MALWARE DETECTION

UNDER SUPERVISION OF: DR. ALI AL-SHARBINY

BY ESRAA MAHMOUD HAMZA

ID: 4211250

CONTENT

INTRODUCTION

What is Malware?

- Malware (malicious software) is designed to harm or exploit systems and data.
- Examples: Viruses, Trojans, Worms, Ransomware, Spyware, Adware.

Importance of Malware Detection:

- Protect sensitive data and systems.
- Prevent financial and operational losses.

Types of Malware Detection

- **Static Detection**: Analyzes files without executing them.

Techniques:

  - Signature Matching: Matches file patterns with known malware.
  - Hash Analysis: Uses unique file identifiers for detection.

- **Dynamic Detection:** Observes file behavior during execution.

Techniques:

  - Sandbox: Executes files in a controlled environment.
  - Behavior Monitoring: Tracks suspicious activities.

LINKING MALWARE DETECTION TO THE PROJECT

- **Scope of the Project:**

Implements Static Detection through Hash Matching.

- **Approach:**

Compare file hashes against a predefined database of known malware hashes.

- **Objective:**

Provide a simplified and efficient way to malware in files and folders.

PROJECT OVERVIEW

- **Key Features:**
  - Calculates MD5, SHA-1, and SHA-256 hashes for files.
  - Compares file hashes with a known malware hash database.
  - Scans all files in a specified folder for classification.
- **Workflow Explanation:**
  a. Input a folder path for scanning.
  b. Compute hashes for each file in the folder.
  c. Classify files as "Safe" or a specific type of malware based on hashes.
  d. Output the results for each file.

Hashing and Its Role

What is Hashing?

- A process to convert data into a fixed-size alphanumeric value.
- Unique for every distinct file, like a digital fingerprint.

How It Works in Malware Detection:

- Generate a hash for each file using hashing algorithms (MD5, SHA-1, SHA-256).
- Compare the generated hash with a database of known malware hashes.

File Classification

- **How the Program Classifies Files:**

1. Generate hashes for the file (MD5, SHA-1, SHA-256).
2. Check each hash against a predefined malware classification dictionary.
3. If a match is found, classify the file as a specific malware type.
4. If no match is found, classify it as "Safe."

- **Safe vs. Malware Classification:**
- Example:
  - KnownMalwareHash: 5e884898da28047151d0e56f8dc62927... → Ransomware.
  - No Match Found: File is "Safe."

FOLDER SCANNING PROCESS

How It Works:

- Input the folder path to be scanned.
- List all files in the folder.
- Compute hashes for each file and classify them.
- Display the classification result for each file.

Outputs and Insights:

- Example Output:
- File1.exe → Classified as "Virus."
- File2.pdf → Classified as "Safe."

Strengths and Limitations

Strengths:

- Lightweight and simple to implement.
- Uses reliable hashing techniques.
- Scalable with an updated malware hash database.

Limitations:

- Relies only on static detection (hash matching).
- Cannot detect new malware variants without updated hashes.
- Limited to the predefined database of hashes.

Conclusion and Future Enhancements

Summary of Contributions:

- Provides an efficient static malware detection tool.
- Simplifies the process of classifying files based on their hashes.

Future Enhancements:

- Add dynamic detection capabilities.
- Implement automated updates for the malware hash database.
- Introduce a graphical interface for ease of use.

THANK YOU