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:
   a. sudo tar -xvf TarDocs.tar

2.

   Command to **create** the Javaless_Doc.tar archive from the TarDocs/ directory, while excluding the TarDocs/Documents/Java directory:
   a. sudo tar -cvf Javaless_Doc.tar --exclude=TarDocs/Documents/Java TarDocs

3.

   Command to ensure Java/ is not in the new Javaless_Docs.tar archive:
   a. sudo tar -tvf 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 -zcvf logs_backup.tar.gz --listed-incremental=snapshot.file --level=0 /var/log

Critical Analysis Question

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

-x extracts an existing archive and -c creates a tar archive. You wouldnt use them together because it just doesnt make sense. You cant extract and create at the same time. Once you enter the file name you would get an error for either creating or extracting.

Step 2: Create, Manage, and Automate Cron Jobs

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

0 6 * * 3 sudo tar -zcvf /auth.backup.gz /var/log/auth.log

Step 3: Write Basic Bash Scripts

1. Brace expansion command to create the four subdirectories:

Sudo mkdir -p ~/backups/{freemem,diskuse,openlist,freedisk}

2. Paste your system.sh script edits below
#! /bin/bash

Print amount of free memory on system and save to freemem.txt

free > ~/backups/freemem/free_mem.txt

Print disk usage and save to disk_usage.txt

du -h > ~/backups/diskuse/disk_usage.txt

Lists all open files and saves in to open_list.txt

ps -aef > ~/backups/openlist/open_list.txt

Prints file system disk space statistics and saves it to free_disk.txt

df -kh > ~/backups/freedisk/free_disk.txt

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

Sudo chmod +x system.sh

Optional - Commands to test the script and confirm its execution:

Sudo ./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. 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 {
rotate 7
weekly
notifempty
delaycompress
missingok
}
```

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:

Num_logs = 7
Max_log_file = 35

Add the edits made to the configuration file below:

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

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

-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

Add the edits made to the rules file below:

```
## First rule - delete all
-D
-w /etc/shadow -p wa -k shadow
-w /etc/passwd -p wa -k passwd
-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
## Increase the buffers to survive stress events.
## Make this bigger for busy systems
-b 8192

## This determine how long to wait in burst of events
--backlog_wait_time 0

## Set failure mode to syslog
-f 1
```

4. Command to restart auditd :

systemctl restart auditd

5. Command to list all auditd rules:

sudo auditctl -l

6. Command to produce an audit report:

    sudo aureport -au

7.

    Create a user with sudo useradd attacker and produce an audit report that lists account modifications:
    a. Sudo aureport -m
        i. 10. 01/28/2021 20:01:30 1000 UbuntuDesktop pts/0 /usr/sbin/groupadd ? yes 30124
        ii. 11. 01/28/2021 20:01:30 1000 UbuntuDesktop pts/0 /usr/sbin/groupadd ? yes 30125
        iii. 12. 01/28/2021 20:01:30 1000 UbuntuDesktop pts/0 /usr/sbin/groupadd ? yes 30126
        iv. 13. 01/28/2021 20:01:30 1000 UbuntuDesktop pts/0 /usr/sbin/useradd ? yes 30134
        v. 14. 01/28/2021 20:01:36 1000 UbuntuDesktop pts/0 /usr/bin/passwd attacker yes 30173
        vi.
        vii.

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

        sudo auditctl -w /var/log/cron -p wra -k Cron_audit

9.

Command to verify auditd rules:

a. Sudo auditctl -l

Bonus (Research Activity): Perform Various Log Filtering Techniques

1.

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

a. sudo journalctl -b -p "0.."3"

2.

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

a. Sudo journalctl --disk-usage

3.

Comand to remove all archived journal files except the most recent two:

a. 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 :

a. sudo journalctl -b -p "0.."2" > /home/sysadmin/Priority_High.txt

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

[Your solution cron edits here]

b. 0 0 * * * sudo journalctl -b -p "0.."2" > /home/sysadmin/Priority_High.txt
c.