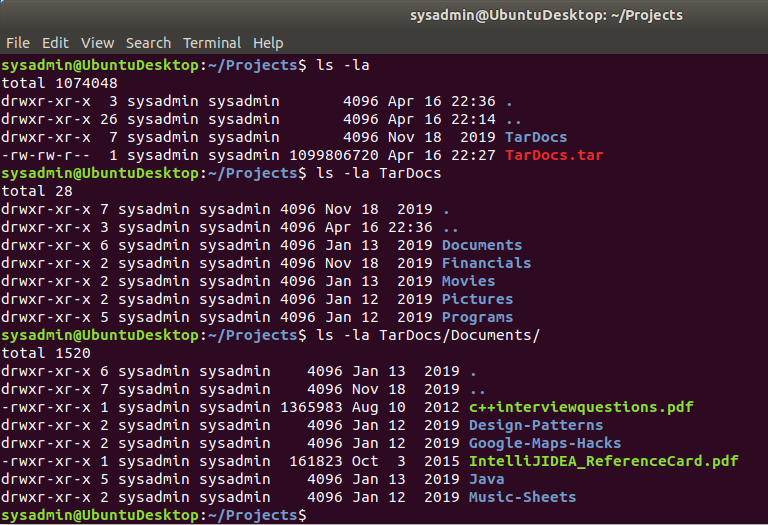
## **Week 5 Homework Submission File: Archiving and Logging Data**

### **Step 1: Create, Extract, Compress, and Manage tar Backup Archives**

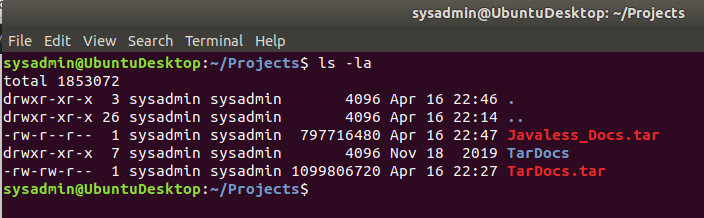
1. Command to **extract** the TarDocs.tar archive to the current directory:

tar xvvf TarDocs.tar



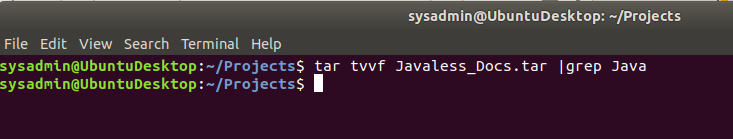
1. Command to **create** the Javaless\_Doc.tar archive from the TarDocs/ directory, while excluding the TarDocs/Documents/Java directory:

tar --exclude=TarDocs/Documents/Java -cvvf Javaless\_Docs.tar TarDocs



1. Command to ensure Java/ is not in the new Javaless\_Docs.tar archive:

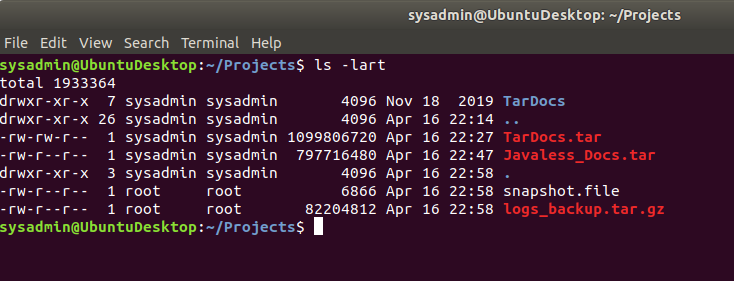
tar tvvf Javaless\_Docs.tar |grep Java



**Bonus**

* Command to create an incremental archive called logs\_backup\_tar.gz with only changed files to snapshot.file for the /var/log directory:

sudo tar czvvf logs\_backup.tar.gz --listed-incremental=snapshot.file /var/log



#### **Critical Analysis Question**

* Why wouldn't you use the options -x and -c at the same time with tar?

Option -c is for creation of a new full or incremental archive. Option -x is to extract all or selective files from an existing archive. So they are mutually exclusive and can’t be used together as part of the tar command.

### **Step 2: Create, Manage, and Automate Cron Jobs**

1. Cron job for backing up the /var/log/auth.log file:

We can add the following command in sysadmin’s crontab by running crontab -e command

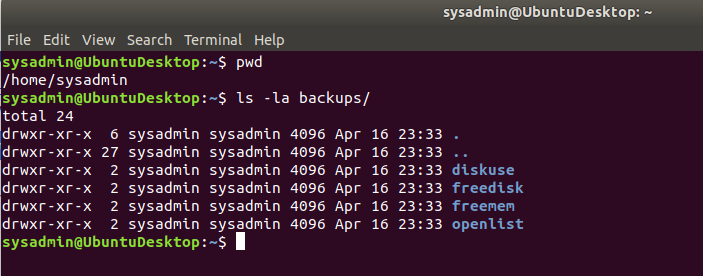
0 6 \* \* 3 sudo tar czvvf /auth\_backup.tgz /var/log/auth.log

Or

We can add the following command in root’s crontab by running sudo crontab -e command

0 6 \* \* 3 tar czvvf /auth\_backup.tgz /var/log/auth.log

### **Step 3: Write Basic Bash Scripts**

1. Brace expansion command to create the four subdirectories:  
   mkdir -p ~/backups/{freemem,diskuse,openlist,freedisk}
2. Paste your system.sh script edits below:

#!/bin/bash

# INSTRUCTIONS: Edit the following placeholder command and output file paths

# For example: cpu\_usage\_tool > ~/backups/cpuuse/cpu\_usage.txt

# The cpu\_usage\_tool is the command and ~/backups/cpuuse/cpu\_usage.txt is the file path

# In the above example, the `cpu\_usage\_tool` command will output CPU usage information into a `cpu\_usage.txt` file.

# Do not forget to use the -h option for free memory, disk usage, and free disk space

# Free memory output to a free\_mem.txt file

/usr/bin/free -h > ~/backups/freemem/free\_mem.txt

# Disk usage output to a disk\_usage.txt file

/usr/bin/du -h > ~/backups/diskuse/disk\_usage.txt

# List open files to a open\_list.txt file

/usr/bin/lsof > ~/backups/openlist/open\_list.txt

# Free disk space to a free\_disk.txt file

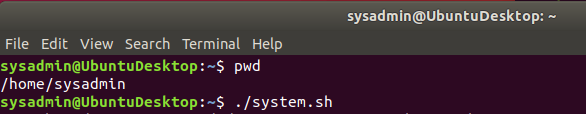
/bin/df -h > ~/backups/freedisk/free\_disk.txt

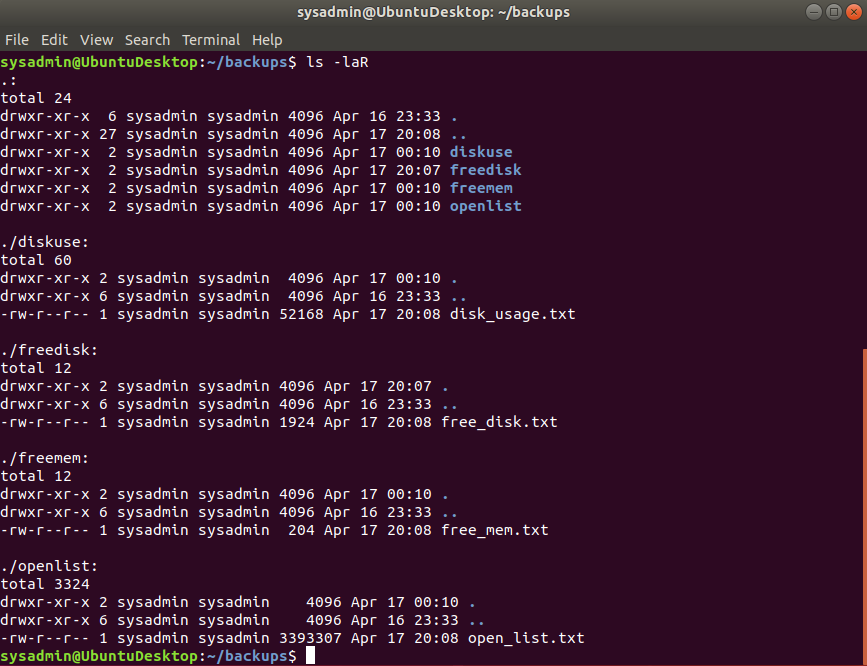
1. Command to make the system.sh script executable:  
   chmod +x system.sh

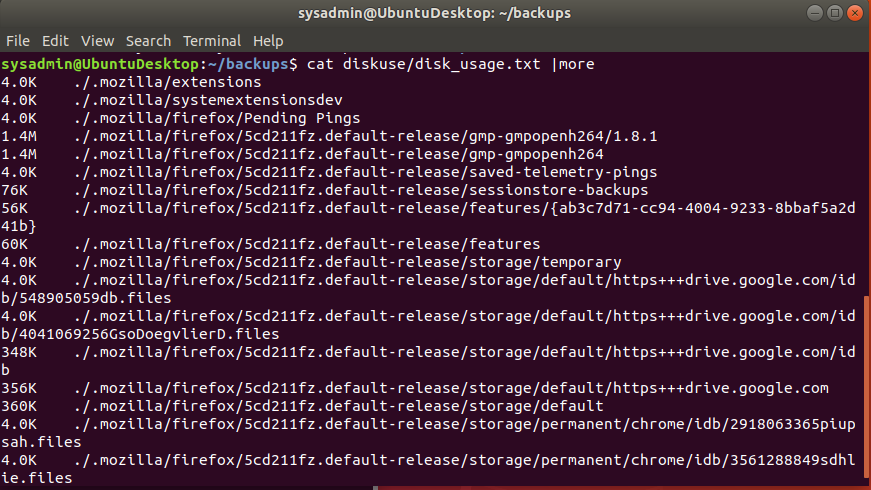
**Optional**

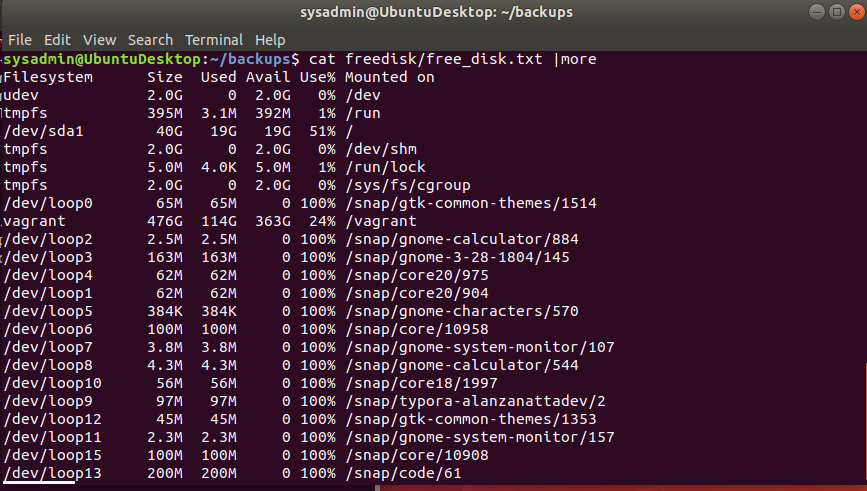
* Commands to test the script and confirm its execution:

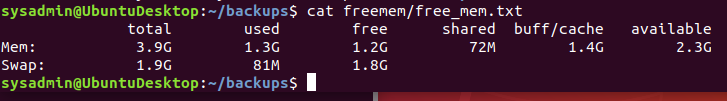
./system.sh

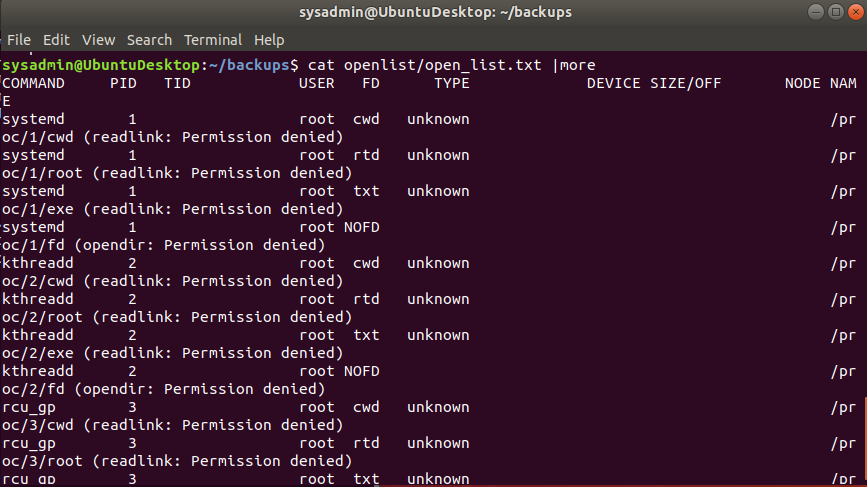




****

****

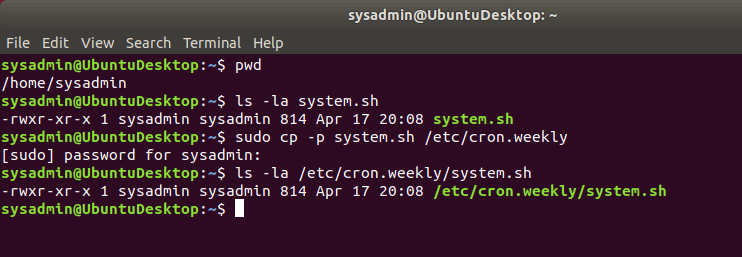
****

****

**Bonus**

* Command to copy system to system-wide cron directory:

sudo cp -p system.sh /etc/cron.weekly



### **Step 4. Manage Log File Sizes**

1. Run sudo nano /etc/logrotate.conf to edit the logrotate configuration file.  
     
    Configure a log rotation scheme that backs up authentication messages to the /var/log/auth.log.  
   * Add your config file edits below:

/var/log/auth.log {

weekly

rotate 7

notifempty

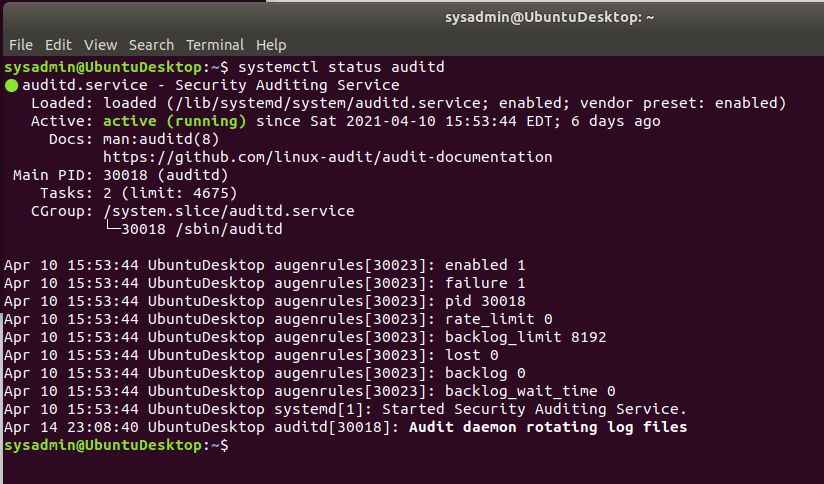
delaycompress

missingok

}

### **Bonus: Check for Policy and File Violations**

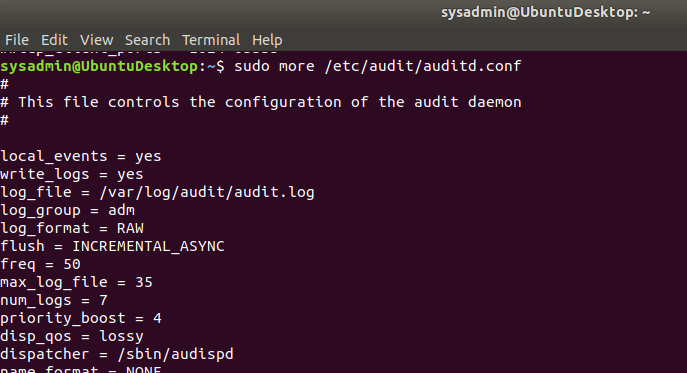
1. Command to verify auditd is active:

systemctl status auditd  


1. Command to set number of retained logs and maximum log file size:  
   * Add the edits made to the configuration file below:

max\_log\_file = 35

num\_logs = 7



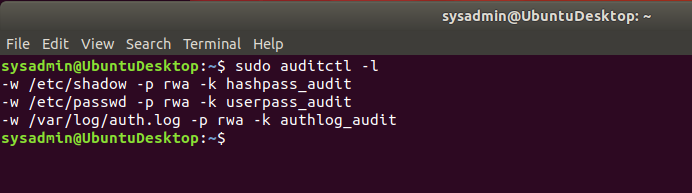
1. Command using auditd to set rules for /etc/shadow, /etc/passwd and /var/log/auth.log:  
   * Add the edits made to the rules file below:

-w /etc/shadow -p wra -k hashpass\_audit

-w /etc/passwd -p wra -k userpass\_audit

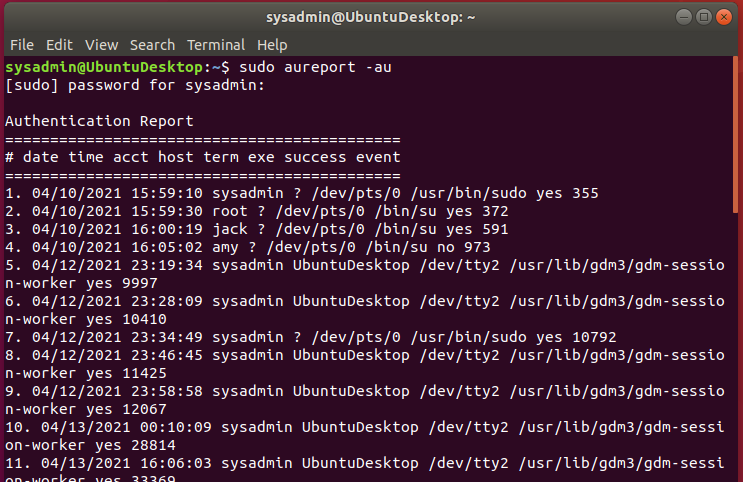
-w /var/log/auth.log -p wra -k authlog\_audit

1. Command to restart auditd:  
   sudo systemctl restart auditd
2. Command to list all auditd rules:  
   sudo auditctl -l



1. Command to produce an audit report:

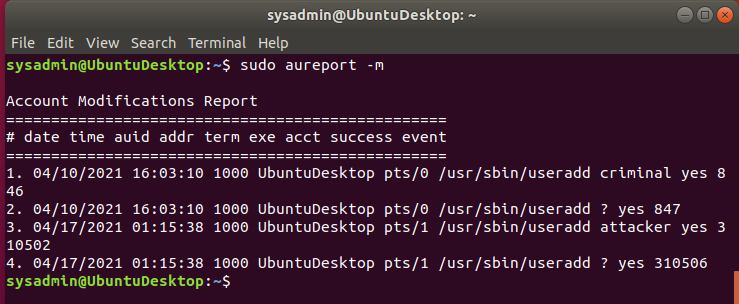
sudo aureport -au



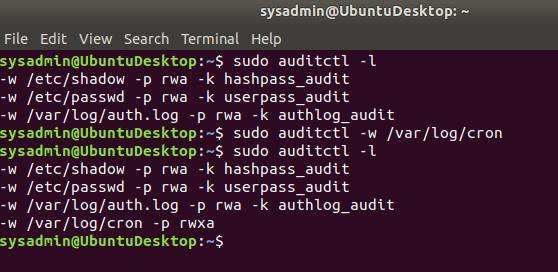
1. Create a user with sudo useradd attacker and produce an audit report that lists account modifications:

sudo useradd attacker

sudo aureport -m



1. Command to use auditd to watch /var/log/cron:  
   sudo auditctl -w /var/log/cron
2. Command to verify auditd rules:

sudo auditctl -l  


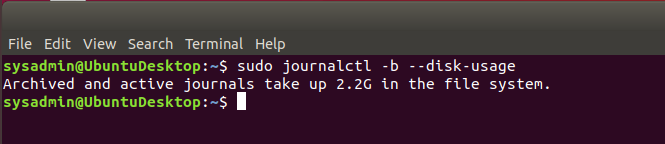
### **Bonus (Research Activity): Perform Various Log Filtering Techniques**

1. Command to return journalctl messages with priorities from emergency to error:

sudo journalctl -b -p err

1. Command to check the disk usage of the system journal unit since the most recent boot:

sudo journalctl -b --disk-usage



1. Command to remove all archived journal files except the most recent two:

sudo journalctl --vacuum-files=2

1. Command to filter all log messages with priority levels between zero and two, and save output to /home/sysadmin/Priority\_High.txt:

sudo sh -c "journalctl -p 0..2 > /home/student/Priority\_High.txt”

1. Command to automate the last command in a daily cronjob. Add the edits made to the crontab file below:  
     
   Though we are able to run the command in the step above in interactive mode, I couldn’t automate this using sysadmin’s cron job because sudo requires an option to enter password. So, we are adding a cron job in root’s crontab.

@daily journalctl -p 0..2 > /home/student/Priority\_High.txt