Digital Egypt Pioneers Initiative (DEPI)

Malware Analysis

Cyber Security Incident Response Analyst Track

Team Members:

1. *Omar Abdelrahman Ahmed*
2. *Fahd Mahmoud Abdelkhalek*
3. *Kholoud Khaled Mohamed*
4. *Nada Saleh Mohamed*
5. *Esraa Matarawy Abdelmoniem*

Supervised by:

Eng. Nour Eldin Essam

Table of Contents

1. Introduction

Malware is short for "malicious software", it is any software that one intends and develops to cause damage to the computer, server, client, or network. Some cybercriminals use malware in attempting to steal sensitive information, gain access to systems, or disrupt operations.

From simple worms to the complex malware of today, cybersecurity has always been an evolving area. It is the leading cyber threat after ransomware.

Basically, protection against malware-both for individual users and within whole organizations-relies on awareness and the sense of responsibility to take proper security measures regarding systems, platforms, and data in use. Accordingly, everybody must take proactive steps in this respect, as without such engagement, one can hardly achieve proper protection.

The following detailed report looks at the analysis of different types of malwares, their impact on systems and networks, and the methods to detect and avoid them.

2. Overview of Malware Types

Malware is classified into types based on its behavior, method of infection, and purpose. The following are some of the key categories of malware:

2.1. Viruses

Description: Viruses attach themselves to a clean file or program, often damaging or destroying it. They, themselves, cannot proliferate and depend on some user interference-such as executing infected software-to spread.

Behavior: It deletes files, corrupts files, consumes system resources, and leads to system crashes.

Propagation: Common methods include email attachments, infected USB drives, and downloads from dubious websites. Once a virus infects a system, it can use that machine to spread to others, often through shared networks.

Example: The Melissa Virus (1999), which infected email systems and crashed corporate servers due to the high volume of emails created by the virus to spread itself.

2.2. Worms

Description: Worms are self-replicating malware propagating without the need for user interaction. The propagation of worms relies on the existence of software and network vulnerabilities.

Behavior: Worms consume network bandwidth and possibly cause network congestion or even a network outage.

Propagation: Spread via networks by taking advantage of security flaws in network protocols, operating system or application vulnerabilities.

Example: The ILOVEYOU Worm (2000) infected millions of computers worldwide, crashed into e-mail systems, and resulted in US$10 billion of damages.

2.3. Trojan Horses

Description: Trojans pretend to be some form of legitimate software but are really a malicious program. They don't replicate but provide unauthorized access or run other malware programs on the system.

Behavior: Trojans can steal sensitive data, create backdoors, and give opportunities for the attacker to control systems remotely.

Propagating: Most of them spread through malicious e-mail attachments, fake software updates, or compromised websites.

Example: The Zeus Trojan (2007): used for banking credential theft using keystroke logging, causing massive losses. Restoring access requires paying the ransom, usually in cryptocurrency.

2.4. Ransomware

Description: Ransomware is a class of malware that either encrypts the files or locks users out of their system and demands an exchange-usually cryptocurrency-for return of access.

Behavior: Makes data and systems unusable until a ransom is paid.

Propagation: It spreads by way of phishing emails, malicious links, or by trying to exploit software vulnerabilities, particularly in remote desktop protocols.

Example: WannaCry Ransomware (2017) spread quickly due to an exploited vulnerability against Windows, affecting over 240,000 computers in 160 countries and causing losses of up to $5 billion.

2.5. Spyware

Description: It gathers information from an infected system in secrecy, which the user is unaware of. Examples include login credentials, tracking of keystrokes, and browsing habits.

Behavior: It monitors user activity as well as sensitive information, resulting in breaches of privacy and identity theft.

Propagation: Spyware normally infects your device through fake software bundling, phishing emails, or security vulnerabilities.

Example: Pegasus Spyware, 2017, had been used to track and collect personal information of targets, including calls and messages, through their mobile devices.

2.6. Adware

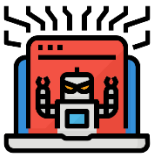
Description: Adware is a malicious program that pops up unwanted ads; sometimes it gathers user browsing data to display targeted advertisements. Generally, adware is not harmful but is intrusive and slow to the system performance.

Behavior: It pops up ads, slows down the performance of a system, and sometimes redirects browsers to malicious sites.

Propagation: It is mostly installed with free software installation or is bundled into some legitimate-appearing downloads.

Example: Fireball Adware (2017) hijacked web browsers and generated revenue out of the fake online traffic.

2.7. Rootkits



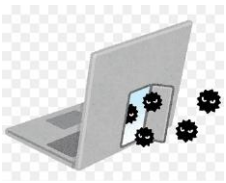
Description: The rootkit provides an attacker with administrative-level access to the system and hides the very existence of the malicious program itself. They are also capable of editing core system functions to disguise other malicious activities of the attacker.

Behavior: Rootkits allow unauthorized access to a system and can't be detected by antivirus.

Propagation: It is mostly installed through Trojans or some other malware that receives administrative privilege.

Example: Sony BMG Rootkit (2006) tracked secretly the media use of users, compromising the security of millions of systems.

2.8. Backdoors



Description: A backdoor provides illegitimate access to an attacker to a system or network bypassing normal authentication procedures.

Behavior: Allows sustained remote accessibility to infected systems, which are often used in conjunction with other malware to reach permanent control.

Propagation: Generally installed as part of another malware or by exploiting security weaknesses.

Example: The Back Orifice Tool (1998) was used to gain a remote control over infected Windows systems.

3. Detection Methods

Detection methods are crucial for identifying and mitigating malware threats.

3.1. Signature-based Detection

Detection Technique: Signature-based detection is the most traditional method. It identifies malware by comparing a file's binary structure or unique signatures (e.g., hash values) against a database of known malware signatures. Each signature is a unique code pattern associated with a specific piece of malware.

Tools:

- **ClamAV**: Open-source antivirus engine.
- **Kaspersky**, **Norton**: Commercial antivirus solutions with extensive signature databases.
- **Virustotal**: An online platform that checks files and URLs against multiple antivirus engines.

How It Works: When a file enters the system, the antivirus scans its code and compares it against an internal signature database. If the code matches a known signature, the system flags it as malware and typically isolates or removes the file.
Pros:

- **Efficiency**: Fast and accurate for identifying well-known malware.

- **Low resource usage**: Because it simply matches patterns, it requires minimal system resources.

Cons:

- **Inability to detect zero-day threats**: Signature-based systems cannot identify new or unknown malware that doesn't have a known signature.
- **Frequent updates needed**: The database must constantly be updated with new malware signatures, making it reactive rather than proactive.

Best Use Case:

- **Well-established malware detection**: Suitable for identifying old or well-known malware, typically in environments with legacy systems or basic threat protection needs.

3.2. Behavior-based Detection (Anomaly Detection)

Detection Technique: This method focuses on identifying unusual or suspicious behaviors in applications and system processes. It monitors real-time activities such as file modifications, network connections, and system resource consumption. When abnormal behavior is detected—such as a program attempting to escalate privileges or modify system files—the system flags it as suspicious.

Tools:

- **CrowdStrike Falcon**: Monitors real-time behavior and can quickly detect anomalous activities.
- **CylancePROTECT**: Focuses on detecting malicious behavior without needing a signature database.

How It Works: Behavior-based detection does not rely on predefined malware signatures. Instead, it creates a baseline of normal activities on a system or network. Any deviation from this norm, such as unusual network traffic or changes to critical system files, triggers an alert. For example, ransomware encrypting files will trigger the system to detect this behavior as abnormal.

Pros:

- **Effective against zero-day and file-less malware**: It can detect unknown malware based on its actions rather than its code.

- **Real-time threat detection**: Provides immediate responses to potentially malicious activities.

Cons:

- **False positives**: Legitimate software may be flagged as suspicious if it behaves in an unusual way, leading to unnecessary alerts.
- **Resource-intensive**: Monitoring the system in real-time can use significant system resources and impact performance.

Best Use Case:

- **Advanced Threat Protection (ATP)**: Ideal for environments that need protection from sophisticated attacks, such as zero-day malware, file-less threats, and advanced persistent threats (APTs).

3.3. Static Analysis

Detection Technique: Static analysis involves examining the malware without executing it. This method inspects the binary code, file structure, headers, and other properties of the file to identify any malicious characteristics. It checks for things like packed or obfuscated code, suspicious API calls, or strange imports that could signal malicious intent.

Tools:

- **IDA Pro**: A powerful disassembler that provides deep insights into malware binaries.
- **Radare2**: An open-source framework for reverse engineering and analyzing binary files.
- **PEiD**: A tool that identifies packers and cryptors used to hide malware.

How It Works: Analysts or automated tools analyze a file's static properties—such as examining the code for known malicious patterns, strings, or signatures—without running it. For example, a file might have encrypted sections or make calls to dangerous APIs. Analysts can detect these issues by carefully inspecting the file structure.

Pros:

- **No risk of execution**: Since the file is never run, there's no risk of infecting the system during analysis.

- **Detailed code inspection**: Allows for deep inspection of a file's code, helping identify potential threats.

Cons:

- **Cannot detect runtime behavior**: Some sophisticated malware alters its behavior during execution (e.g., polymorphic malware), which static analysis cannot detect.
- **Requires expertise**: It can be time-consuming and complex, requiring skilled analysts to perform effectively.

Best Use Case:

- **Malware Reverse Engineering**: Ideal for analyzing malware samples without executing them, especially when conducting forensic investigations or developing malware signatures.

3.4. Dynamic Analysis

Detection Technique: Dynamic analysis (also called runtime analysis) observes how a file behaves when executed in a controlled environment. It watches for malicious actions such as file encryption, unauthorized data access, or network connections. This method is useful for detecting malware that hides its true nature until it runs.

Tools:

- **Cuckoo Sandbox**: A popular open-source sandbox environment used to execute and analyze malware in isolation.
- **Process Monitor (Sysinternals)**: A tool for real-time system monitoring of file, registry, and process activities.

How It Works: The suspected malware is executed in a sandbox, which is an isolated environment that mimics the target system but does not affect the actual network or system. As the malware runs, dynamic analysis tools log its actions, such as modifying system files, accessing sensitive data, or attempting to communicate with command and control servers.

Pros:

- **Detects runtime behavior**: Can uncover malicious activities that static analysis might miss, such as dynamic code execution or network attacks.

- **Effective against obfuscated malware**: Many malware families attempt to disguise their code, but dynamic analysis can detect their actual behavior during execution.

Cons:

- **Time and resource-intensive**: Running files in a sandbox or dynamic analysis environment requires more system resources and time.
- **Evasion techniques**: Some malware can detect when it's being run in a sandbox and modify its behavior to avoid detection.

Best Use Case:

- **Malware Behavior Analysis**: Ideal for understanding how malware behaves in real-world execution, especially for advanced threats that use polymorphism or code obfuscation.

3.5. Reputation-based Detection

Detection Technique: Reputation-based detection focuses on assessing the trustworthiness of a file or URL by comparing it to a vast database of known malicious and benign files. It uses attributes like file hashes, file size, origin, and digital certificates to score the file's reputation.

Tools:

- **Symantec Insight**: A reputation-based security tool that assesses files based on their prevalence and origin.
- **Norton Safe Web**: Checks URLs for safety by assessing their reputation against known bad sites.

How It Works: A file or URL is checked against a cloud database or community-reported threat intelligence platform. The database contains a history of previous interactions with the file or URL and gives it a reputation score. Files with poor reputation (e.g., those from suspicious origins or rarely seen by other users) are flagged as malicious or suspicious.

Pros:

- **Quick and lightweight**: It doesn't require in-depth analysis of the file's behavior or code and is fast at detecting previously flagged threats.

- **Effective against known threats**: Ideal for environments that require fast detection of well-known threats or phishing sites.

Cons:

- **Ineffective against brand-new malware**: If a file has no history or reputation, it may not be flagged despite being dangerous.
- **Relies on external sources**: Requires an up-to-date and reliable reputation database, often from external sources.

Best Use Case:

- **Web and file downloads protection**: Best for quickly assessing the safety of files or links before allowing user interaction, particularly in enterprise or user-driven environments.

3.6. Hybrid Detection

Detection Technique: Hybrid detection combines two or more techniques to offer comprehensive malware detection. For instance, it may merge signature-based detection with behavior-based monitoring, or static analysis with dynamic analysis. By combining methods, hybrid detection can detect a wider variety of malware, including zero-day threats.

Tools:

- **Palo Alto WildFire**: Combines static and dynamic analysis, as well as behavioral detection, to uncover sophisticated threats.
- **Bitdefender GravityZone**: Employs hybrid techniques like signature-based scanning, heuristic analysis, and behavior monitoring.

How It Works: A hybrid system first attempts to identify malware using quick methods, such as signature-based detection. If the file is unknown or shows suspicious behavior, it will trigger further in-depth analysis using behavioral or dynamic analysis. The system dynamically switches between methods to maximize detection rates and minimize system load.

Pros:

- **High detection accuracy**: Combining multiple techniques allows for both quick detection of known threats and deep analysis of unknown ones.
- **Flexible and adaptive**: It can

Flexible and adaptive: It can adapt its detection method based on the nature of the threat, using different techniques for different malware types.

Cons:

- **Resource-intensive**: Since it may use multiple methods in tandem, hybrid detection can be more demanding in terms of processing power and memory.
- **Complex to implement**: Creating a balanced hybrid detection system requires expertise to ensure the right combination of techniques and minimize false positives or system slowdowns.

Best Use Case:

- **Advanced Threat Protection for Enterprises**: Ideal for businesses that need comprehensive protection against a wide array of threats, including zero-day vulnerabilities, fileless malware, and advanced persistent threats (APTs).

4. Impact Analysis

Malware impacts the organizations and individuals regarding system performance, network integrity, data confidentiality, and user experience. Understanding the impact is essential in developing effective means for prevention and response.

4.1. System-Level Impact

Performance Degradation: Malware hijacks system resources to cause slow system performance. CPU usages increase, malicious processes consume memory, and system responsiveness is reduced.

File Corruption: Most types of malwares, especially viruses and ransomware, often corrupt or destroy critical files, making them unusable or irretrievable sans backups.

System Instability: Malware causes frequent system crashes or forced reboots. These disrupt user activity and business operations.

4.2. Network-Level Impact

Bandwidth Consumption: Worms and some types of malwares replicate and proliferate across the network. It does so by consuming bandwidth, hence reducing overall network performance.

Unauthorized Access: Malware of types like Trojans or backdoors provide unauthorized access to sensitive networks to the attacker, thus enabling them to exfiltrate data, manipulate systems, or further spread infections.

Network Disruption: Most malware variants are designed to conduct DDoS attacks through infecting hundreds of devices to overwhelm target networks, a process often known as botnets.

4.3. Data Impact

Data Theft: Spyware and backdoors steal sensitive information related to finance, personal identity, intellectual property, or trade secrets.

Data Corruption/Deletion: Viruses and worms may cause corruption or permanent deletion of some valuable data. For instance, ransomware encrypts data and then asks for a ransom in exchange for decryption.

Data Privacy Violations: Malware collecting data without consent can lead to severe breaches of data privacy, exposing sensitive personal or business information.

4.4. User Impact

Financial Loss: Ransomware can cause the loss of great financial resources by either paying ransoms or causing data loss and downtime for the user or organization concerned.

Identity Theft: Spyware would normally target personal information to be used in identity theft or fraud. In turn, stolen credentials could be sold in the dark web or be leveraged in unauthorized transactions.

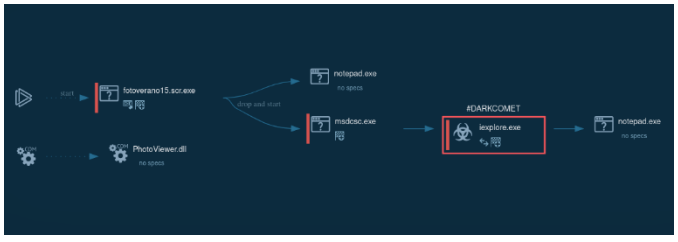
Productivity Loss: System slowdowns and crashes, data unavailability-these are productivity losses, especially in organizations where uptime means business.

5. Case Studies

5.1. Case Study 1: “DarkComet”

Type: Remote Access Trojan (RAT).

Impact: Enabled attackers to gain complete control over the victim's system, allowing for keystroke logging, screenshot capture, and webcam access.



Static Analysis:

- ✓ Using tools like "Strings," we can extract hardcoded URLs or suspicious IP addresses intended for command-and-control (C2).
- ✓ Disassembling with IDA Pro reveals functions used for process manipulation and methods for maintaining persistence through registry changes.

Dynamic Analysis:

- ✓ In a sandbox environment, "DarkComet" connects to its C2 server, transmits system data, and attempts to disable security measures.
- ✓ Analyzing network traffic shows encrypted communication over specific ports.

Consequences: Infected systems were often repurposed into botnets, utilized for launching distributed denial-of-service (DDoS) attacks or spying on users.

5.2. Case Study 2: "CryptoLocker"

Type: Ransomware.

Impact: Encrypted user files, demanding a ransom in Bitcoin for decryption.

Static Analysis:

- ✓ Disassembly indicates the use of robust RSA encryption to lock files.
- ✓ Inspection of the binary shows the creation of a ransom note, typically displayed post-encryption.

Dynamic Analysis:

- ✓ Executing "CryptoLocker" in a virtual environment reveals its immediate scanning of local drives for target file types (.docx, .pdf, etc.).
- ✓ It encrypts files using an RSA public key and contacts a remote server to retrieve the private key.

Consequences: Victims faced limited choices: pay the ransom or restore data from backups, often resulting in significant financial repercussions.

6. Conclusion

Understanding malware types, their various impacts, ways of effective detection, and prevention strategies helps an organization reduce its potential for vulnerability to some specific kinds of cyberattacks and strengthens its cybersecurity posture.

Any organization should be focusing on a multilayered security approach, including routine updates, exceptional antivirus solutions, training employees, and adequate monitoring systems. Besides, the development of a solid incident response and recovery plan means that, should an intrusion occur, damage will be minimal, and operations can return to normal as soon as possible.