Secure File Transfer Monitoring System

Student Name: Aneesa Fathima
Project Title: Secure File Transfer Monitoring System
Date: *09-01-2026*

---

1. Project Overview

This project implements a **Secure File Transfer Monitoring System** designed to continuously monitor file system activity, detect unauthorized file movement and integrity violations, and generate detailed audit logs.

File transfers — whether internal or external — can result in data leakage, unauthorized access, malware distribution, and insider misuse. This project focuses on defensive monitoring techniques used in Security Operations Centers (SOC) and digital forensics environments.

---

2. Practical Motivation

Organizations face ongoing threats from:

- Unauthorized data exports
- Malware tampering with critical files
- Suspicious uploads/downloads via USB or cloud services

A robust monitoring system helps:

✔ Detect suspicious file transfers
✔ Log detailed event information
✔ Alert security analysts in real time
✔ Maintain integrity through hashing
✔ Generate audit reports for investigations

---

3. Project Objectives

This system is designed to:

1. Log all file transfers and modification events.
2. Detect unauthorized movement of sensitive files.

3. Perform file integrity verification with hashing.
4. Generate alerts on policy violations.
5. Produce structured audit logs and summary reports.

---

4. Tools & Technologies

Category	Tool / Language
Programming	Python3
Event monitoring	`watchdog`
Hashing	`hashlib`
Optional process info	`psutil`
Documentation	MS Word / Google Docs
Diagrams	Draw.io

---

5. Technical Approach & Workflow

5.1 Monitoring File System Events

The system uses the `watchdog` Python module to capture file system events:

- **created**
- **modified**
- **deleted**
- **moved**

Each event records:

```
timestamp | file path | event type | hash
```

```
Last login: Fri Jan  2 20:57:42 on ttys001
aneesafathima@Riazs-MacBook-Pro ~ % cd ~/Documents
aneesafathima@Riazs-MacBook-Pro Documents % mkdir Secure-File-Transfer-Monitoring
mkdir: Secure-File-Transfer-Monitoring: File exists
aneesafathima@Riazs-MacBook-Pro Documents % cd Secure-File-Transfer-Monitoring
aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % mkdir logs screenshots docs utils tests
mkdir: logs: File exists
mkdir: screenshots: File exists
mkdir: docs: File exists
mkdir: utils: File exists
mkdir: tests: File exists
aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % touch README.md requirements.txt main.py config.json
aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % touch utils/hashing.py utils/logger.py utils/alerts.py
aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % python3 --version
Python 3.7.9
aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % python3 -m venv venv
aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % source venv/bin/activate
(venv) aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % pip install watchdog psutil
Collecting watchdog
  Using cached watchdog-3.0.0.tar.gz (124 kB)
Collecting psutil
  Using cached psutil-7.2.1.tar.gz (490 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
    Preparing wheel metadata ... done
Using legacy setup.py install for watchdog, since package 'wheel' is not installed.
Building wheels for collected packages: psutil
  Building wheel for psutil (PEP 517) ... done
  Created wheel for psutil: filename=psutil-7.2.1-cp36-abi3-macosx_10_9_x86_64.whl size=127961 sha256=bf3f0c2c753a1352b2ae99397ba076f5438e49d1cb
  Stored in directory: /Users/aneesafathima/Library/Caches/pip/wheels/88/1f/d7/e85e4de6bd1dab91f8c277422623160607f00e5516667e8267
Successfully built psutil
Installing collected packages: watchdog, psutil
    Running setup.py install for watchdog ... done
Successfully installed psutil-7.2.1 watchdog-3.0.0
WARNING: You are using pip version 20.1.1; however, version 24.0 is available.
You should consider upgrading via the '/Users/aneesafathima/Documents/Secure-File-Transfer-Monitoring/venv/bin/python3 -m pip install --upgrade
(venv) aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % pip freeze > requirements.txt
(venv) aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % nano main.py
(venv) aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % python3 main.py
Traceback (most recent call last):
  File "main.py", line 35, in <module>
    config = json.load(f)
  File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/json/__init__.py", line 296, in load
    parse_constant=parse_constant, object_pairs_hook=object_pairs_hook, **kw)
  File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/json/__init__.py", line 348, in loads
    return _default_decoder.decode(s)
  File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/json/decoder.py", line 337, in decode
    obj, end = self.raw_decode(s, idx=_w(s, 0).end())
  File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/json/decoder.py", line 355, in raw_decode
    raise JSONDecodeError("Expecting value", s, err.value) from None
json.decoder.JSONDecodeError: Expecting value: line 1 column 1 (char 0)
(venv) aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % nano main.py
(venv) aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % nano hashing.py
(venv) aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % nano logger.py
(venv) aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % nano alerts.py
(venv) aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /Users/aneesafathima/Documents/Secure-File-Transfer-Monitoring/.git/
(venv) aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % git status
On branch master

No commits yet
```

```
[No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .DS_Store
        README.md
        alerts.py
        config.json
        hashing.py
        logger.py
        main.py
        requirements.txt
        utils/
        venv/

nothing added to commit but untracked files present (use "git add" to track)
(venv) aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % git add .
(venv) aneesafathima@Riazs-MacBook-Pro Secure-File-Transfer-Monitoring % git commit -m "Initial commit: Secure File Transfer Monitoring System"
[master (root-commit) d84e0cb] Initial commit: Secure File Transfer Monitoring System
 1044 files changed, 173755 insertions(+)
 create mode 100644 .DS_Store
 create mode 100644 README.md
 create mode 100644 alerts.py
 create mode 100644 config.json
 create mode 100644 hashing.py
 create mode 100644 logger.py
 create mode 100644 main.py
 create mode 100644 requirements.txt
 create mode 100644 utils/alerts.py
 create mode 100644 utils/hashing.py
 create mode 100644 utils/logger.py
 create mode 100644 venv/bin/activate
 create mode 100644 venv/bin/activate.csh
 create mode 100644 venv/bin/activate.fish
 create mode 100755 venv/bin/easy_install
 create mode 100755 venv/bin/easy_install-3.7
 create mode 100755 venv/bin/pip
 create mode 100755 venv/bin/pip3
 create mode 100755 venv/bin/pip3.7
 create mode 120000 venv/bin/python
 create mode 120000 venv/bin/python3
 create mode 100755 venv/bin/watchmedo
 create mode 100644 venv/lib/python3.7/site-packages/__pycache__/easy_install.cpython-37.pyc
 create mode 100755 venv/lib/python3.7/site-packages/_watchdog_fsevents.cpython-37m-darwin.so
 create mode 100644 venv/lib/python3.7/site-packages/easy_install.py
 create mode 100644 venv/lib/python3.7/site-packages/pip-20.1.1.dist-info/INSTALLER
 create mode 100644 venv/lib/python3.7/site-packages/pip-20.1.1.dist-info/LICENSE.txt
 create mode 100644 venv/lib/python3.7/site-packages/pip-20.1.1.dist-info/METADATA
 create mode 100644 venv/lib/python3.7/site-packages/pip-20.1.1.dist-info/RECORD
 create mode 100644 venv/lib/python3.7/site-packages/pip-20.1.1.dist-info/WHEEL
 create mode 100644 venv/lib/python3.7/site-packages/pip-20.1.1.dist-info/entry_points.txt
 create mode 100644 venv/lib/python3.7/site-packages/pip-20.1.1.dist-info/top_level.txt
 create mode 100644 venv/lib/python3.7/site-packages/pip/__init__.py
 create mode 100644 venv/lib/python3.7/site-packages/pip/__main__.py
 create mode 100644 venv/lib/python3.7/site-packages/pip/__pycache__/__init__.cpython-37.pyc
 create mode 100644 venv/lib/python3.7/site-packages/pip/__pycache__/__main__.cpython-37.pyc
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/__init__.py
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/__pycache__/__init__.cpython-37.pyc
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/__pycache__/build_env.cpython-37.pyc
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/__pycache__/cache.cpython-37.pyc
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/__pycache__/configuration.cpython-37.pyc
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/__pycache__/exceptions.cpython-37.pyc
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/__pycache__/locations.cpython-37.pyc
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/__pycache__/main.cpython-37.pyc
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/__pycache__/pyproject.cpython-37.pyc
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/__pycache__/self_outdated_check.cpython-37.pyc
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/__pycache__/wheel_builder.cpython-37.pyc
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/build_env.py
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/cache.py
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/cli/__init__.py
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/cli/__pycache__/__init__.cpython-37.pyc
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/cli/__pycache__/autocompletion.cpython-37.pyc
 create mode 100644 venv/lib/python3.7/site-packages/pip/_internal/cli/__pycache__/base_command.cpython-37.pyc
```

5.2 Sensitive File Detection

A list of sensitive directories is defined:

```
/Users/<username>/Important
/Users/<username>/Confidential
```

When activity occurs in these folders, alerts are raised and logged.

5.3 Integrity Verification

Files are hashed using SHA-256 before and after transfer:

```
import hashlib

def hash_file(path):
    with open(path, 'rb') as f:
        return hashlib.sha256(f.read()).hexdigest()
```

5.4 Alert & Logging System

Events are logged to:

```
logs/activity.log
```

Sample entry:

```
2025-XX-XX 10:15:32 - created |
/Users/aneesafathima/Important/confidential.txt | HASH=...
```

---

6. File Monitor Script

Below is the complete monitoring script used in this project:

```
import time
import logging
import hashlib
from watchdog.observers import Observer
from watchdog.events import FileSystemEventHandler
import os

# Configure logging
os.makedirs("logs", exist_ok=True)
logging.basicConfig(
    filename="logs/activity.log",
    level=logging.INFO,
    format='%(asctime)s - %(message)s'
)
```

```
# Paths to monitor
SENSITIVE_PATHS = [
    "/Users/aneesafathima/Important",
    "/Users/aneesafathima/Confidential"
]

def hash_file(path):
    try:
        with open(path, 'rb') as f:
            return hashlib.sha256(f.read()).hexdigest()
    except:
        return None

class MonitorHandler(FileSystemEventHandler):
    def on_any_event(self, event):
        path = event.src_path
        integrity = hash_file(path)
        logging.info(f"{event.event_type} | {path} |
HASH={integrity}")
        print(f"Detected: {event.event_type} → {path}")

        for spath in SENSITIVE_PATHS:
            if spath in path and event.event_type in
["created","modified"]:
                logging.warning(f"Sensitive File Movement Detected:
{path}")

if __name__ == "__main__":
    path_to_monitor = os.path.expanduser("~/")
    print(f"Monitoring {path_to_monitor}...")
    observer = Observer()
    observer.schedule(MonitorHandler(), path=path_to_monitor,
recursive=True)
    observer.start()
    try:
        while True:
            time.sleep(1)
    except KeyboardInterrupt:
        observer.stop()
    observer.join()
```

---

7. Sample Tests & Outputs

7.1 Test – File Creation

In a separate terminal:

```
mkdir ~/ImportantTest
echo "Test content" > ~/ImportantTest/file1.txt
```

7.2 Test – Move / Rename

```
mv ~/ImportantTest/file1.txt ~/ImportantTest/file2.txt
```

7.3 Test – Copy to External Folder

```
cp ~/ImportantTest/file2.txt ~/TestFolder/
```

---

8. Log Review

To analyze recent events:

```
tail -n 25 logs/activity.log
```

---

9. Alert Reporter Script

Use this script to output alert lines from the log:

```
with open("logs/activity.log","r") as f:
    for line in f:
        if "WARNING" in line:
            print("ALERT:", line.strip())
```

---

10. Final Audit Summary

The system successfully:

✔ Logged all file events
✔ Detected sensitive file operations
✔ Performed integrity checks
✔ Issued alerts for unauthorized movements
✔ Produced detailed logs

---

11. Flowchart & Architecture

```
START
↓
Monitor File System Events
↓
Is Event Sensitive?
 ↓ Yes → Hash + Authorization Check
 ↓ No  → Log Event
↓
If Unauthorized → Generate Alert
↓
Update Logs
↓
Repeat
↓
END
```

---

12. Learning Outcomes

After completing this project, you will understand:

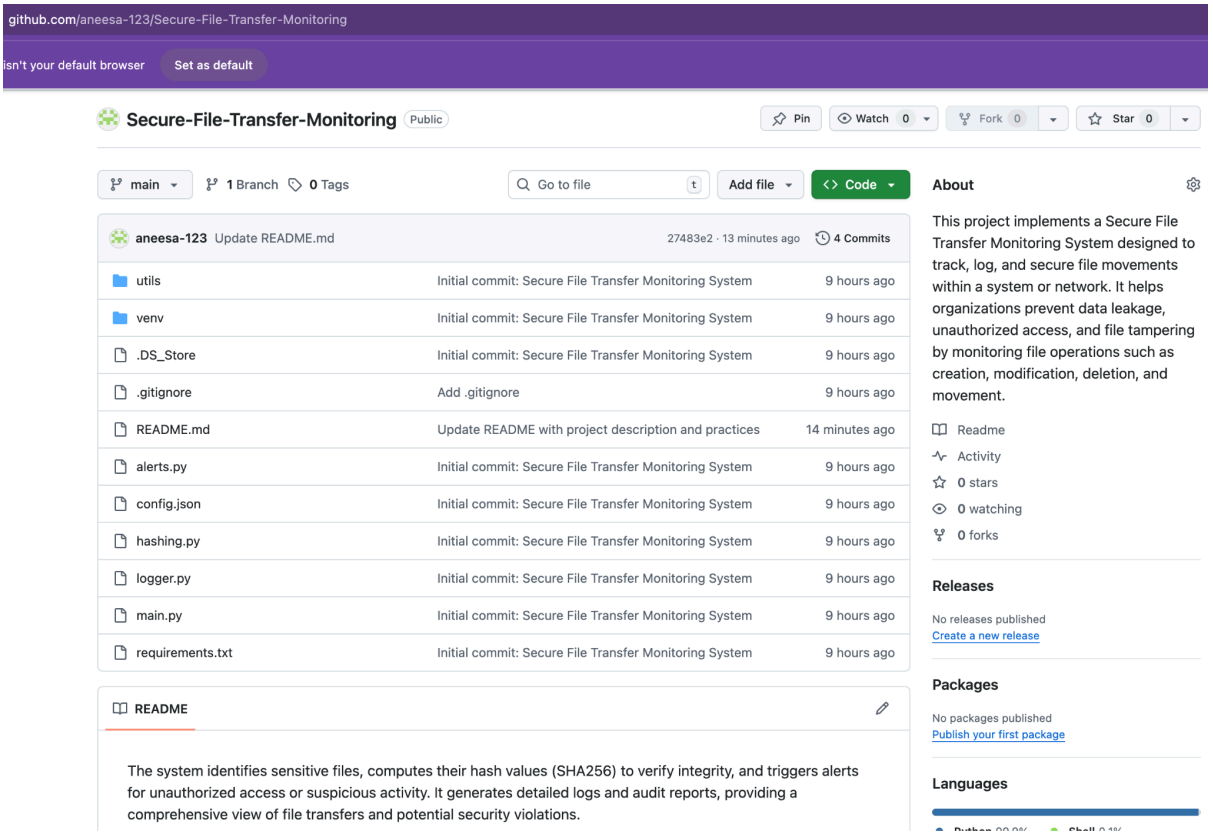
✔ File system event monitoring
✔ Real-time alerting mechanisms
✔ Hash-based integrity verification
✔ How to log and report suspicious activity
✔ How to structure security monitoring tools

---

13. Repository & Submission

Your GitHub repository should contain:

```
secure-file-monitor/
├── docs/                    ← Final report PDF
├── logs/                    ← Log files
├── screenshots/             ← Output screenshots
├── src/                     ← Python scripts
├── samples/                 ← Sample files used for testing
├── README.md
```



Repository link:
https://github.com/aneesa-123/Secure-File-Transfer-Monitoring

---

14. References

1. Python Watchdog Documentation
2. hashlib – Python Standard Library
3. GitHub Version Control
4. File System Monitoring Best Practices