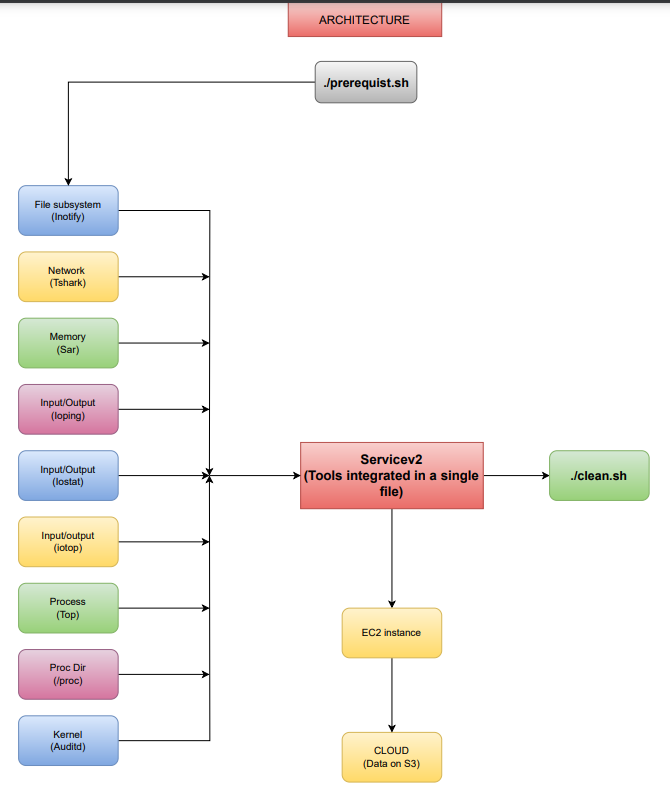
**OS TELEMETRY**

Introduction:

OS Telemetry involves the collection and analysis of data(logs) from various sources within an operating system to gain insights that can inform system optimization, troubleshooting, and future development. Operating System (OS) telemetry data plays a crucial role in identifying these anomalies, enabling timely responses and maintaining the integrity of the system. This report delves into the utilization of various telemetry tools to collect data from a Linux OS environment for the purpose of anomaly detection.

Architecture:



Methodology / Approach:

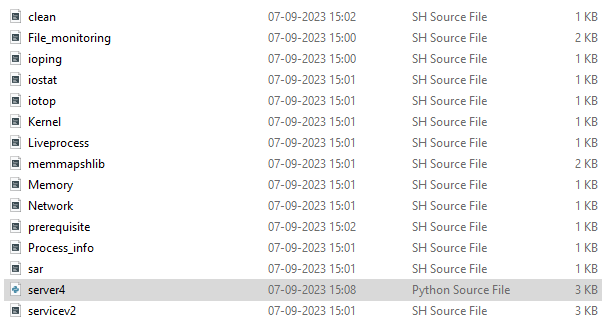
Language used: Shell scripting

Tools used:

1. Inotifywait: A tool used for monitoring file system events on the linux system for all directories. This tool captures the log data based on the events(for eg: create, delete, attrib, move) occurring on the monitored directories and logs them into the output directory.
2. Tshark: A command-line network protocol analyzer for unix-like systems which runs in the background and continuously monitors the network activity in real time.
3. sar: A system activity reporting tool for reports on various system activities like CPU usage, memory usage, and disk I/O activity. It gives information like available memory, used memory etc.
4. Ioping: ioping is a utility that measures disk I/O latency by sending small I/O requests to a storage device and calculating the time it takes to receive responses, providing insights into storage performance.
5. Iotop: Iotop is a command-line tool that monitors and displays real-time I/O usage of individual processes on a system, helping users identify which processes are consuming the most disk I/O resources.
6. Iostat: Iostat is a system monitoring tool that provides statistics about input/output (I/O) operations and CPU utilization for storage devices and partitions, aiding in the analysis of system performance and resource usage.
7. Auditd: Auditd is the userspace component of the Linux Audit Framework, which enables system administrators to track and monitor security-related events on a system, including file accesses, authentication events, kernel information(for eg.: creation/ deletion of user and user groups) and other system activities.
8. Top: Top is a command-line utility that provides a real-time view of system processes and resource utilization, offering information about CPU usage, memory usage, and other system metrics, allowing users to monitor and manage system performance.

Data collection methods:

We created separate bash scripts to run each tool and then store all data collected into text files(logs). We then integrated all the tools together in a single service bash file to run all the tools at the same time and store all the logs in the same output folder to make it easier to find all the data in one place to further analyze it.



When running the tool, we need to specify if you want to run only specific number of tools or run all the tools at once. If you want to run a specific set of tools you can enter the number of tools you want to run and then in the next line you specify which tools you want to run by entering the index number of the shown tools.

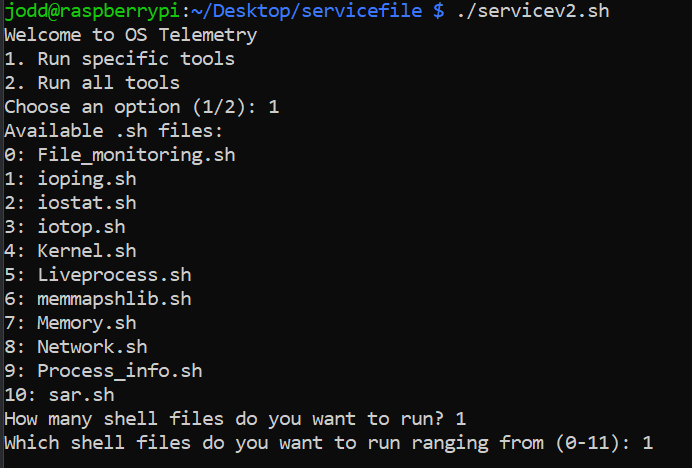
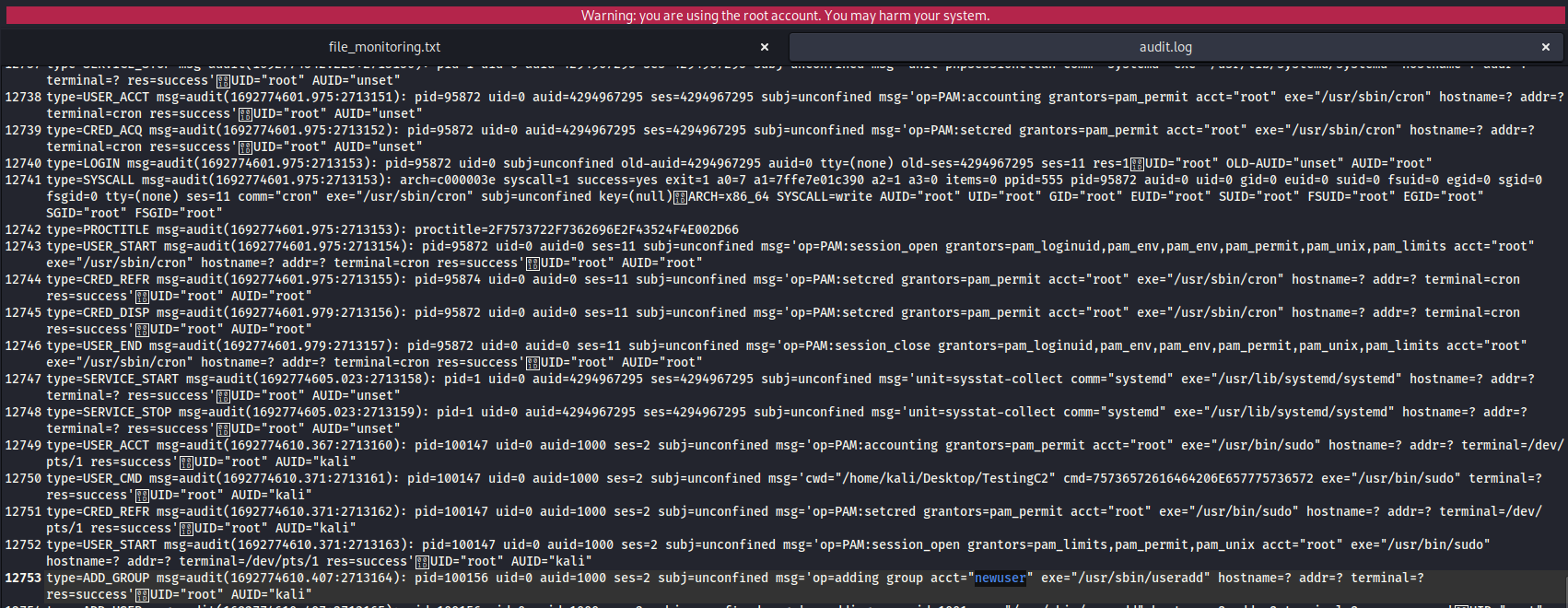


Figure 1 Execution of the service file.

Output of some of the subsystem:

1. Kernel subsystem: Tool used: Auditd

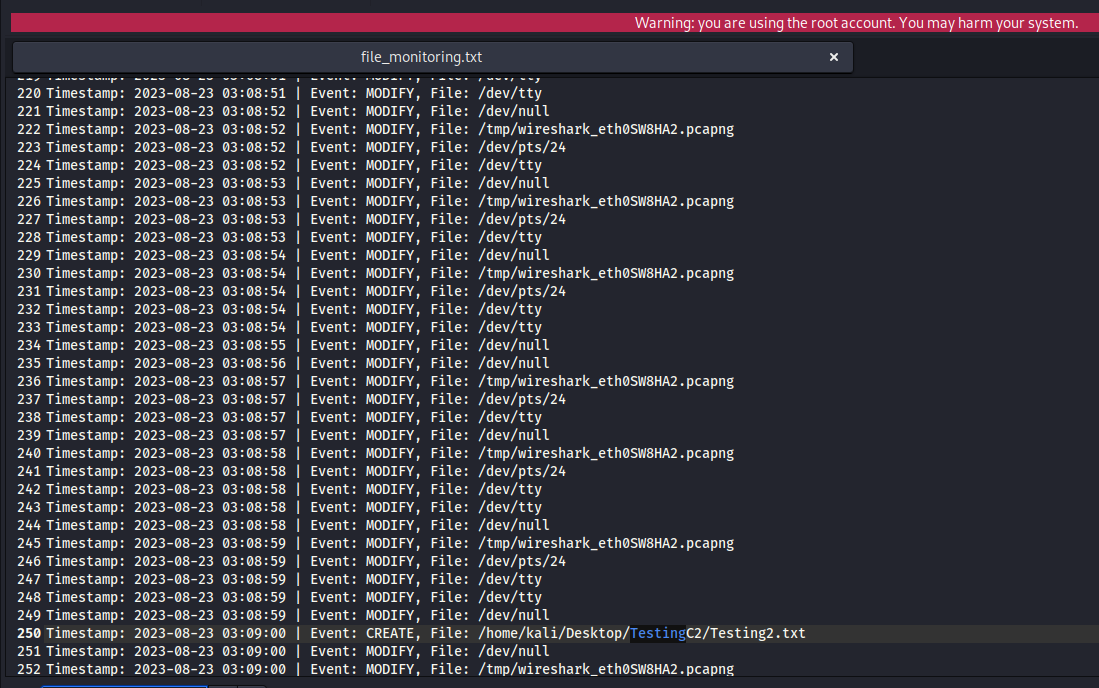
Here we can see the creation of new user is detected in this log.



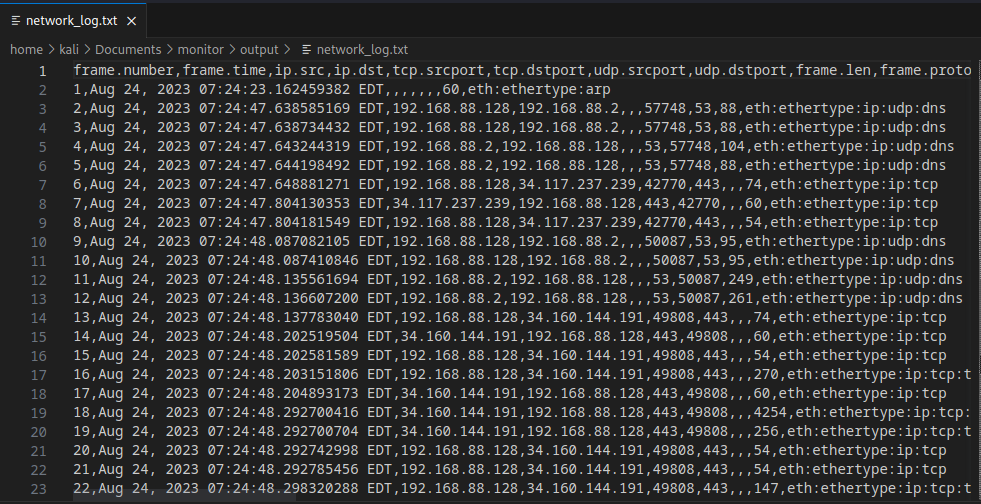
1. File Subsystem: Tool used: inotifywait

Here for example we have created a text file testing.txt and it is being logged.

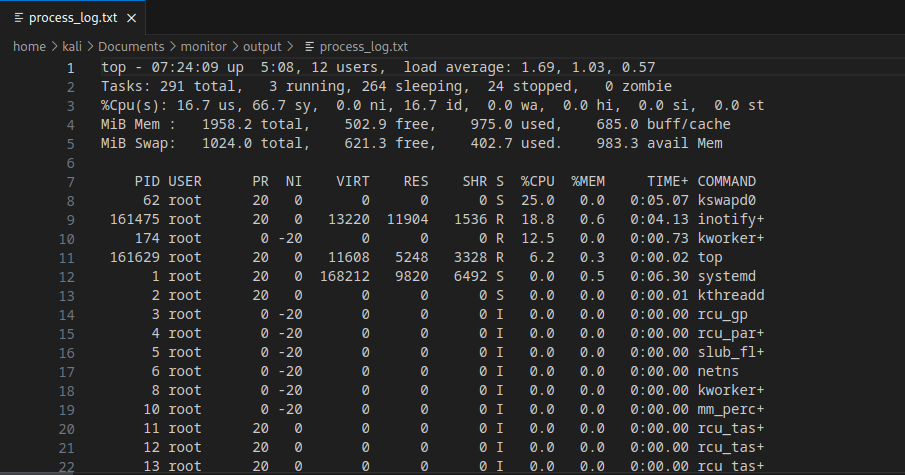
Also the modification of the temporary file created by wireshark is also being logged.



1. Network Subsystem: Tool used: Tshark



1. Process Subsytem:

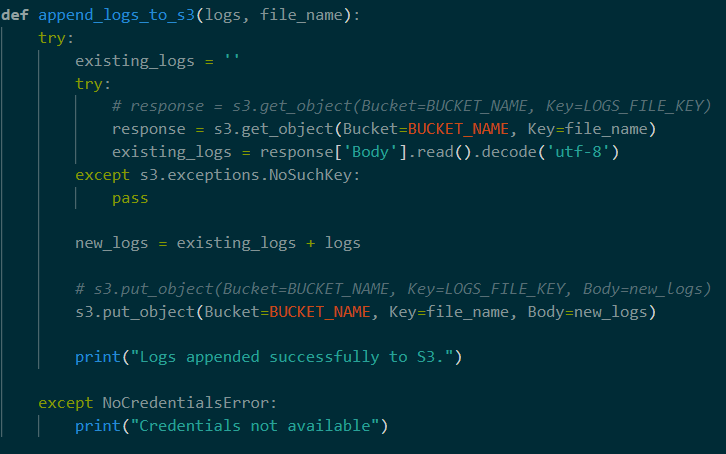


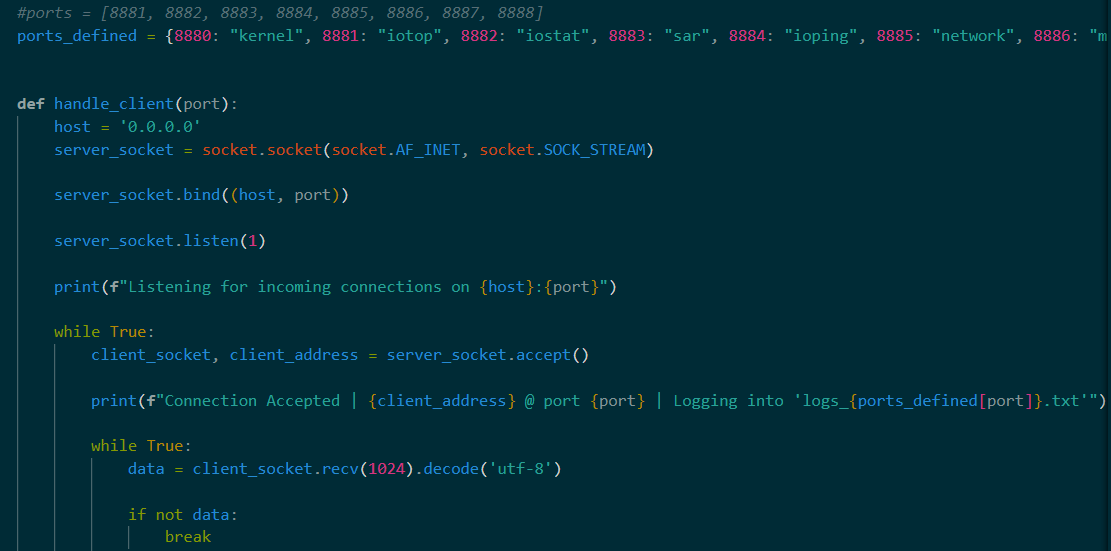
Uploading and storing data on AWS S3

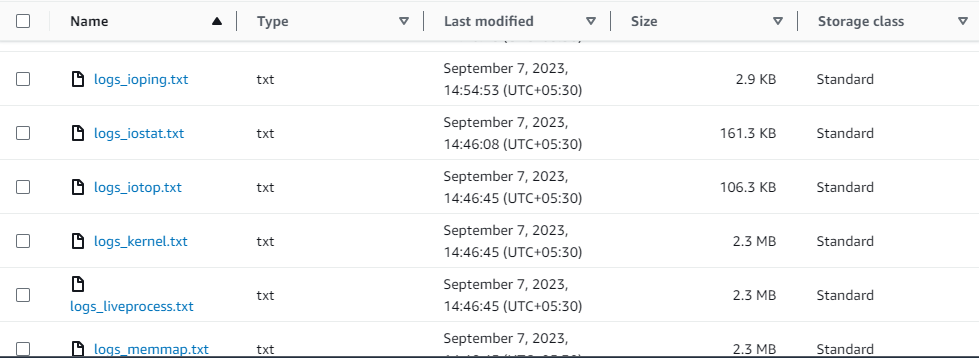
We store the data collected from the telemetry tools in the Amazon Web Services(AWS) Simple Storage Service(S3). This cloud storage service enables us to store and retrieve any amount of data from anywhere on the web.

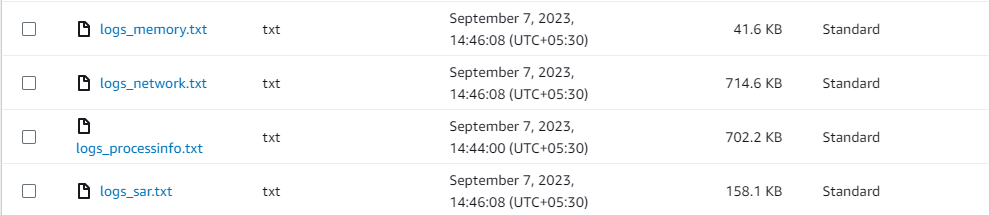
We created a server file in python which will be executed on an EC2 instance which will listen to the data sent through a port using netcat and this directly dumps the data to txt files in the S3 bucket in Realtime.

Code snippet for server file:









Limitations and challenges:

One challenge is the computation power required by the tools is very high and it can be sometimes time consuming to boot up the tool if sufficient computing power is not provided. We faced the challenge of maintain the system performance while telemetry tools collect system data in the background. One other limitation is that the tool is not platform independent and will only work for Linux based OS.

Conclusion:

Using OS telemetry tools to collect system data is an effective way to monitor system performance and can be further used for anomaly detection. This is the first phase of the project and we’re able to collect the data for some of the important subsystems.