# **🐧 Linux System Administration PoC Tasks**

🎯 Objective: Learn and demonstrate essential Linux skills through automation, monitoring, and scripting.

## **🔐 Task 1: File Structure & Permissions:**

* Create a user: studentuser
* Directory structure:  
  + /home/studentuser/projectX/logs
  + /home/studentuser/projectX/scripts
* Create welcome.txt with "Welcome to Linux"
* Set permissions: only studentuser can read/write
* Script: backup.sh to copy welcome.txt to logs with a timestamp

#!/bin/bash

cp /home/studentuser/projectX/welcome.txt /home/studentuser/projectX/logs/welcome\_$(date +%F\_%T).txt



## **🌐 Task 2: Networking Toolkit PoC**

Script: netinfo.sh  
 Includes:

* Show IP address, subnet, gateway
* List open ports (ss or netstat)
* Ping google.com and log response
* DNS lookup for openai.com
* Output saved to network\_report.txt

📝 Script (netinfo.sh):

bash

CopyEdit

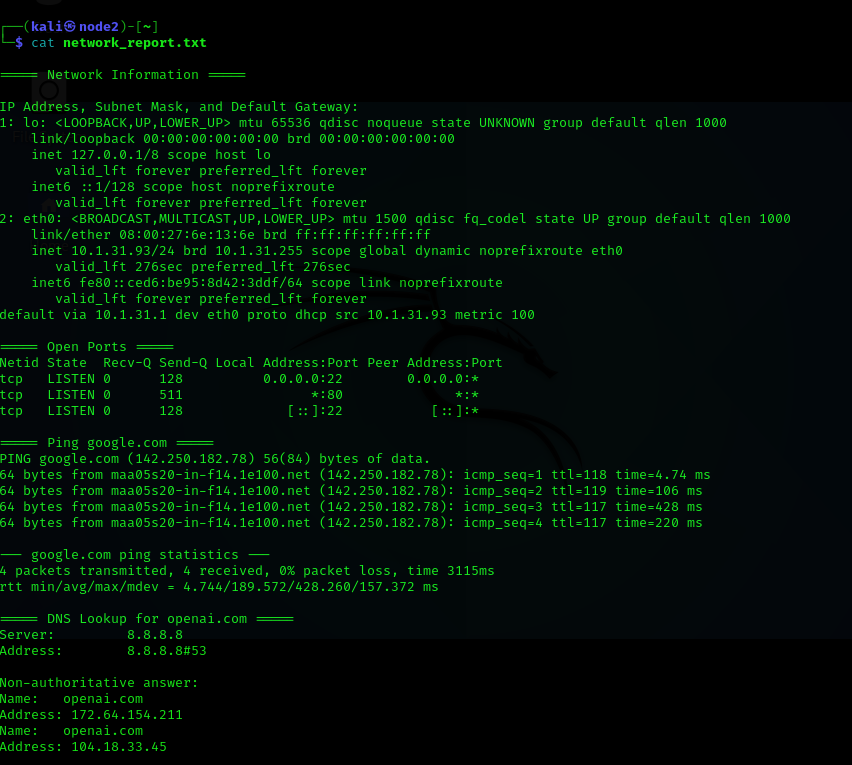
#!/bin/bash

{

echo "IP Address: $(hostname -I)"

echo "Gateway: $(ip route | grep default | awk '{print

} > network\_report.txt



## **🖥️ Task 3: Mini Server Monitor**

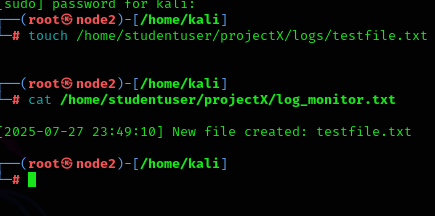
Script: monitor.sh

* Check if nginx is running (start if not)
* Show memory, CPU, disk usage
* Log activities with timestamps
* Schedule to run every 5 minutes via cron

⏱ Cron Entry:

bash

\*/5 \* \* \* \* /home/studentuser/projectX/scripts/[monitor.sh](http://monitor.sh)



## **👀 Task 4: File Watcher Script**

Script: watch\_dir.sh

* Monitor /home/studentuser/projectX/logs for new .txt files
* Log filename + timestamp to log\_monitor.txt

#!/bin/bash

WATCH\_DIR="/home/studentuser/projectX/logs"

LOG\_FILE="/home/studentuser/projectX/log\_monitor.txt"

touch "$LOG\_FILE"

echo "Watching directory: $WATCH\_DIR"

inotifywait -m -e create --format '%w%f' "$WATCH\_DIR" | while read NEW\_FILE

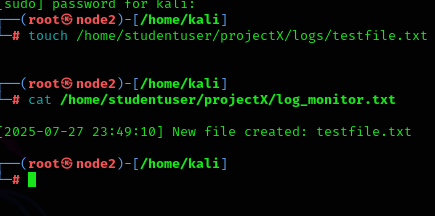
do

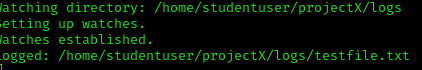
if [[ "$NEW\_FILE" == \*.txt ]]; then

TIMESTAMP=$(date '+%Y-%m-%d %H:%M:%S

echo "[$TIMESTAMP] New file created: $(basename "$NEW\_FILE")" >> "$LOG\_FILE"

echo "Logged: $NEW\_FILE"





## **🔐 Task 5: SSH Login Audit**

Script: ssh\_audit.sh

* Display last 5 successful logins
* Show last 5 failed login attempts
* Output saved to ssh\_audit.txt

📝 Script (ssh\_audit.sh):

bash

CopyEdit

#!/bin/bash

{

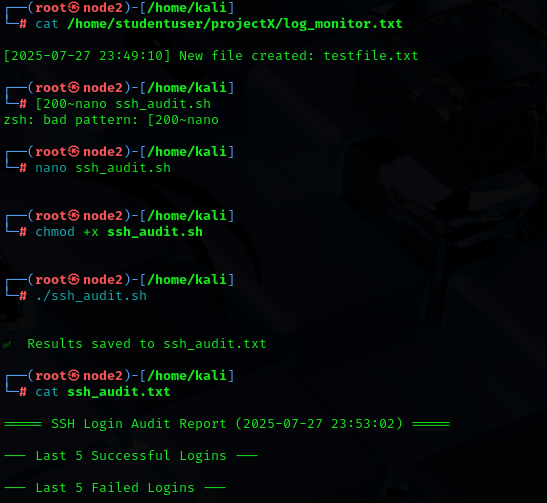
echo "Last 5 Successful Logins:"

last | head -n 5

echo -e "\nLast 5 Failed Attempts:"

grep "Failed password" /var/log/auth.log | tail -5

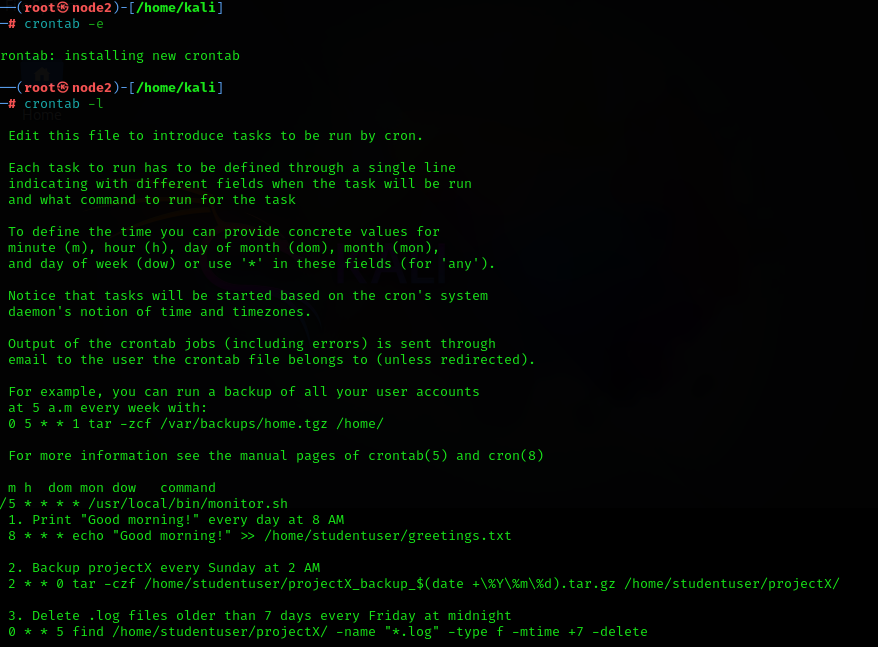
} > ssh\_audit.txt



## **⏰ Task 6: Crontab Practice**

Set up 3 scheduled jobs:

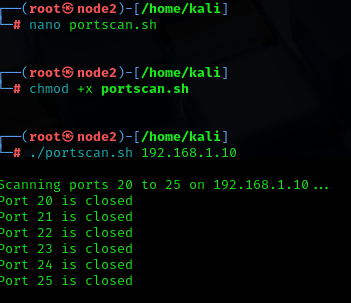
* ⏰ 8 AM daily → Print "Good morning!"
* 📦 Every Sunday → Backup /projectX
* 🧹 Every Friday @ midnight → Delete .log files older than 7 days



## **🔍 Task 7: Port Scanner**

Script: portscan.sh

* Accept user-supplied IP
* Scan ports 20–25 using nc or timeout
* Print status: OPEN or CLOSED



## **🌐 Task 8: Website Availability Checker**

Script: check\_sites.sh

* Input file: sites.txt
* Use curl or ping to check each site
* Log [UP] or [DOWN] to site\_status.log

📁 File (sites.txt):

https://google.com

https://openai.com

<https://github.com>

📝 Script (check\_sites.sh):

bash

Copy

Edit

#!/bin/bash

while read url; do

if curl -s --head "$url" | grep "200 OK" > /dev/null; then

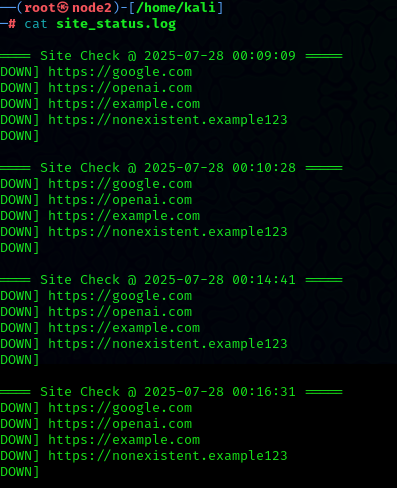
echo "$(date): $url [✅ UP]" >> site\_status.log

else

echo "$(date): $url [❌ DOWN]" >> site\_status.log

fi

done < sites.txt



## **📊 Task 9: Environment & Disk Report**

Script: sysreport.sh  
 Includes:

* Current user, hostname, uptime
* Filesystem info and usage
* PATH, SHELL, and environment variables
* Output to env\_disk\_report.txt

#!/bin/bash

echo " User: $(whoami)"

echo " Hostname: $(hostname)"

echo " Uptime: $(uptime -p)"

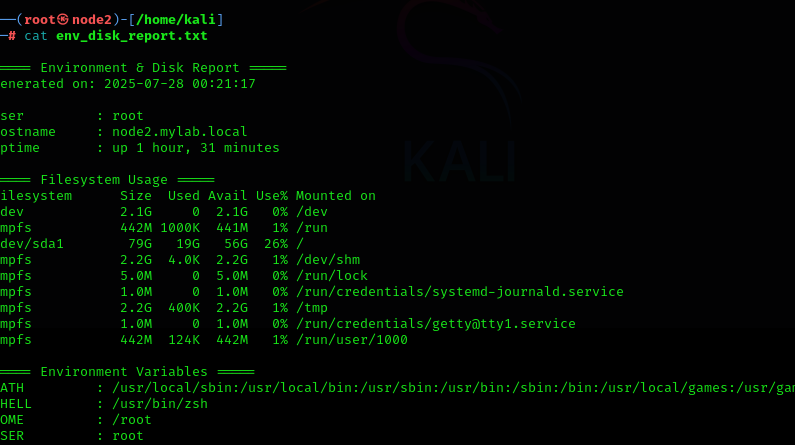
echo -e "\n Disk Usage:"

df -h

echo -e "\n Environment Variables:"

echo "PATH=$PATH"

echo "SHELL=$SHELL"



## **🗜️ Task 10: Compress & Archive Automation**

Script: a.sh

* Find .log files >10MB in logs/
* Compress into archive\_<date>.tar.gz
* Move archive to backup/ directory

#!/bin/bash

LOGS="/home/studentuser/projectX/logs"

BACKUP="/home/studentuser/projectX/backup"

DATE=$(date +%F)

find "$LOGS" -type f -name "\*.log" -size +10M | tar -czf "$BACKUP/archive\_${DATE}.tar.gz" -T -

