File Integrity Monitor

Description:

The File Integrity Monitor is a Python-based system that monitors the integrity of files on a computer system. It detects unauthorized changes or modifications to files, ensuring the security and integrity of the system.

Dependencies:

- Python 3.x

- hashlib module (standard library)

Usage

1. Update the `TARGET_FILES` list in the code with the file paths you want to monitor for integrity. These files can include critical system files, configuration files, or any other files of interest.

2. Set the `CHECK_INTERVAL` parameter to specify the time interval (in seconds) between integrity checks. This determines how frequently the files will be monitored for modifications.

3. Run the script using Python: `python file_integrity_monitor.py`

Algorithm:

The file integrity monitor follows the following algorithm:

1. Initialize the system by computing the initial hash values for the target files and storing them in a dictionary.

2. Enter a monitoring loop that periodically checks the integrity of the target files.

3. For each file, compute the current hash value and compare it with the stored hash value.

4. If a file modification is detected (i.e., the hashes differ), perform appropriate actions such as logging the event or sending notifications.

5. Update the stored hash value for the modified file with the new computed hash.

6. Repeat the monitoring loop at the specified interval until the monitoring process is stopped.

Functions:

The project includes the following functions:

- `compute_file_hash(file_path)`: Computes the hash value (SHA-256) of a given file.

- `initialize_fim()`: Initializes the file integrity monitor by computing and storing the initial hashes of the target files.

- `monitor_files()`: Monitors the integrity of target files by periodically checking for modifications.

Error Handling:

- If a target file does not exist during initialization or monitoring, a warning message is displayed.

Example:

```python
import hashlib
import os
import time
```

```
#Configuration parameters
TARGET_FILES = [r'C:\path\to\file1.txt', r'C:\path\to\file2.txt']  # List of files to monitor
CHECK_INTERVAL = 1  # Time interval between integrity checks


# ... rest of the code ...
```

In this example, the file integrity monitor is set up to monitor two files: `file1.txt` and `file2.txt` located at specified paths. The integrity of these files will be checked every 1 second.

Note: Make sure to replace the file paths in the `TARGET_FILES` list with the actual paths you want to monitor.

### Limitations

- The file integrity monitor assumes that it has appropriate access privileges to read and monitor the target files.

- The code provided is a simplified example and may require modifications to suit specific requirements or handle additional scenarios.

### Conclusion

The File Integrity Monitor provides a basic framework for monitoring the integrity of files in a computer system. It can be extended and customized to fit the needs of different applications and environments. By regularly checking file integrity, the system helps ensure the security and reliability of critical files.