**Assignment 17 - Bash Script for Auto Ping and Log**

**Name:** Prajwal S Hangaragi  
**Student No:** 2462125

**1. Methodology**

* I created a Bash script that pings google.com every 5 minutes.
* The script extracts the response time or logs timeout if there is no response.
* It saves the results in a CSV file (ping\_log.csv) with the date, time, domain, and response time.
* The script keeps running in a loop until manually stopped.

**2. Script Used**

bash

Copy code

#!/bin/bash

DOMAIN="google.com"

LOGFILE="ping\_log.csv"

if [ ! -f "$LOGFILE" ]; then

echo "Timestamp,Domain,Response\_Time(ms)" > "$LOGFILE"

fi

while true; do

PING\_RESULT=$(ping -c 1 $DOMAIN | grep 'time=')

if [ $? -eq 0 ]; then

RESPONSE\_TIME=$(echo $PING\_RESULT | awk -F'time=' '{print $2}' | awk '{print $1}')

else

RESPONSE\_TIME="timeout"

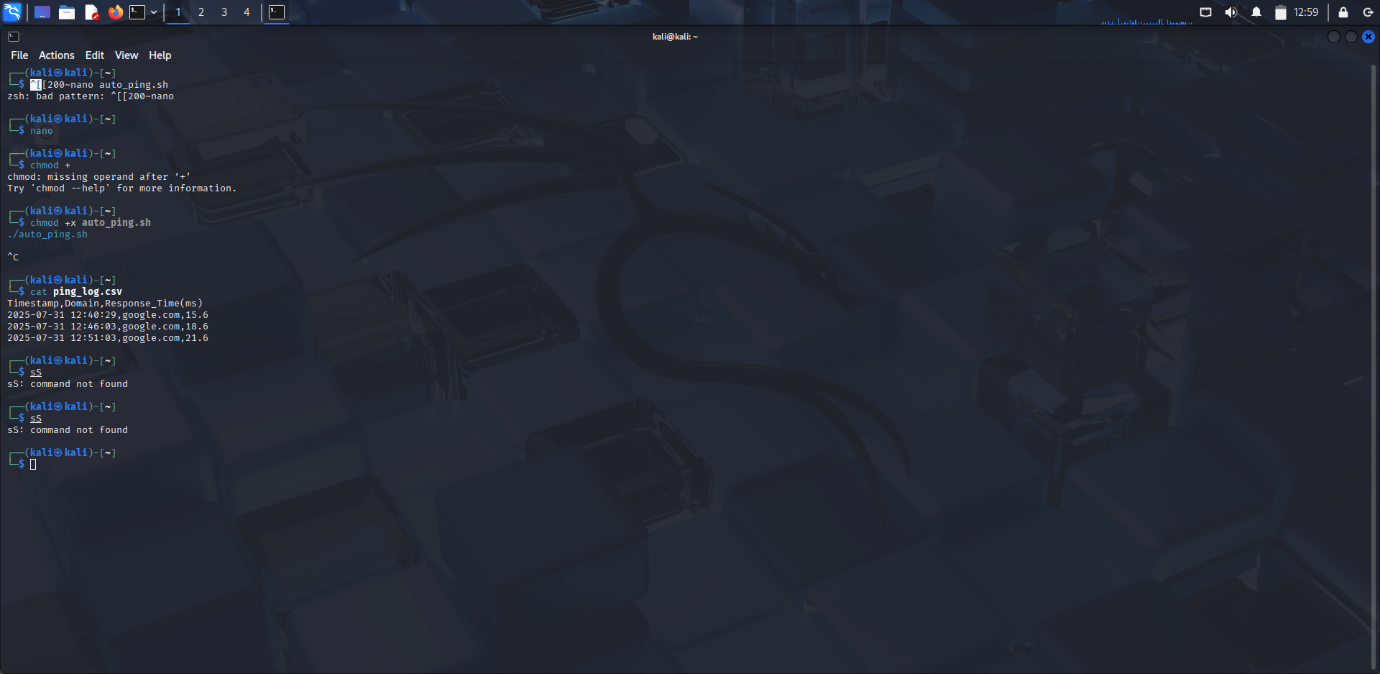
fi

echo "$(date '+%Y-%m-%d %H:%M:%S'),$DOMAIN,$RESPONSE\_TIME" >> "$LOGFILE"

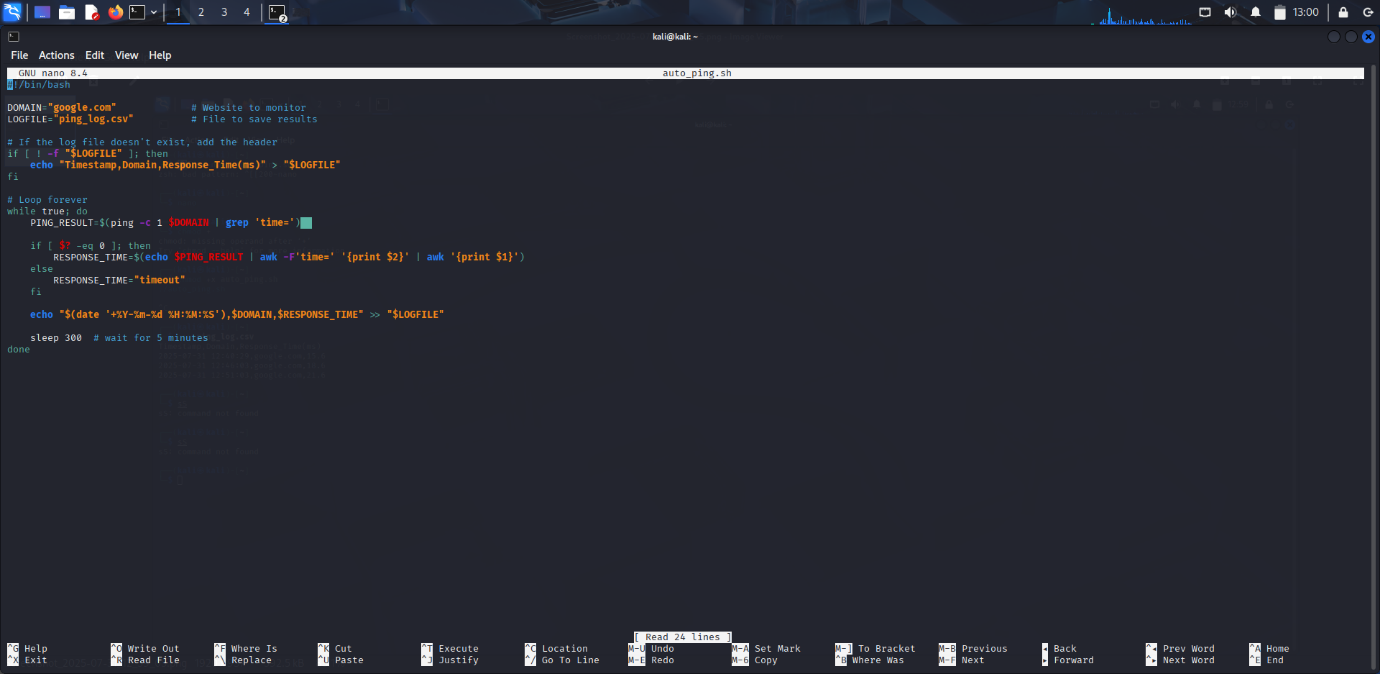
sleep 300

done

**3. Screenshot**

****

* Screenshot of ping\_log.csv output

****

* Screenshot of the script running in the terminal

**4. Findings (Sample Output)**

yaml

Timestamp, Domain, Response\_Time(ms)

2025-07-31 12:40:29,google.com,15.6

2025-07-31 12:46:03,google.com,18.6

2025-07-31 12:51:03,google.com,21.6

**5. Conclusion**

This script allows continuous monitoring of a website’s availability and response time.

* If the response time increases or timeout appears, it can indicate a network issue or outage.
* Such monitoring is useful for detecting DDoS attacks, slowdowns, or downtime early.