CSHO331CSP: Ethical Hacking

Directory Monitoring Bash Script using inotify tools
Assignment 1

BY

Bettina Soni (2463012)

3BTAIML

Department of Computer Science and Engineering

School of Engineering and Technology
CHRIST (Deemed to be University)
Kumbalagodu, 560074
August, 2025

Inotify Tools:

inotify-tools is a Linux utility that provides a simple interface to the inotify feature of the Linux kernel. It allows users to watch for file system events like creation, deletion, modification, and movement of files and directories. With inotifywait, we can listen for changes in real-time, making it ideal for monitoring directories for any unauthorized or unexpected activity.

Bash Scripts:

Bash scripts are plain text files containing a series of commands that are executed by the Bash shell. They automate repetitive tasks, reduce human error, and allow for the creation of powerful tools on Unix-like systems. In this case, the Bash script uses inotifywait to continuously track file changes and logs them for further review.

Purpose:

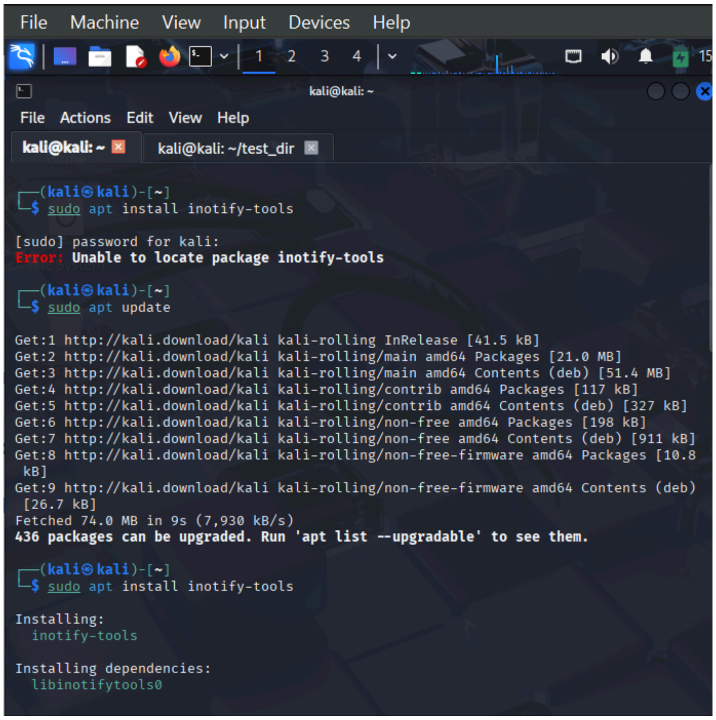
This script demonstrates how real-time monitoring can be achieved with minimal tools on Kali Linux. By watching a test directory, we successfully logged the creation and modification of files. This simple implementation highlights how security teams or developers can track critical file changes as part of auditing, debugging, or even detecting suspicious behavior.

Implementation:

→Inotify tools were installed as it was not already present in Kali Linux

command used: sudo apt update

```
sudo apt install inotify-tools
```


File Machine View Input Devices Help
1 2 3 4
kali@kali: ~
File Actions Edit View Help
kali@kali: ~
kali@kali: ~/test_dir

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install inotify-tools

[sudo] password for kali:
Error: Unable to locate package inotify-tools

┌──(kali㉿kali)-[~]
└─$ sudo apt update

Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [21.0 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [51.4 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [117 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [327 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [198 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [911 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [10.8
 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb)
 [26.7 kB]
Fetched 74.0 MB in 9s (7,930 kB/s)
436 packages can be upgraded. Run 'apt list --upgradable' to see them.

┌──(kali㉿kali)-[~]
└─$ sudo apt install inotify-tools

Installing:
  inotify-tools

Installing dependencies:
  libinotifytools0
```

→A new file named directory_monitor.sh was created to write the script in.

<u>command used: nano directory_monitor.sh</u>

→The script was added to the file:

```
#!/bin/bash

MONITOR_DIR="/home/kali/test_dir"
LOG_FILE="/home/kali/directory_changes.log"

echo "Monitoring directory: $MONITOR_DIR"
echo "Logging to: $LOG_FILE"

inotifywait -m -r -e create -e modify -e delete --format '%T %w %f %e' --timefmt
'%Y-%m-%d %H:%M:%S' "$MONITOR_DIR" >> "$LOG_FILE"
```

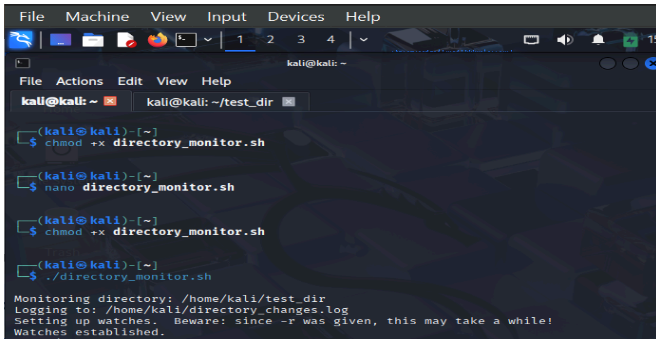
→The script was then made executable with a command line

command line: chmod +x directory_monitor.sh

→ The script was then run and also to start monitoring it.

Command line: ./directory_monitor.sh

→The following output was displayed



```
┌──(kali㉿kali)-[~]
└─$ chmod +x directory_monitor.sh

┌──(kali㉿kali)-[~]
└─$ nano directory_monitor.sh

┌──(kali㉿kali)-[~]
└─$ chmod +x directory_monitor.sh

┌──(kali㉿kali)-[~]
└─$ ./directory_monitor.sh

Monitoring directory: /home/kali/test_dir
Logging to: /home/kali/directory_changes.log
Setting up watches.  Beware: since -r was given, this may take a while!
Watches established.
```

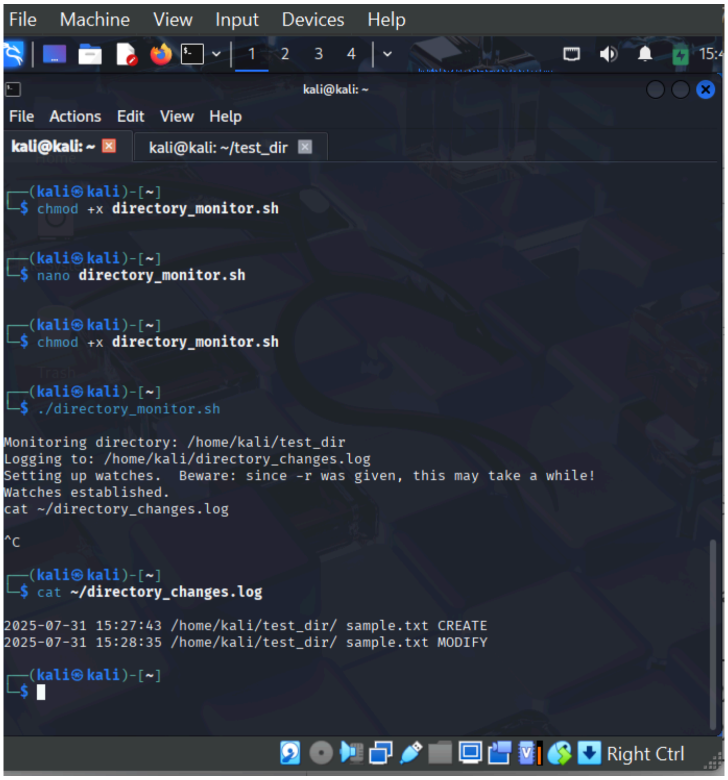
→**Created** a file called sample.txt inside /home/kali/test_dir and it was modified

→ A command to view the logs was given,

Command used: cat ~/directory_changes.log

The following was displayed on the terminal

```
2025-07-31 15:27:43 /home/kali/test_dir/ sample.txt CREATE
2025-07-31 15:28:35 /home/kali/test_dir/ sample.txt MODIFY
```



```
┌──(kali㉿kali)-[~]
└─$ chmod +x directory_monitor.sh

┌──(kali㉿kali)-[~]
└─$ nano directory_monitor.sh

┌──(kali㉿kali)-[~]
└─$ chmod +x directory_monitor.sh

┌──(kali㉿kali)-[~]
└─$ ./directory_monitor.sh

Monitoring directory: /home/kali/test_dir
Logging to: /home/kali/directory_changes.log
Setting up watches.  Beware: since -r was given, this may take a while!
Watches established.
cat ~/directory_changes.log

^C

┌──(kali㉿kali)-[~]
└─$ cat ~/directory_changes.log

2025-07-31 15:27:43 /home/kali/test_dir/ sample.txt CREATE
2025-07-31 15:28:35 /home/kali/test_dir/ sample.txt MODIFY

┌──(kali㉿kali)-[~]
└─$ 
```

Conclusion:

This assignment helped to understand how to use the inotify-tools package in Linux to monitor real-time changes in a directory. I gained practical experience in writing a Bash script to detect file creation, modification, and deletion events, and to trigger corresponding actions automatically. This enhanced my understanding of event-driven scripting, process automation, and file system monitoring in Linux. The activity also improved my ability to implement efficient system administration tasks and reinforced the importance of automation in maintaining secure and well-managed file systems.