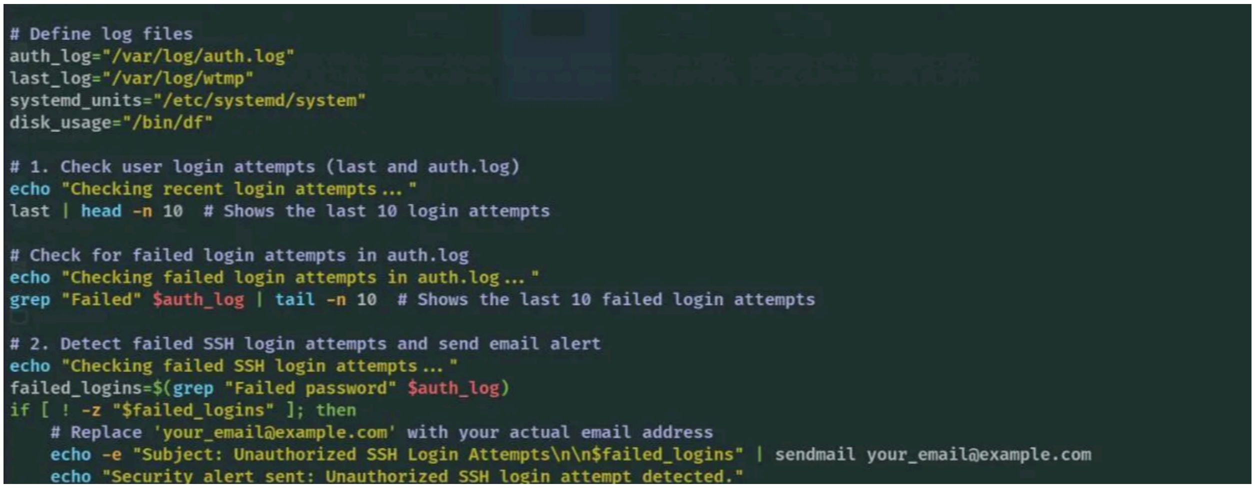
TASK 5

TASK 5 :

Automated Security Auditing & Scripting

Bash Script Creation

This guide outlines the process of developing and running a Bash script to configure system settings, identify vulnerabilities, and apply security measures. It also includes automating the script with scheduled tasks for continuous monitoring and integrating email alerts for real-time notifications.



```
# Define log files
auth_log="/var/log/auth.log"
last_log="/var/log/wtmp"
systemd_units="/etc/systemd/system"
disk_usage="/bin/df"

# 1. Check user login attempts (last and auth.log)
echo "Checking recent login attempts ... "
last | head -n 10  # Shows the last 10 login attempts

# Check for failed login attempts in auth.log
echo "Checking failed login attempts in auth.log ... "
grep "Failed" $auth_log | tail -n 10  # Shows the last 10 failed login attempts

# 2. Detect failed SSH login attempts and send email alert
echo "Checking failed SSH login attempts ... "
failed_logins=$(grep "Failed password" $auth_log)
if [ ! -z "$failed_logins" ]; then
    # Replace 'your_email@example.com' with your actual email address
    echo -e "Subject: Unauthorized SSH Login Attempts\n\n$failed_logins" | sendmail your_email@example.com
    echo "Security alert sent: Unauthorized SSH login attempt detected."
```

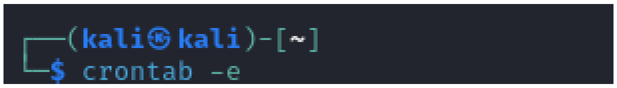
Explanation of the Script:

1. **Login Attempts:** Check recent logins and failed attempts from authentication logs.
2. **Running Services:** List active services to monitor system processes.
3. **Disk Usage:** Display storage consumption in a readable format.
4. **Inactive Users:** Identify accounts that have never logged in.

5. **Weak Passwords:** Scan the **/etc/shadow** file for common passwords (use tools like **John** or **Cracklib** for better detection).

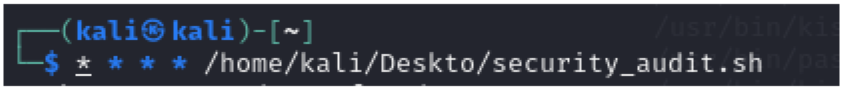
Mitigation – Automating Monitoring with Cron

Access the crontab configuration to schedule regular security checks.

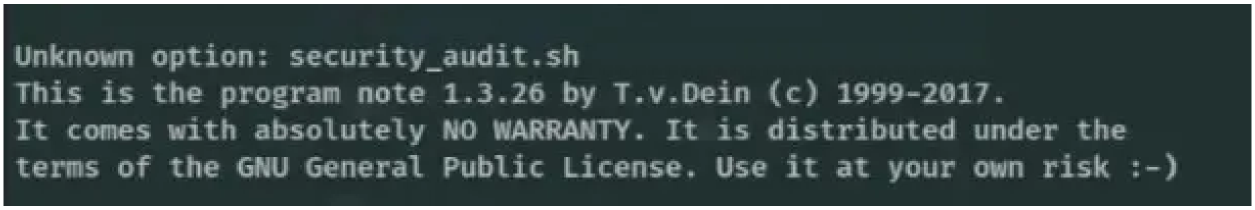


```
┌──(kali㉿kali)-[~]
└─$ crontab -e
```

To automate proactive monitoring with cron, add the following line to your **cron** jobs**: 0 * * * * /path/to/system_monitoring.sh**This configuration schedules the script to run hourly, ensuring consistent system monitoring**.**



```
┌──(kali㉿kali)-[~]
└─$ * * * * /home/kali/Deskto/security_audit.sh
```

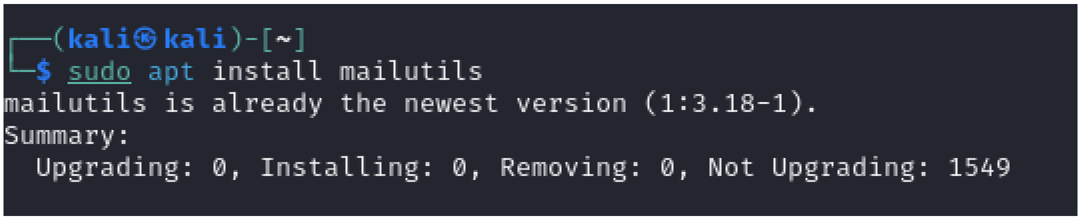


```
Unknown option: security_audit.sh
This is the program note 1.3.26 by T.v.Dein (c) 1999-2017.
It comes with absolutely NO WARRANTY. It is distributed under the
terms of the GNU General Public License. Use it at your own risk :-)
```

Implementing Security Alerts (Email Notification):

1.Ensure the mail service is installed to enable email alerts for unauthorized SSH attempts. Implementing this enhances system security by providing real-time notifications of suspicious login activities.

security posture by providing timely notifications and valuable insights into potential attack vectors.



```
┌──(kali㉿kali)-[~]
└─$ sudo apt install mailutils
mailutils is already the newest version (1:3.18-1).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1549
```

2.Update the script to send an email on detecting failed login attempts: **# Detect failed SSH login attempts and send email alert**

```
echo "Checking failed SSH login attempts..."

failed_logins=$(grep "Failed password" $auth_log)

if [ ! -z "$failed_logins" ]; then

echo -e "Subject: Unauthorized SSH Login Attempts\n\n$failed_logins" | sendmail your_email@example.com

echo "Security alert sent: Unauthorized SSH login attempt detected." fi
```