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/
```

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

2. #For memory:
3.
4. free -m > backups/freemem/free_mem.txt
5.
6. #For disk usage in human readable form:
7.
8. df -BM -h > backups/diskuse/disk_usage.txt
9.
10. #For all open files:
11.
12. lsod > backups/openlist/open_list.txt
13.
14. #For file system disk space and statistics:
15.
16. df -k -BM -h | awk '{print $1,$4}' > backups/freedisk/free_disk.txt
17.
18. #End of script
19.

```
#!/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:

Not performed

Step 4. Manage Log File Sizes

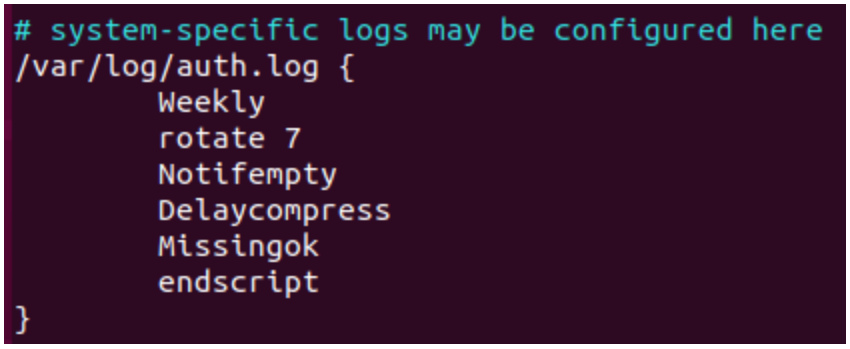
1. Run sudo nano /etc/logrotate.conf to edit the logrotate configuration file.

   ```
   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:
2. Command to set number of retained logs and maximum log file size:
    - Add the edits made to the configuration file below:
3. [Your solution edits here]
4. 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:
5. [Your solution edits here]
6. Command to restart auditd:
7. Command to list all auditd rules:
8. Command to produce an audit report:
9. Create a user with sudo useradd attacker and produce an audit report that lists account modifications:
10. Command to use auditd to watch /var/log/cron:
11. Command to verify auditd rules:

Bonus (Research Activity): Perform Various Log Filtering Techniques

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

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

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

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

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

   [Your solution cron edits here]