**Name:** Chirag Maheshwari

**Class:** MSc (Cyber Security & Digital Forensics) 3rd Semester

**Title: Hash-Based Signature Detection with File Change Monitoring for Malware Detection and Integrity Verification**

**Introduction:**

In today’s digital landscape, the increasing prevalence of malware and unauthorized modifications to critical files poses significant challenges to system security. Hash-based signature detection is a proven method to identify malware, while file integrity monitoring ensures that unauthorized changes to files can be promptly detected. This project aims to combine these two approaches into a single tool that not only detects malware but also monitors file integrity by comparing file hashes between scans. This dual-purpose solution will enhance both security and system integrity.

**Objectives:**

1. To develop a Python-based tool that identifies malware by comparing file hashes against a database of known malware signatures.
2. To implement a file integrity monitoring mechanism that tracks file changes by comparing current file hashes with previously stored ones.
3. To create a system that alerts users to potential malware infections or unauthorized file modifications based on hash mismatches.

**Methodology:**

1. **Hash-Based Signature Detection**:
   * The tool will use **SHA-256 hashing** to detect known malware. It will compare the hashes of files within a specified directory against a pre-existing database of malware hashes stored in a CSV file.
   * If any file matches a known malware signature, an alert will be triggered, identifying the infected file.
2. **File Integrity Monitoring**:
   * During the first run, the system will generate hashes for all files within a target directory and store them in a CSV file (file\_hashes.csv).
   * In subsequent runs, the tool will compare the newly generated hashes with the previously stored hashes to detect if any files have been modified, added, or deleted.
3. **CSV File Management**:
   * **malware\_hashes.csv**: This file will store known malware signatures (file hashes) and their corresponding malware names.
   * **file\_hashes.csv**: This file will store file paths and their respective SHA-256 hashes, enabling file integrity comparison between scans.

**System Requirements:**

* **Programming Language**: Python 3.x
* **Libraries**:
  + hashlib: To compute SHA-256 hashes of files.
  + os: To traverse the file system and scan directories.
  + csv: To read/write malware signatures and file hashes for comparison.

**Detailed Workflow:**

1. **Initial Scan**:
   * The tool will scan the specified directory and compute the SHA-256 hash for every file.
   * The computed file hashes will be saved in the file\_hashes.csv for future reference.
   * Malware detection will be performed by comparing each file hash with the entries in malware\_hashes.csv.
2. **Subsequent Scans**:
   * For subsequent runs, the tool will:
     + Compare each file's hash to the pre-existing malware\_hashes.csv to identify any known malware.
     + Compare the current file hashes to those stored in file\_hashes.csv to check for file modifications.
     + Notify the user of any file changes, such as added, deleted, or modified files.
3. **Output**:
   * The system will generate an alert for any file identified as malware.
   * It will also log and notify the user of any files that have changed since the previous scan, helping identify unauthorized modifications.

**Expected Outcomes:**

* A working tool capable of identifying known malware based on hash signatures.
* A file integrity monitoring system that tracks changes and detects unauthorized file modifications.
* Improved system security through the combination of malware detection and file integrity verification.

**Applications:**

* **Malware Detection**: The tool can be used in various environments, including personal computers and servers, to detect malware based on known signatures.
* **File Integrity Monitoring**: The project can be employed in sensitive environments where maintaining the integrity of files is critical, such as in corporate networks, research labs, and digital forensics.

**Conclusion:**

The proposed tool will integrate hash-based malware detection and file integrity monitoring into a single solution, enhancing security and preventing unauthorized file changes. This combination of features makes the tool a valuable asset for cybersecurity professionals and system administrators, allowing them to maintain both system integrity and security effectively.

**References:**

* SHA-256 hashing algorithm for integrity verification and malware detection.
* Signature-based detection techniques used in cybersecurity.