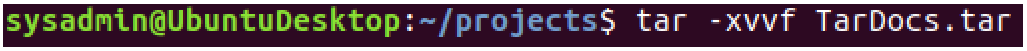
Week 5 Homework Submission File: Archiving and Logging Data

Please edit this file by adding the solution commands on the line below the prompt.

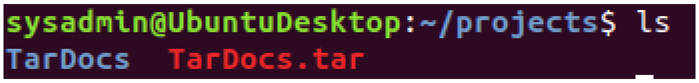
Save and submit the completed file for your homework submission.

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

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

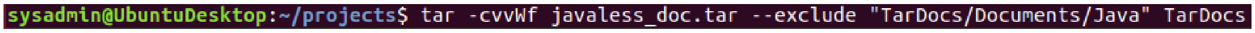


```
sysadmin@UbuntuDesktop:~/projects$ tar -xvvf TarDocs.tar
```



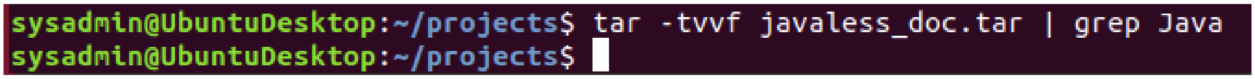
```
sysadmin@UbuntuDesktop:~/projects$ ls
TarDocs  TarDocs.tar
```

2. Command to **create** the Javaless_Doc.tar archive from the TarDocs/ directory, while excluding the TarDocs/Documents/Java directory:



```
sysadmin@UbuntuDesktop:~/projects$ tar -cvvWf javaless_doc.tar --exclude "TarDocs/Documents/Java" TarDocs
```

3. Command to ensure Java/ is not in the new Javaless_Docs.tar archive:



```
sysadmin@UbuntuDesktop:~/projects$ tar -tvvf javaless_doc.tar | grep Java
sysadmin@UbuntuDesktop:~/projects$
```

Bonus

- Command to create an incremental archive called logs_backup_tar.gz with only changed files to snapshot.file for the /var/log directory:

Not performed

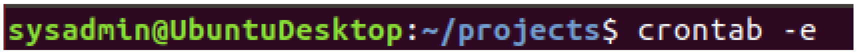
Critical Analysis Question

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

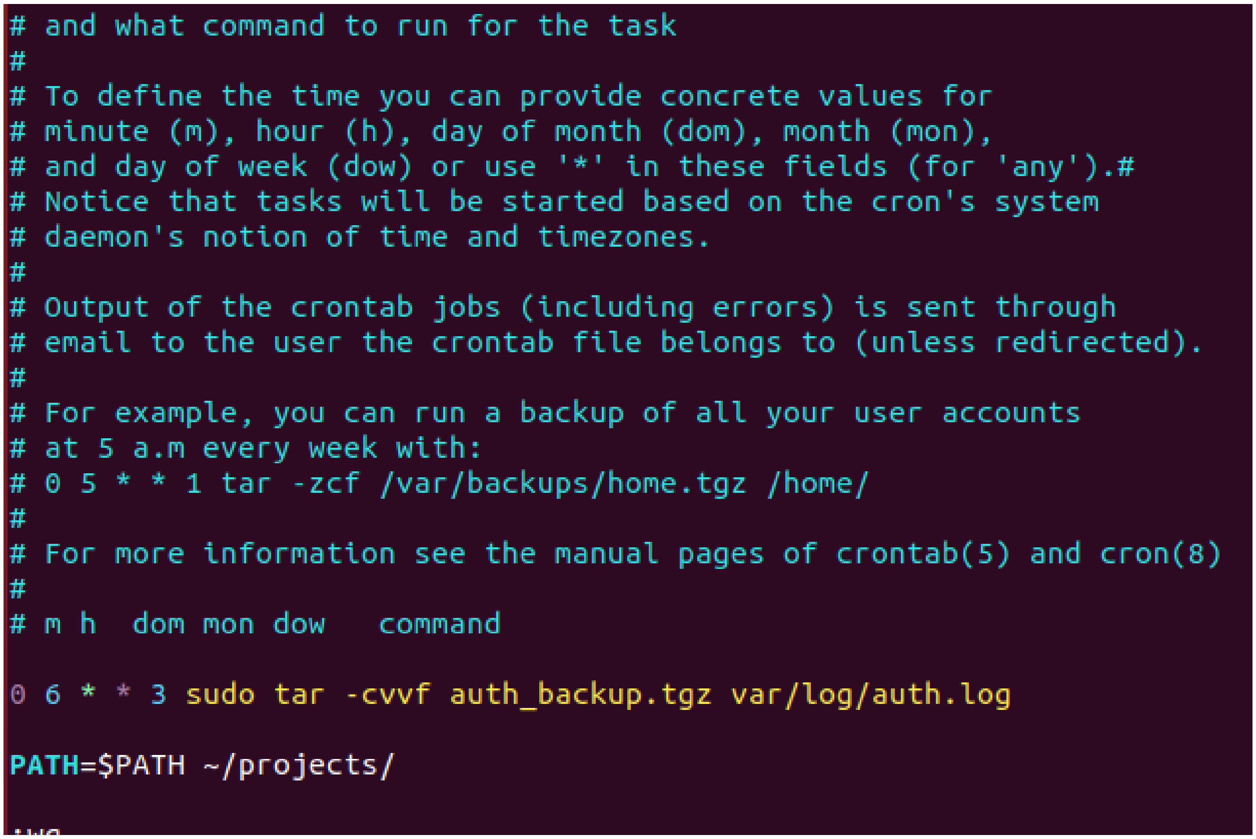
The option “-x” is to extract and “-c” is to create. They cannot be used together because a tar file cannot be extracted and then created or vice versa at the same time. Therefore they must be performed sequentially as for them to function simultaneously cannot be done.

Step 2: Create, Manage, and Automate Cron Jobs

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

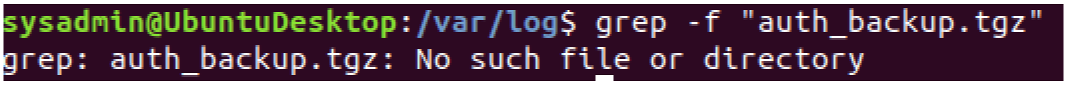


```
sysadmin@UbuntuDesktop:~/projects$ crontab -e
```



```
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

0 6 * * 3 sudo tar -cvvf auth_backup.tgz var/log/auth.log

PATH=$PATH ~/projects/

:wq
```

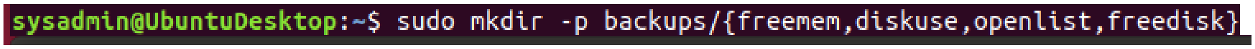
So this process will only happen every Wednesday at 6am, and only if the system is active. If we check the directory right now, there will be nothing.



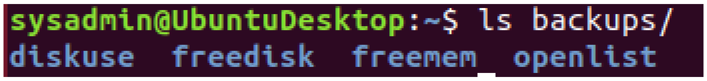
```
sysadmin@UbuntuDesktop:/var/log$ grep -f "auth_backup.tgz"
grep: auth_backup.tgz: No such file or directory
```

Step 3: Write Basic Bash Scripts

1. Brace expansion command to create the four subdirectories:
   No directory is specified other than a general “backup” directory, so we will make one in our home directory including all four of the required directories.



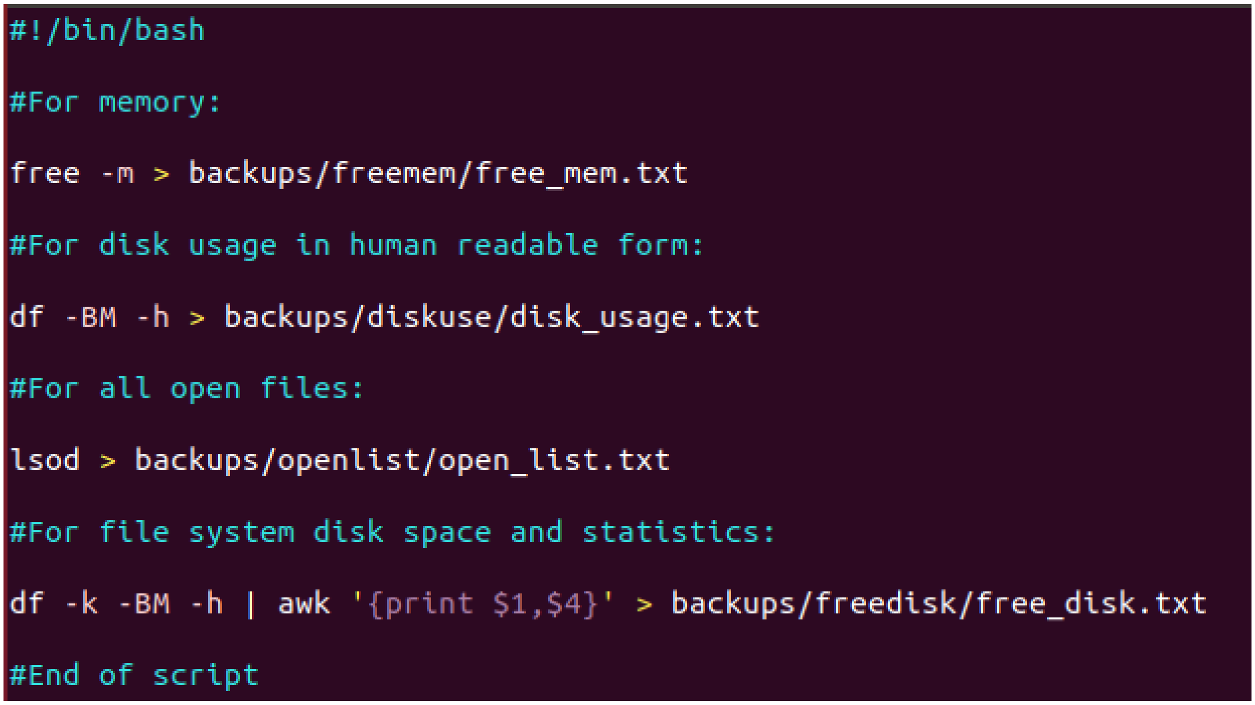
```
sysadmin@UbuntuDesktop:~$ sudo mkdir -p backups/{freemem,diskuse,openlist,freedisk}
```



```
sysadmin@UbuntuDesktop:~$ ls backups/
diskuse  freedisk  freemem  openlist
```

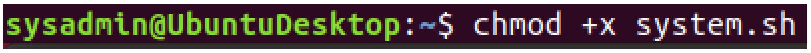
Paste your system.sh script edits below:

```
#!/bin/bash

#For memory:

free -m > backups/freemem/free_mem.txt

#For disk usage in human readable form:

df -BM -h > backups/diskuse/disk_usage.txt

#For all open files:

lsod > backups/openlist/open_list.txt

#For file system disk space and statistics:

df -k -BM -h | awk '{print $1,$4}' > backups/freedisk/free_disk.txt

#End of script

```



```
#!/bin/bash

#For memory:

free -m > backups/freemem/free_mem.txt

#For disk usage in human readable form:

df -BM -h > backups/diskuse/disk_usage.txt

#For all open files:

lsod > backups/openlist/open_list.txt

#For file system disk space and statistics:

df -k -BM -h | awk '{print $1,$4}' > backups/freedisk/free_disk.txt

#End of script
```

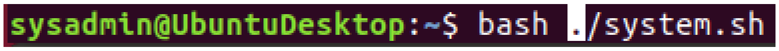
Command to make the system.sh script executable:



```
sysadmin@UbuntuDesktop:~$ chmod +x system.sh
```

Optional

- Commands to test the script and confirm its execution:



```
sysadmin@UbuntuDesktop:~$ bash ./system.sh
```

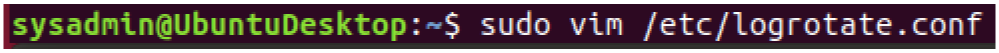
Bonus

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

Sudo cp system.sh /etc/cron.weekly

Step 4. Manage Log File Sizes

1. Run sudo nano /etc/logrotate.conf to edit the logrotate configuration file.

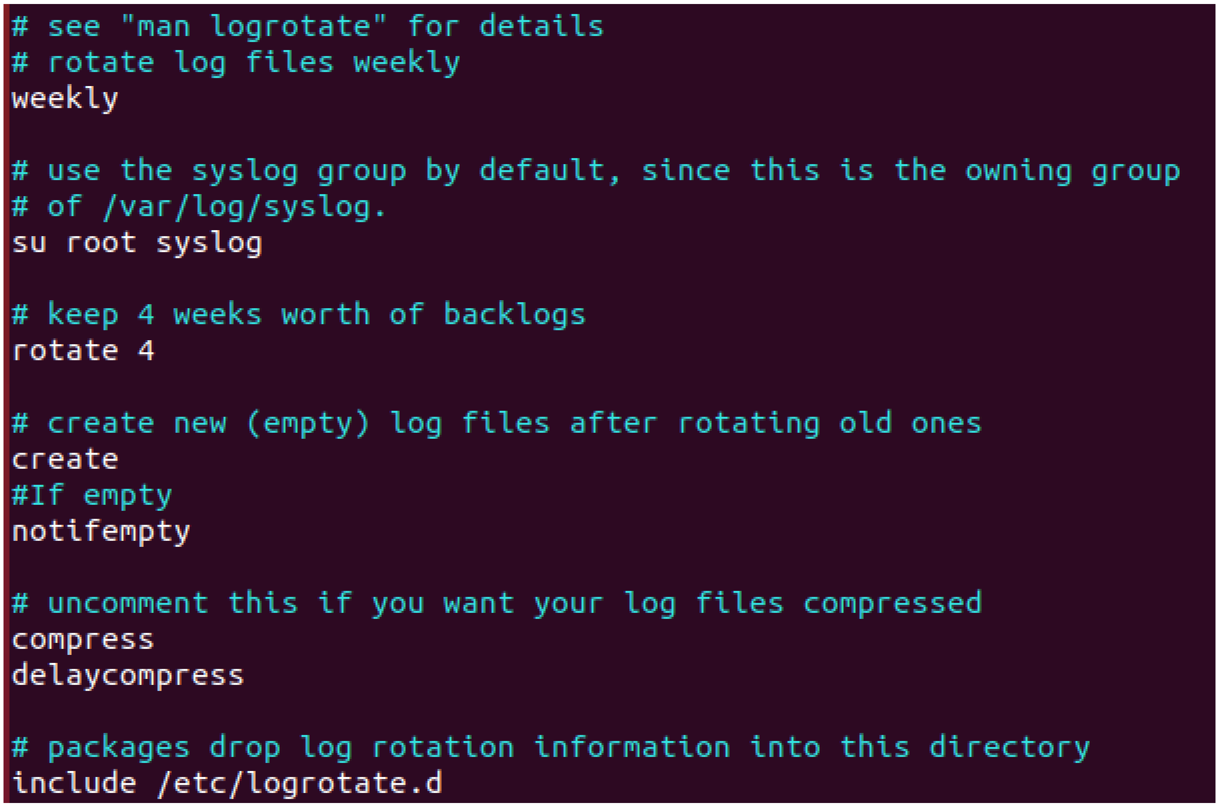


```
sysadmin@UbuntuDesktop:~$ sudo vim /etc/logrotate.conf
```

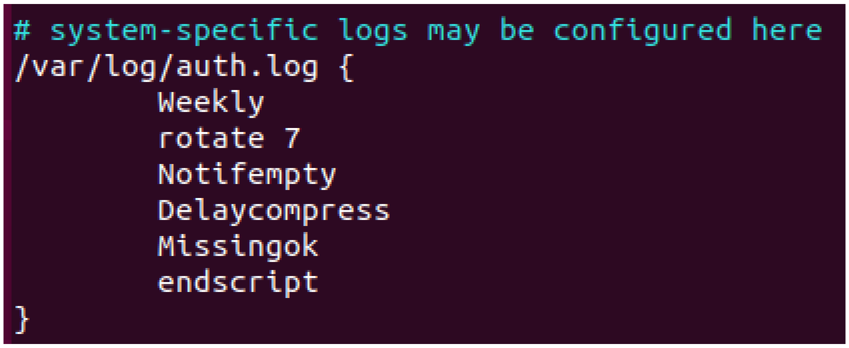
Configure a log rotation scheme that backs up authentication messages to the /var/log/auth.log.

- Add your config file edits below:

2.



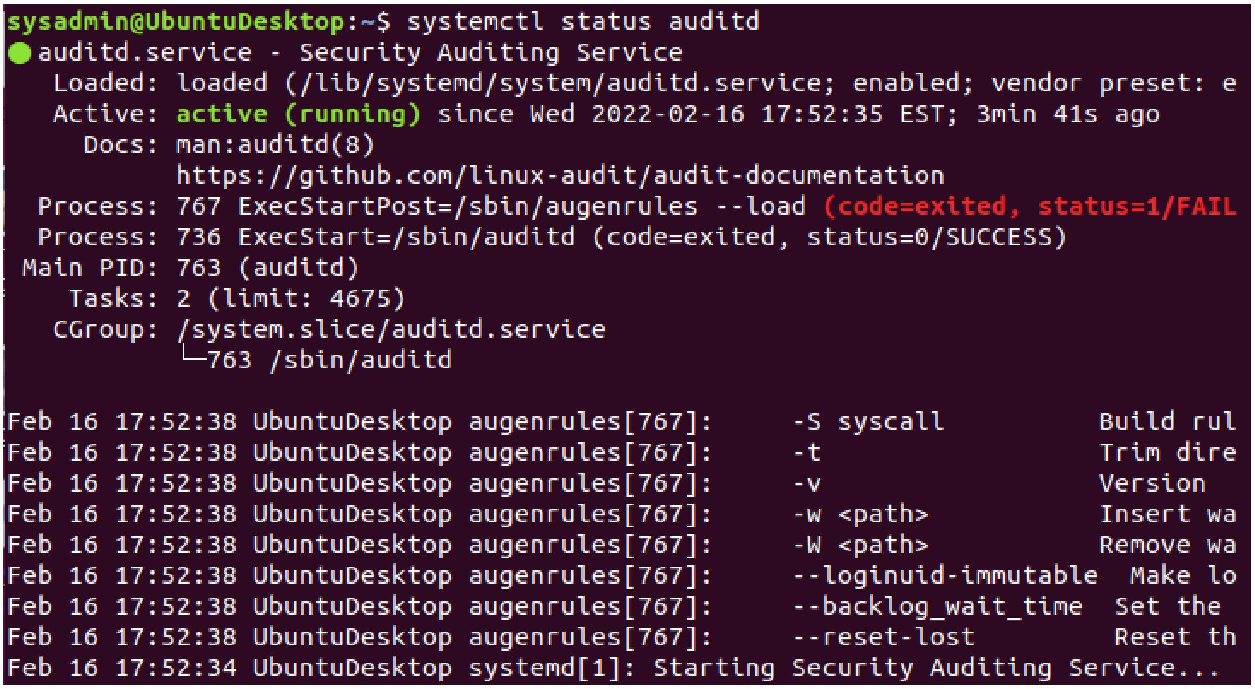
```
# see "man logrotate" for details
# rotate log files weekly
weekly

# use the syslog group by default, since this is the owning group
# of /var/log/syslog.
su root syslog

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create
#If empty
notifempty

# uncomment this if you want your log files compressed
compress
delaycompress

# packages drop log rotation information into this directory
include /etc/logrotate.d
```

3. 

```
# system-specific logs may be configured here
/var/log/auth.log {
        Weekly
        rotate 7
        Notifempty
        Delaycompress
        Missingok
        endscript
}
```

Bonus: Check for Policy and File Violations

1. Command to verify auditd is active:



```
sysadmin@UbuntuDesktop:~$ systemctl status auditd
● auditd.service - Security Auditing Service
   Loaded: loaded (/lib/systemd/system/auditd.service; enabled; vendor preset: e
   Active: active (running) since Wed 2022-02-16 17:52:35 EST; 3min 41s ago
     Docs: man:auditd(8)
           https://github.com/linux-audit/audit-documentation
  Process: 767 ExecStartPost=/sbin/augenrules --load (code=exited, status=1/FAIL
  Process: 736 ExecStart=/sbin/auditd (code=exited, status=0/SUCCESS)
 Main PID: 763 (auditd)
    Tasks: 2 (limit: 4675)
   CGroup: /system.slice/auditd.service
           └─763 /sbin/auditd

Feb 16 17:52:38 UbuntuDesktop augenrules[767]:     -S syscall          Build rul
Feb 16 17:52:38 UbuntuDesktop augenrules[767]:     -t                  Trim dire
Feb 16 17:52:38 UbuntuDesktop augenrules[767]:     -v                  Version
Feb 16 17:52:38 UbuntuDesktop augenrules[767]:     -w <path>           Insert wa
Feb 16 17:52:38 UbuntuDesktop augenrules[767]:     -W <path>           Remove wa
Feb 16 17:52:38 UbuntuDesktop augenrules[767]:     --loginuid-immutable  Make lo
Feb 16 17:52:38 UbuntuDesktop augenrules[767]:     --backlog_wait_time  Set the
Feb 16 17:52:38 UbuntuDesktop augenrules[767]:     --reset-lost         Reset th
Feb 16 17:52:34 UbuntuDesktop systemd[1]: Starting Security Auditing Service...
```

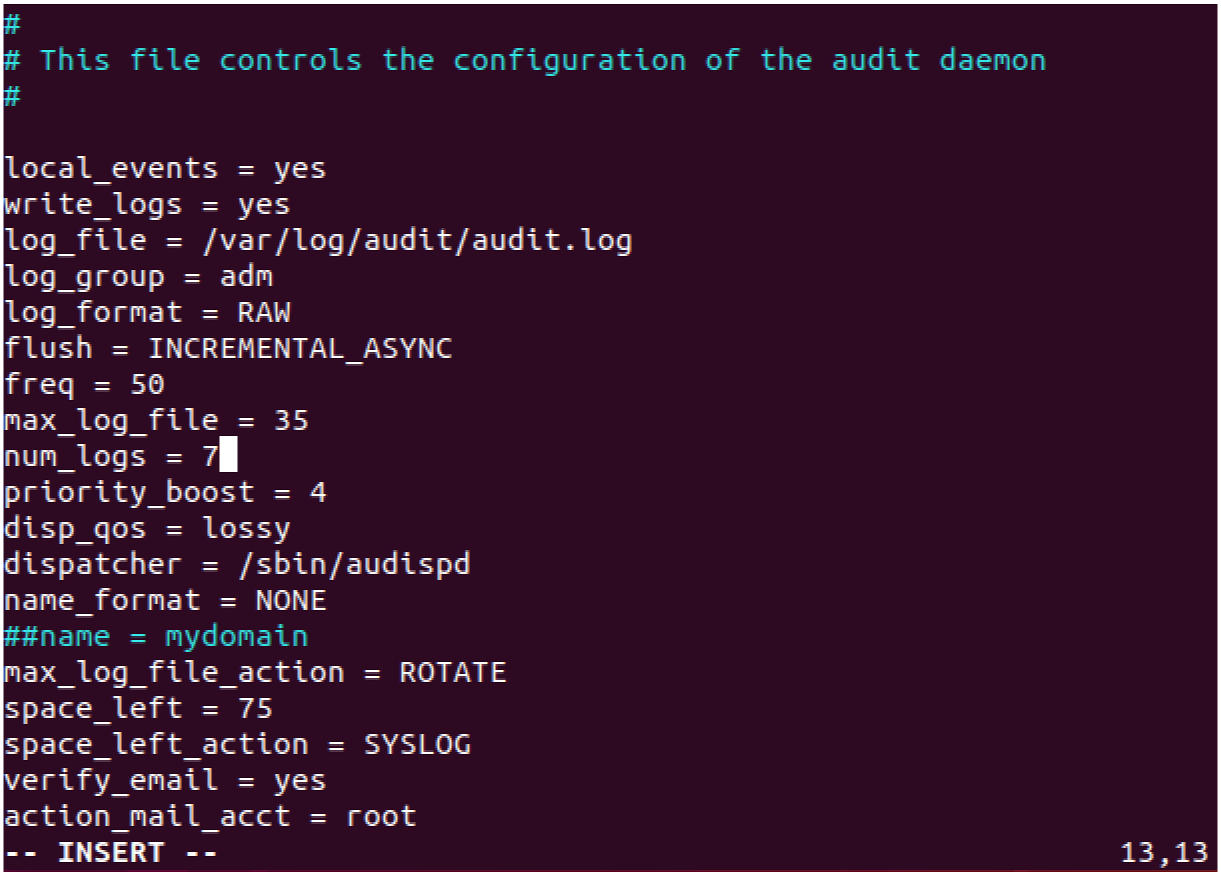
2. Command to set number of retained logs and maximum log file size:

   ○ Add the edits made to the configuration file below:

Sudo vim /etc/audit/auditd.conf

max_log_file = 35

num_logs = 7



```
#
# This file controls the configuration of the audit daemon
#

local_events = yes
write_logs = yes
log_file = /var/log/audit/audit.log
log_group = adm
log_format = RAW
flush = INCREMENTAL_ASYNC
freq = 50
max_log_file = 35
num_logs = 7
priority_boost = 4
disp_qos = lossy
dispatcher = /sbin/audispd
name_format = NONE
##name = mydomain
max_log_file_action = ROTATE
space_left = 75
space_left_action = SYSLOG
verify_email = yes
action_mail_acct = root
-- INSERT --                                                  13,13
```

3. Command using auditd to set rules for /etc/shadow, /etc/passwd and /var/log/auth.log:

    ○ Sudo vim /etc/audit/rules.d/audit.rules

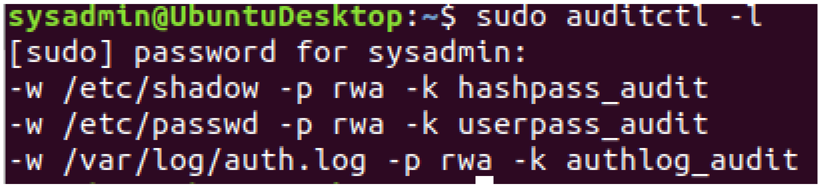
    Permissions:

    -w /etc/shadow -p wra -k hashpass_audit

    -w /etc/shadow -p wra -k userpass_audit

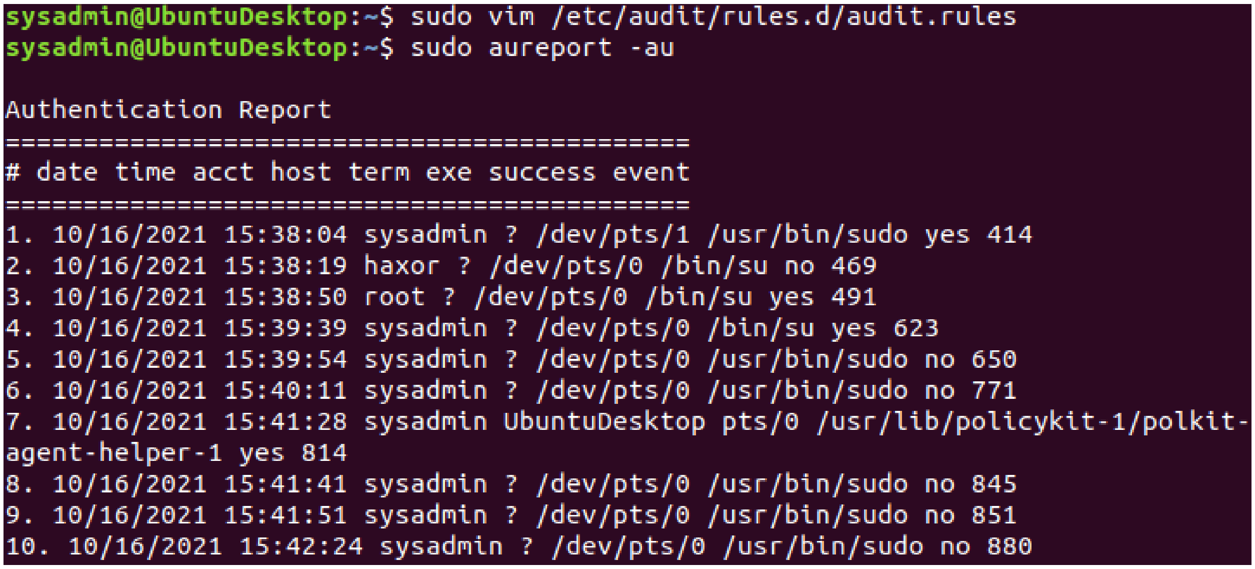
    -w /var/log/auth.log -p wra -k authlog_audit

4. Command to restart auditd:
   Sudo systemctl restart auditd
5. Command to list all auditd rules:
   Sudo auditctl -l



```
sysadmin@UbuntuDesktop:~$ sudo auditctl -l
[sudo] password for sysadmin:
-w /etc/shadow -p rwa -k hashpass_audit
-w /etc/passwd -p rwa -k userpass_audit
-w /var/log/auth.log -p rwa -k authlog_audit
```

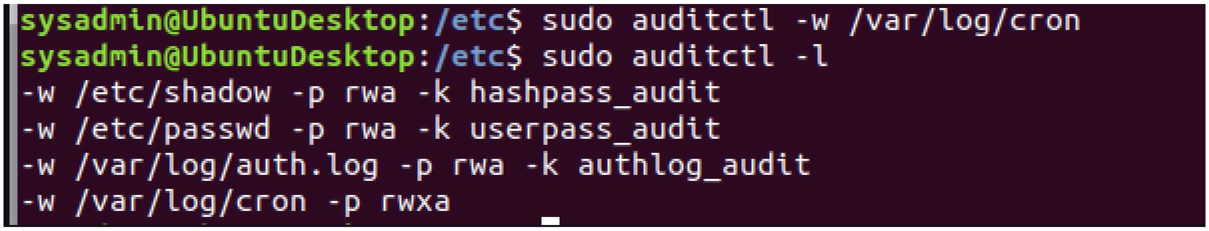
6. Command to produce an audit report:
   Sudo aureport -au



```
sysadmin@UbuntuDesktop:~$ sudo vim /etc/audit/rules.d/audit.rules
sysadmin@UbuntuDesktop:~$ sudo aureport -au

Authentication Report
============================================
# date time acct host term exe success event
============================================
1. 10/16/2021 15:38:04 sysadmin ? /dev/pts/1 /usr/bin/sudo yes 414
2. 10/16/2021 15:38:19 haxor ? /dev/pts/0 /bin/su no 469
3. 10/16/2021 15:38:50 root ? /dev/pts/0 /bin/su yes 491
4. 10/16/2021 15:39:39 sysadmin ? /dev/pts/0 /bin/su yes 623
5. 10/16/2021 15:39:54 sysadmin ? /dev/pts/0 /usr/bin/sudo no 650
6. 10/16/2021 15:40:11 sysadmin ? /dev/pts/0 /usr/bin/sudo no 771
7. 10/16/2021 15:41:28 sysadmin UbuntuDesktop pts/0 /usr/lib/policykit-1/polkit-
agent-helper-1 yes 814
8. 10/16/2021 15:41:41 sysadmin ? /dev/pts/0 /usr/bin/sudo no 845
9. 10/16/2021 15:41:51 sysadmin ? /dev/pts/0 /usr/bin/sudo no 851
10. 10/16/2021 15:42:24 sysadmin ? /dev/pts/0 /usr/bin/sudo no 880
```

7. Create a user with sudo useradd attacker and produce an audit report that lists account modifications:
   Sudo aureport -m
8. Command to use auditd to watch /var/log/cron:
   Sudo auditctl -w /var/log/cron
9. Command to verify auditd rules:

   sudo auditctl -l



```
sysadmin@UbuntuDesktop:/etc$ sudo auditctl -w /var/log/cron
sysadmin@UbuntuDesktop:/etc$ sudo auditctl -l
-w /etc/shadow -p rwa -k hashpass_audit
-w /etc/passwd -p rwa -k userpass_audit
-w /var/log/auth.log -p rwa -k authlog_audit
-w /var/log/cron -p rwxa
```