situation we should use offline scanners, or use an offline sandbox. It is software that creates an isolated or controlled environment and runs a program inside. Mostly sandbox used for mitigation a threat or a possible threat from spreading to other systems. Example of commonly used sandbox for malware analysis is Cuckoo https://cuckoosandbox.org/ sandbox. It is open source.

3. File format Analysis

Analyst should be familiar with Portable Executable (PE) file format. PE structure presented below.

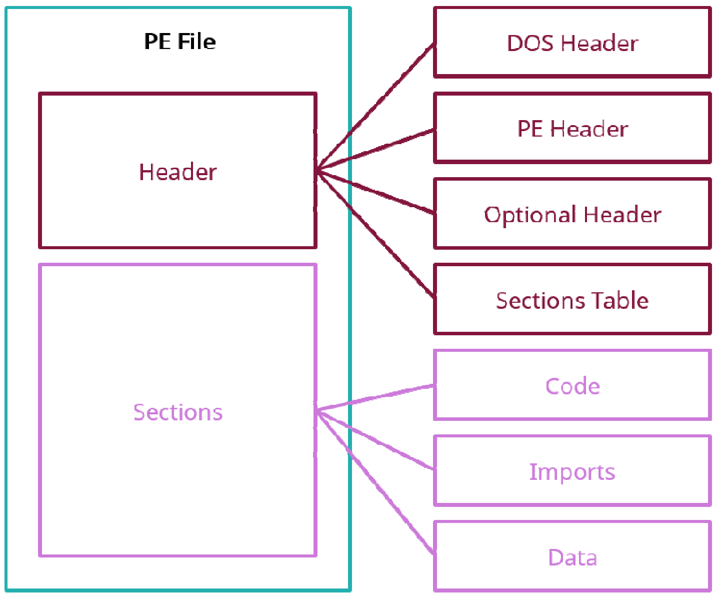

Figure 1 - PE structure

When we run any executable file or some shell program such as cmd.exe in the background being initiated many activities. The difference between programs that each one of them has requirements:

1. Memory space
2. Memory permissions
3. Place for program in memory
4. Dependencies for program to run
5. Address space to start the execution

In conclusion one executable file not always work with zero dependencies, moreover it will use some libraries in operation system. The same idea goes with malware: it also depends on functions and libraries. These interactions can be tracked, so it will help to understand what exactly does the program. There are Microsoft libraries implementation of the shared library it is called Dynamic-link Library (DLL). It helps to make program distribution easier, maintenance easier.

The file types listed below can be executed, some of them through Windows Shell (Explorer.exe), while others, such as .dll files, require another program or service to run them.

Table 1

Executable file types

Extension	Description	Extension	Description
.exe	Executable	.acm	Audio Compression Manager
.dll	Dynamic-link Library	.ocx	ActiveX Forms
.efi	Extensible Firmware Interface file. EFI files are boot loader executables	.mui	Multilingual User Interface file which contain resources that allow changing the Windows interface to display different languages.
.ax	MPEG-4 DVD Filter	.scr	Screenasaver
.cpl	Control Panel	.sys	Device driver (kernel)
.drv	Driver file (kernel)	.tsp	Windows Telephony Service Provider file

4. Identifying Obfuscation

Obfuscation is one of the methods that threat actors use in order to hide their malware. The idea is to modify the code in a way to make understanding the program more difficult while preserving its functionality. Also it is important to know that obfuscation can be used for software copyright protection or digital rights management purposes.

6. Indicators of Compromise (IOCs)

Main detection mechanisms are: signature based, anomaly and behavioral based, reputation based, hybrid based. Types of IOCs in the table below:

Table 2

Indicators of Compromise

Network-based Indicators	Host-based Indicators
IP Addresses (IPv4 or IPv6)	File Names
URLs	File Hashes
Domain Names	File Locations (paths)
Source Email Addresses	DLLs used
Email Attachments	Registry Keys
Email Message Objects	Process Handle
X509 Certificate Hashes	Etc.

A signature could include one or more IOC. To identify and classify malware samples we can use YARA https://yara.readthedocs.io/en/v3.4.0/gettingstarted.html. It is a language that helps to describe malware.

7. Dynamic Analysis

It is a process of analyzing malware samples by running them in a contained environment. For dynamic analysis it is a good idea to run the sample within a VM, and also on the computer where no any important or personal information.

There are good solutions like Shadow Defender (SD) https://www.shadowdefender.com/download.html. Shadow Defender is an easy-to-use security solution (for Windows operating systems) that protects your PC real environment against malicious activities and unwanted changes. Shadow Defender can run your system in a virtual environment called "Shadow Mode". This mode redirects each system change to a virtual environment with no change to your real environment. If you experience malicious activities or unwanted changes, perform a reboot to restore your system back to its original state, as if nothing happened [8].

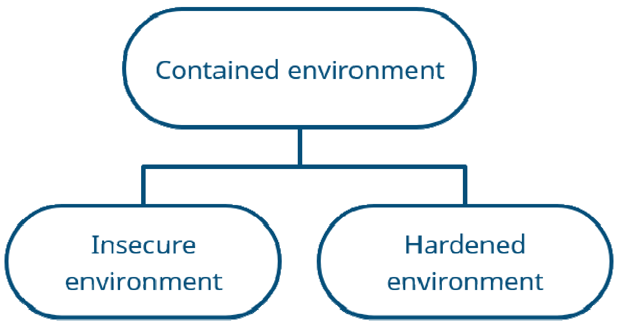

Figure 2 - Security level states of contained environment

Insecure environment lacks many of the security measures on purpose, to see how the malware acts. It is important to be able to see what the capabilities of the malware are. Also we can run the sample with high privileges to see its capabilities.

Hardened environment usually used to test the ability to withstand a successful compromise in the environment. This means that analyst wants to test how well secure is the environment if that malware sample was executed inside. This approach helps find hidden exploitable misconfigurations or vulnerabilities in the environment.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Table 3

Static versus dynamic analysis

	Methods	Notes
Static	Hashing, header analysis, file type analysis	Fast, easy
	Running scanners	Fast, easy
	PE analysis	Fast, intermediate
	Static code analysis	Slow, difficult
	Reverse engineering	Slow, difficult
Dynamic	Running malware in a Sandbox	Relatively fast, easy
	Running malware without a Sandbox	Relatively fast, intermediate
	Debugging	Slow, difficult

The process of dynamic analysis can be described as follow:

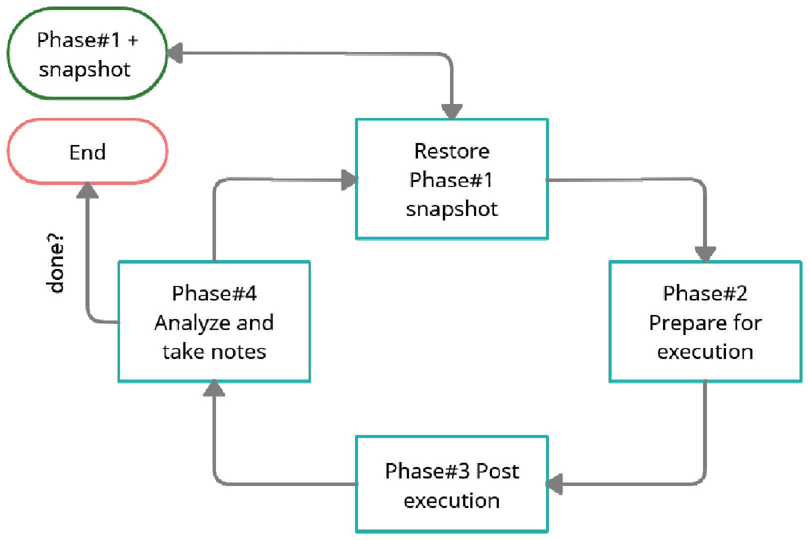

Figure 3 - Dynamic analysis phases

1. Deploy a virtual machine with the OS needed; install all the tools needed; take a snapshot of the vm.

2. Transfer malware to the created machine; prepare monitoring tools.

3. Execute the malware; start tracking malware activity; capture screenshots, copy memory dumps, config files, registry files.

4. Analyze and take notes of everything that happened.

8. Conclusion

Based on studied techniques, methods and tools that related to malware analysis process the following conclusions are obtained:

1. Malware analysis is one of the unique cybersecurity skills that every analyst should improve.

2. Static and dynamic analysis approaches each have advantages in the process of analyzing malware. But by combining the two methods will be able to provide more accurate results.

3. Each malware type has its own way of working, therefore malware analysis is an important thing to do in order to find the right steps to overcome and prevent malware attacks.

4. There are enough open source, free resources and tools to start and do progress in malware analysis field.

REFERENCES

1. 6 Common Types of Malware [Electronic resource] – Resource access mode: https://blog.totalprosource.com/5-common-malware-types.
2. What is Keylogging? [Electronic resource] // Avast Software s.r.o – Resource access mode: What is Keylogging?
3. The 11 most common types of malware [Electronic resource] // Crowdstrike. – 2020. – Resource access mode: https://www.crowdstrike.com/cybersecurity-101/malware/types-of-malware/.
4. Backdoor computing attacks [Electronic resource] // Terms of Service – Resource access mode: https://www.malwarebytes.com/backdoor/.
5. Volatile Memory Acquisition Tool: RAM Capturer [Electronic resource]. – 2016. – Resource access mode: https://vulners.com/n0where/N0WHERE:131256.
6. Kroll Artifact Parser and Extractor (KAPE) Training [Electronic resource] // Duff & Phelps – Resource access mode: https://www.kroll.com/en/insights/publications/cyber/kroll-artifact-parser-extractor-kape.
7. GRR on GitHub [Electronic resource] // GRR team. – 2019. – Resource access mode: https://grr-doc.readthedocs.io/en/latest/.
8. Xoslab.com [Electronic resource] // Xoslab.COM. – 2020. – Resource access mode: http://xoslab.com.

Received 30.04.2021.
Accepted 04.05.2021.

Практичний підхід до аналізу шкідливого програмного забезпечення

Аналіз шкідливого програмного забезпечення займає значне місце на стику реагування на інциденти, судово-медичної експертизи та моніторингу безпеки, а також системного та мережевого адміністрування. Необхідність проведення аналізу шкідливого програмного забезпечення обумовлена вилученням інформації із зразка шкідливого програмного забезпечення, яка може допомогти в реагуванні на інциденти. З точки зору бізнесу аналіз шкідливого програмного забезпечення є критичним для збереження цінних

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

даних багатьох організацій, оскільки контроль будь-якої вразливості IT-системи в чужих руках може мати непередбачені наслідки. У цій статті ми розглянемо загальні практичні аспекти та вимоги, необхідні для швидкого старту в цій галузі.

A practical approach to malware analysis

Malware analysis takes significant place at the intersection of incident response, forensics, and security monitoring, and system and network administration. The reason behind performing malware analysis is to extract information from a malware sample that can assist in responding to a malware incident. From a business perspective malware analysis is critical for saving valuable data of many organizations since the control of any IT system vulnerability in the wrong hands can have unexpected consequences. In this article we will cover general practical aspects and pre requirements needed for quick start in this area.

Кліщ Сергій Михайлович – аспірант, кафедра інформаційних технологій та систем, Національна металургійна академія України.
Гуда Антон Ігорович – д.т.н., професор, кафедра інформаційних технологій та систем, Національна металургійна академія України.
Мала Юлія Анатоліївна - доцент кафедри комп'ютерних наук та інженерії програмного забезпечення, Університет митної справи та фінансів.
Синиціна Юлія Петрівна – к.т.н., кафедра економічної та інформаційної безпеки, Дніпропетровский Держаний університет внутрішніх справ.

Клищ Сергей Михайлович – аспирант, кафедра информационных технологий и систем, Национальная металлургическая академия Украины.
Гуда Антон Игоревич – д.т.н., профессор, кафедра информационных технологий и систем, Национальная металлургическая академия Украины.
Малая Юлия Анатольевна - доцент кафедры компьютерных наук и инженерии программного обеспечения, Университет таможенного дела и финансов.
Синицына Юлия Петровна – к.т.н., кафедра экономической и информационной безопасности, Днепропетровский Государственный университет внутрених дел.

Klishch Sergey – post-graduate student, assistant, Department of information technology and systems, The National Metallurgical Academy of Ukraine.
Guda Anton – doctor of technical sciences, professor, Department of information technology and systems, The National Metallurgical Academy of Ukraine.
Mala Yuliia - associated professor of computer science and software engineering chair, University of Customs and Finance.
Synytsina Yuliia – PhD in Technical Sciences, Department of Economic and Information Security, Dnipropetrovsk State University of Internal Affairs.