Title(SELF USECASE)

System File-Sharing Activity Monitor – Sender & Receiver Identification

Problem Statement

In shared or cloud-based systems, users may transfer files to external systems using tools like SCP, SFTP or FTP.
If such file-sharing activities are not monitored, it may lead to data leakage or security incidents.

Hence, there is a need for a system-level monitoring solution to detect file-sharing events and capture sender and receiver details.

Objectives

The system should,

detect file-sharing or file-transfer events

Identify the sender (local user and process)

Identify the receiver details (IP address)

implementation (python code)

```
import subprocess
import pwd
import os
from datetime import datetime
TRANSFER_COMMANDS = ["scp", "sftp", "rsync", "ftp"]
def get_user(pid):
  try:
    uid = os.stat(f"/proc/{pid}").st_uid
    return pwd.getpwuid(uid).pw_name
  except:
    return "unknown"
output = subprocess.getoutput("ss -tnp").splitlines()
print("\n Checking for file-sharing activity...\n")
for line in output:
  for cmd in TRANSFER_COMMANDS:
    if cmd in line
```

```
try:
        parts = line.split()
        pid = int(parts[-1].split("pid=")[1].split(",")[0])
        receiver_ip = parts[4].split(":")[0]
        user = get_user(pid)
        print(" File Transfer Detected")
        print("Time       :", datetime.now())
        print("Sender User :", user)
        print("Process    :", cmd)
        print("Receiver IP :", receiver_ip)
        print("-" * 40)
      except:
        pass
```

FLOW

User initiates file transfer ->Linux OS processes & network activity -->Python monitoring script --> Sender and receiver details extracted --> Output displayed