MALWARE ANALYSIS

Dr. Zeyad Safaa Younus Saffawi

Malware

- **Malware** is defined as any software designed to perform actions that are harmful to the user, the host computer system, or the associated network.
- Common examples of malware include viruses, trojan horses, worms, rootkits, scareware, and spyware.
- Although malware can manifest in a wide variety of forms and behaviors, researchers and cybersecurity professionals employ a set of standardized techniques and methodologies to analyze, detect, and mitigate its effects.
- These analytical approaches often involve static and dynamic analysis, behavioral monitoring, and signature-based detection, enabling a systematic understanding of malware functionality and facilitating the development of effective defensive measures.

The Goals of Malware Analysis

- The primary objective of malware analysis is to generate actionable information that facilitates an effective response to network intrusions. Specifically, it aims to:
  - **Precisely determine the nature and scope of the security incident.**
  - **Identify and locate all compromised systems and infected files.**
  - **Assess the capabilities and behavior of a particular suspicious binary.**
  - **Develop detection mechanisms to identify the malware within the network.**
  - **Evaluate and contain the potential damage caused by the malware.**
- Once the critical files requiring comprehensive analysis are identified, signatures can be developed to detect and mitigate malware infections across the network.

Malware analysis

- **Malware analysis** can be employed to develop both **host-based** and **network-based** signatures.
- **Host-based signatures**, also referred to as **indicators**, are designed to **detect malicious activity on compromised systems**.
- These indicators typically identify files created or modified by the malware, as well as specific **alterations made to system components** such as the registry.
- Unlike traditional antivirus signatures, which rely on the inherent characteristics of the malware, host-based indicators focus on the effects of the malware on the system.
- This approach enhances detection capabilities, particularly for malware that changes its form frequently or has been removed from the hard disk.

Malware analysis

- **Network-based signatures** are utilized to detect malicious code through the monitoring and analysis of network traffic patterns.
- Although such signatures can be developed without conducting malware analysis, signatures derived from in-depth malware analysis are generally more effective, as they provide higher detection rates while significantly reducing false positives.
- After obtaining reliable host-based and network-based signatures, the final objective of malware analysis is to **achieve a comprehensive understanding of the malware's internal functionality, behavior, and operational** mechanisms.

Malware Analysis Techniques

- **Malware analysis techniques** are commonly classified into static and dynamic analysis.
- **Static analysis** involves examining malware **without executing** it to understand its structure, embedded strings, and potential functionality. Typical **tools used in static analysis** include **Virus Total, string extraction utilities, and disassemblers such as IDA Pro**.
- In contrast, **dynamic analysis** entails **executing the malware in a controlled environment to detect its behavior and impact on the system**. This process is usually performed within **virtual machines**, where **system snapshots are taken to ensure safe analysis and easy restoration**. Common **tools employed in dynamic analysis** include **RegShot, Process Monitor, Process Hacker, and CaptureBAT**.

Malware Analysis Techniques

- Additionally, **RAM analysis** is used to **examine malware behavior in memory**, leveraging tools such as **Mandiant Redline and Volatility to identify malicious processes, injected code, and memory artifacts**.

- Both **static and dynamic analysis techniques** can be further categorized into **basic** and **advanced** approaches, depending on the depth of analysis and the level of technical expertise required.

Basic Analysis

- **Basic static analysis** involves examining an executable file without inspecting its actual program instructions. This type of analysis can be used to determine whether a file is malicious and to provide preliminary insights into its potential functionality.
- In some cases, basic static analysis may also yield sufficient information to generate simple network signatures.
- Although this approach is relatively **quick and easy to perform, it has notable limitations**.
- In particular, it is often ineffective against advanced malware and may fail to reveal critical behaviors, especially when the malware employs obfuscation or packing techniques.
- **Common tools** used in basic static analysis include **VirusTotal** and **string extraction utilities**.

Basic analysis

- **Basic dynamic analysis** involves **executing the malware and observing its behavior on the system** with the aim of removing the infection, generating effective detection signatures, or achieving both objectives.
- This form of analysis allows researchers to directly monitor changes made by the malware during runtime, such as file modifications, process creation, and network activity.
- Despite its relative simplicity, basic dynamic analysis requires a secure and isolated testing environment to prevent any potential damage to the host system or network.
- Furthermore, this approach is not universally effective, as certain malware samples may evade detection or suppress malicious behavior in controlled environments, leading to the omission of critical functionality during analysis.

Advanced Analysis

- **Advanced static analysis** involves **reverse engineering** the internal structure of malware by loading the executable into a disassembler and examining its program instructions to determine its functionality.
- Since these instructions are executed directly by the CPU, advanced static analysis enables a detailed and precise understanding of the malware's behavior at the code level.
- Despite its effectiveness, advanced static analysis presents a steeper learning curve compared to basic static analysis, as it requires specialized expertise in disassembly techniques, low-level code constructs, and a deep understanding of the Windows operating system architecture.

Advanced Analysis

- **Advanced dynamic analysis** utilizes a **debugger** to examine the internal state of a running malicious executable in real time.
- This approach enables analysts to extract detailed information about the malware's execution flow, memory usage, API calls, and interaction with the operating system.
- By monitoring the malware at runtime and controlling its execution, advanced dynamic analysis provides deep insight into complex or evasive behaviors that may not be revealed through basic analysis techniques.

Types of Malware

- A **backdoor** is a type of malicious code that installs itself on a target computer to provide an attacker with unauthorized access to the system. Backdoors enable attackers to connect to compromised machines with little or no authentication, allowing them to execute commands and perform malicious activities on the local system.
- A **botnet** is similar in functionality to a backdoor in that it grants attackers remote access to infected systems. However, in a botnet architecture, multiple compromised computers are centrally controlled and receive coordinated instructions from a single command-and-control (C&C) server, enabling large-scale malicious operations.
- A **downloader** is a form of malicious code whose primary function is to download and install additional malware onto a compromised system. Downloaders are often deployed by attackers during the initial stages of system compromise, serving as a mechanism to retrieve and execute further malicious payloads.

Types of Malware

- **Information-stealing malware** is a category of malicious software designed to collect sensitive information from a victim's computer and transmit it to an attacker. Common examples include network sniffers, password hash grabbers, and keyloggers. This type of malware is typically employed to obtain unauthorized access to online services and accounts, such as email platforms and online banking systems.
- A **launcher** is a type of malicious program used to initiate the execution of other malicious components. Launchers often employ nontraditional or stealthy execution techniques to activate additional malware, thereby ensuring persistence, evasion, or elevated access within the compromised system.

Types of Malware

- A **rootkit** is a type of malicious code designed to conceal the presence of other malicious components within a compromised system. Rootkits are commonly deployed alongside other forms of malware, such as backdoors, to provide attackers with persistent remote access while making detection and removal by the victim significantly more difficult.
- **Scareware** is a form of malware intended to manipulate and intimidate users into purchasing fraudulent or unnecessary software. It typically presents a graphical user interface that mimics legitimate antivirus or security applications, falsely alerting users to nonexistent threats. The software then claims that the only way to eliminate these threats is to purchase its product, which in reality serves only to remove the scareware itself without providing genuine security functionality.

Types of Malware

- **Spam-sending malware** is a type of malicious software that compromises a user's machine and subsequently exploits it to distribute unsolicited spam messages. This category of malware generates financial gain for attackers by enabling them to offer spam-sending services through infected systems, often as part of larger malicious infrastructures such as botnets.
- A **worm or virus** is a form of malicious code capable of self-replication and autonomous propagation, allowing it to spread and infect additional computers without direct user intervention. These types of malware can cause widespread damage by rapidly exploiting network connections or removable media to proliferate across systems.

Types of Malware

- Malware can also be classified according to the attacker's intended objective, specifically as **mass** or **targeted** malware.

1. **Mass malware**, such as **scareware**, is designed to affect as many systems as possible through indiscriminate propagation. It is the most common type of malware and is generally less sophisticated, making it easier to detect and defend against, as security software is typically optimized to identify and mitigate such widespread threats.
2. **Targeted malware**, such as **backdoors**, is engineered to compromise specific systems or networks. Targeted malware poses a significantly greater risk to organizations because it is not widespread, and standard security solutions are unlikely to provide protection. Effective defense against targeted malware requires comprehensive and detailed analysis to understand its behavior, identify infections, and implement appropriate countermeasures.

- **Targeted malware** is often **highly sophisticated**, necessitating advanced analytical skills to detect, prevent, and remove infections. A notable example of targeted malware is **Stuxnet**, which demonstrated precision and complexity in its design and deployment.

General Rules for Malware Analysis

- **During malware analysis**, it is not always necessary to understand every detail of the malicious code. Instead, analysts should focus on identifying key features and behaviors that are most relevant to detection, mitigation, and response.
- **Employing multiple analysis tools** is also recommended, as the limitations of one tool may be compensated for by another. To ensure efficiency, analysts should avoid spending excessive time on a single issue and should proceed methodically when progress becomes limited.
- Furthermore, **malware authors continuously evolve their techniques in response to advancements in malware analysis**. As new analytical methods are developed, adversaries adapt by introducing new strategies aimed at evading detection and obstructing analysis, highlighting the ongoing arms race between malware developers and security researchers.

References

- Sikorski, M., & Honig, A. (2012). *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. No Starch Press.
- Bowne, S. (n.d.). *CNIT 126 Ch 0: Malware Analysis Primer & 1: Basic Static Techniques* [PowerPoint slides]. SlideShare. https://www.slideshare.net/slideshow/cnit-126-ch-0-malware-analysis-primer-1-basic-static-techniques/71040577
- Bowne, S. (n.d.). *CNIT 126: Ch 0-1 Basic Static Techniques (Part 1)* [Video]. YouTube. https://www.youtube.com/watch?v=4ZY4LC2XpFM