Secure File Transfer Monitoring System
(Project Documentation)

1. Project Overview / Description

This project focuses on developing a Secure File Transfer Monitoring System designed to ensure data confidentiality and track file movement across a system or network.

File transfers—both internal and external—pose significant risks including data leakage, unauthorized access, malware distribution, and insider misuse.

This monitoring system provides:

- File transfer logging

- Unauthorized file movement detection

- File integrity verification

2. Practical Motivation

Organizations face constant threats involving unauthorized file uploads/downloads, data theft, and malicious file tampering.

Examples include:

- Employees copying sensitive data outside the organization

- Malware modifying or replacing files

- Unauthorized transfers via USB, network shares, or cloud sync tools

A monitoring system helps detect:

- Suspicious file transfers

- Unauthorized data movement

- Integrity violations

- Potential exfiltration attempts

This project provides hands-on defensive monitoring experience used in SOC and digital forensics.

3. Project Objectives

1. Log all file transfers performed on the system.

2. Detect unauthorized movement of sensitive or restricted files.

3. Implement file integrity checks using hashing (SHA256/MD5).

4. Generate alerts on policy violations.

5. Produce detailed audit logs and security reports.

4. Practical Scope of the Project

A. File Transfer Logging:

- Monitor file copy, move, delete, upload, download events.

- Log timestamp, source path, destination path, user, and process name.

B. Unauthorized Movement Detection:

- Maintain a list of sensitive directories or restricted files.

- Trigger alerts when such files are moved or accessed without permission.

- Detect suspicious outbound transfers (USB, network shares, cloud folders).

C. File Integrity Checks:

- Calculate pre- and post-transfer hash values.

- Detect tampering, corruption, or unauthorized modifications.

- Highlight mismatches in integrity.

D. Reporting & Alert System:

- Generate logs for all file events.

- Highlight violations and suspicious transfers.

- Produce a final audit report summarizing activity.

5. Tools & Technologies Used

Programming Languages:

- Python (recommended)
- PowerShell (optional)

Modules/Tools:

- watchdog (filesystem event monitoring)
- hashlib (for hashing and integrity verification)
- psutil (optional process tracking)
- win32api / PowerShell Get-ChildItem (optional for Windows)

Documentation Tools:

- Word / Google Docs
- Draw.io for architecture diagrams

6. Practical Techniques Implemented

Security Techniques:

- File system activity monitoring
- Tamper detection through hashing
- Unauthorized access alerting
- Sensitive data movement tracking

Blue Team Techniques:

- Detecting insider threats
- Monitoring suspicious data transfers
- Identifying modified or replaced files
- Strengthening data loss prevention (DLP) strategies

7. Workflow / Architecture (Practical Explanation)

STEP 1: Monitor File System

- Detect copy, move, delete, and modification events.

STEP 2: Classify Event

- Identify whether the event involves sensitive or normal files.

STEP 3: Integrity Hashing

- Compute hash before and after transfer.

STEP 4: Authorization Check

- Validate whether the event is allowed or suspicious.

STEP 5: Logging & Alerting

- Record event details and trigger alerts if necessary.

STEP 6: Final Reporting

- Provide a comprehensive audit log and summary report.

8. Flowchart (Text Version)

START

↓

Monitor File System Events

↓

Is File Sensitive? → Yes

↓

Run Hash & Authorization Check

↓

Is Movement Authorized? → Yes → Log Event

↓ No

Generate Alert + Log

↓

Create Final Audit Report

↓

END

9. Expected Practical Output

The system should output:

- Detailed file activity logs
- Unauthorized movement alerts
- Integrity check results
- Detection of abnormal file transfers
- Final audit summary

Examples:

- Alert: Sensitive file copied to USB drive
- Integrity Failure: File hash mismatch detected
- Suspicious movement: 200 files transferred to unknown directory

10. Learning Outcomes

This project teaches:

- How file systems handle transfers and modifications
- Data loss prevention concepts
- Hash-based integrity checking
- How monitoring improves defensive posture

- Real-world file auditing techniques

11. Project Deliverables

1. Project documentation (Word/PDF)

2. File transfer monitoring toolkit

3. Logs/screenshots of monitoring activity

4. Integrity check evidence

5. Flowcharts & architecture diagrams

6. Final presentation (PPT)