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:

   tar xvvf TarDocs.tar

2. Command to create the Javaless_Doc.tar archive from the TarDocs/ directory, while

   excluding the TarDocs/Documents/Java directory:

   tar --exclude='Java' -cvvf Javaless_Docs.tar TarDocs/Documents

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

   tar tvvf Javaless_Docs.tar | grep 'Java'

   Or

   tar tvvwf Javaless_Docs.tar (TO VERIFY ONLY TARDOCS/DOCUMENTS/)
   drwxr-xr-x sysadmin/sysadmin 0 2021-10-04 23:24 TarDocs/Documents/

Bonus:

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

1. (Full backup)

sudo tar cvvwf logs_backup_zero.tar --listed-incremental=**logs_backup.snar** --level=0 /var/log

2. (Create an incremental with changes made after a full backup!)

sudo tar cvvwf logs_backup_tar.gz --listed-incremental=**logs_backup.snar** /var/log

Critical Analysis Question:

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

You can not use -x and -c at the same time since they are used for two completely different purposes, and can not be used in the same command line since they clash with one another in respect to what they were created to do in the command line. The -c is to **create an information** backed archive, and -x is to **extract** information from a backed archive.

Step 2: Create, Manage, and Automate Cron Jobs

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

1. Sudo systemctl status crontab (To check that crontab is active & running)

2. Sudo crontab -l (Listing the content of the crontab with -l)
3. Sudo crontab -e (Editing crontab with -e)

Inside crontab <-------

4. Choose #1 nano & Enter (Select an editor to use. #1 for /bin/nano)

5. 0 6 * * 3 Tar cvvf auth_backup.tgz --listed-incremental=auth_backup.snar --level=0 /var/log/auth.log

Outside crontab←---------

6. Sudo gzip auth_backup.tgz (Use gzip to compress)
7. auth_backup.tgz.gz (Final result)

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
free_mem_tool > ~/backups/freemem/free_mem.txt

# Disk usage output to a disk_usage.txt file
disk_use_tool > ~/backups/diskuse/disk_usage.txt

# List open files to a open_list.txt file
list_open_tool > ~/backups/openlist/open_list.txt

# Free disk space to a free_disk.txt file
free_disk_tool > ~/backups/freedisk/free_disk.txt
```

Then:

```
# Free memory output to a free_mem.txt file
free -h > ~/backups/freemem/free_mem.txt

# Disk usage output to a disk_usage.txt
du -h > ~/backups/diskuse/disk_usage.txt
```

Then: (changes: free -h, du -h, lsof, df -h)

```
# List open files to a open_list.txt file
lsof > ~/backups/openlist/open_list.txt

# Free disk space to a free_disk.txt file
df -h > ~/backups/freedisk/free_disk.txt
```

1. Command to make the system.sh script executable:

   chmod u+x system.sh

Optional:

Commands to test the script and confirm its execution**:**

cat backups/freedisk/free_disk.txt

cat backups/openlist/open_list.txt

cat backups/diskuse/disk_usage.tx

cat backups/freemem/free_mem.txt

Bonus:

Command to copy system to system-wide cron directory:

1. Put in crontab to run it weekly
2. open crontab -e to input:

* * * * 3 sh system.sh /home/sysadmin/system.sh ←--------bash script in crontab
sudo cp system.sh /etc/cron.weekly/ ←----------Or

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.

   sudo nano /etc/logrotate.conf (To edit logrotate)

   # Rotate log files weekly:

   weekly

   # Keep 7 logs:

   rotate 7

   # Do not rotate the log if it is empty:

   notifempty

   # Old log files are compressed with gzip (default). Uncomment if needed compressed:
   Compress

   # Postpone compress of previous log file to next rotation cycle:
   delaycompression

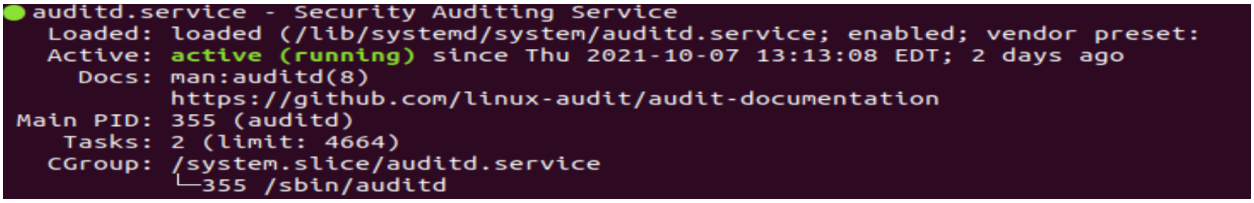
   # If log files are missing, go to the next one without an error message:
   missingok

   /var/log/auth.log {
   weekly
   rotate 7
   notifempty
   compress
   delaycompression
   missingok
   endscript
   }

Bonus: Check for Policy and File Violations

1. Command to verify auditd is active:

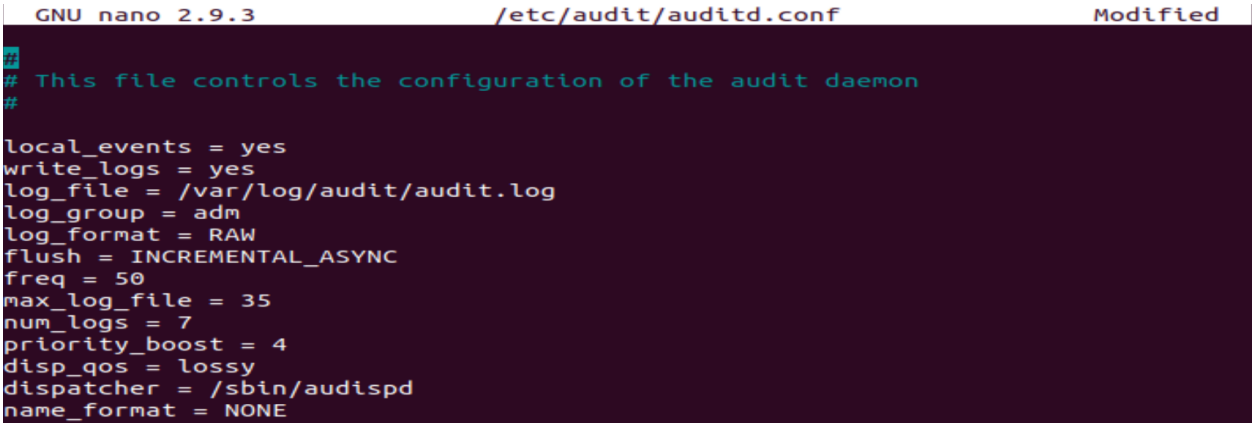
   systemctl status auditd



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

   sudo nano /etc/audit/auditd.conf

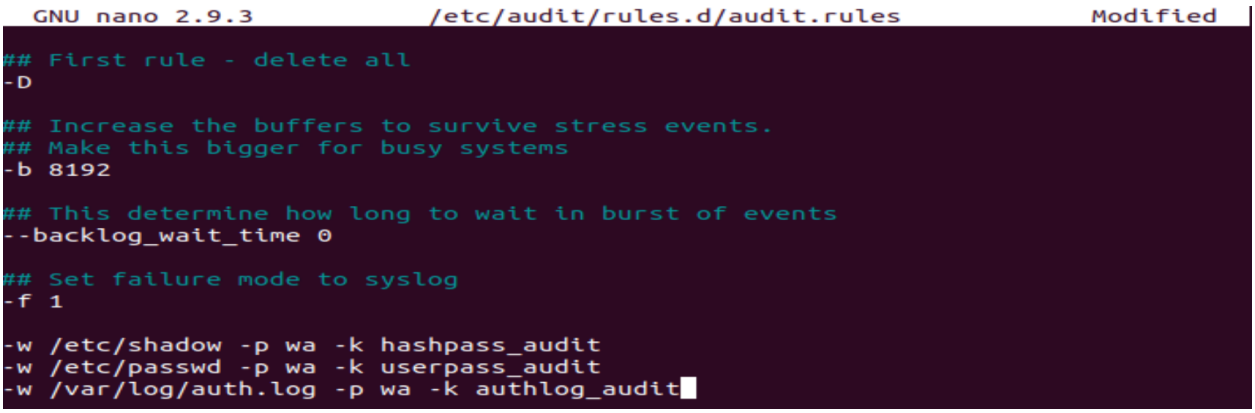
   Add the edits made to the configuration file below:



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

   sudo nano /etc/audit/rules.d/audit.rules (to add from nano audit)

Add the edits made to the rules file below:



4. Command to restart auditd:

   sudo systemctl restart auditd

5. Command to list all auditd rules:

   sudo auditctl -l ----------> (audit rules)

   -w /tc/shadow -p wa -k hashpass_aduit

   -w /etc/passwd -p wa -k userpass_aduit

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

6. Command to produce an audit report:

   sudo aureport -au ----------> (Authentication Repot)

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

   sudo useradd attacker (step 1)

sudo aureport -m (step 2)

```
Account Modifications Report
============================================
# date time auid addr term exe acct success event
============================================
1. 10/05/2021 23:47:21 1000 UbuntuDesktop pts/1 /usr/sbin/useradd criminal yes
282
2. 10/05/2021 23:47:21 1000 UbuntuDesktop pts/1 /usr/sbin/useradd ? yes 283
3. 10/05/2021 23:48:18 1000 UbuntuDesktop pts/1 /usr/sbin/useradd criminal no 3
14
4. 10/05/2021 23:48:18 1000 UbuntuDesktop pts/1 /usr/sbin/useradd criminal no 3
15
5. 10/06/2021 00:10:08 1000 UbuntuDesktop pts/1 /usr/sbin/useradd criminal1 yes
 933
6. 10/06/2021 00:10:08 1000 UbuntuDesktop pts/1 /usr/sbin/useradd ? yes 934
7. 10/06/2021 02:27:39 -1 ? ? /usr/sbin/useradd vboxadd no 264
8. 10/06/2021 02:27:39 -1 ? ? /usr/sbin/useradd vboxadd no 265
9. 10/06/2021 02:27:39 -1 ? ? /usr/sbin/useradd vboxadd no 266
10. 10/06/2021 02:27:39 -1 ? ? /usr/sbin/useradd vboxadd no 267
11. 10/06/2021 02:29:04 -1 ? ? /usr/sbin/useradd vboxadd no 266
12. 10/06/2021 02:29:04 -1 ? ? /usr/sbin/useradd vboxadd no 267
13. 10/06/2021 02:29:04 -1 ? ? /usr/sbin/useradd vboxadd no 268
14. 10/06/2021 02:29:04 -1 ? ? /usr/sbin/useradd vboxadd no 269
15. 10/09/2021 20:39:08 1000 UbuntuDesktop pts/1 /usr/sbin/useradd attacker yes
 8678
16. 10/09/2021 20:39:08 1000 UbuntuDesktop pts/1 /usr/sbin/useradd ? yes 8679
```

8. Command to use auditd to watch /var/log/cron:

   sudo auditctl -w /var/log/cron

   -w /var/log/cron -p rwxa ----------> (final result in auditctl -l)

9. Command to verify auditd rules:

   sudo auditctl -l

   ---------> (Audit rules available to view)

```
sysadmin@UbuntuDesktop: ~
File  Edit  View  Search  Terminal  Help
sysadmin@UbuntuDesktop:~$ sudo auditctl -l
-w /etc/shadow -p wa -k hashpass_audit
-w /etc/passwd -p wa -k userpass_audit
-w /var/log/auth.log -p wa -k authlog_audit
-w /var/log/cron -p rwxa
sysadmin@UbuntuDesktop:~$
```

Bonus (Research Activity): Perform Various Log Filtering Techniques

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

Journalctl --boot -p emerg..err

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

journalctl --disk-usage | less

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

sudo journalctl --vacuum-files=2

4. Command to filter all log messages with priority levels between zero and two, and save output to /home/sysadmin/Priority_High.txt:

journalctl --boot -p 0..2 > /home/student/Priority_High.txt

5. Command to automate the last command in a daily cron job. Add the edits made to the crontab file below:

The automated cron job is under the cron-daily which is under /etc/cron.daily

```
# 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
* * * * 3 ./home/sysadmin/system.sh
* * * * * journalctl --boot -p 0..2 > Priority_High.txt /home/student/
```