Week 5: 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

a. Command to **extract** the `TarDocs.tar` archive to the current directory:
   a) Sudo tar -xvpzf mybackup.tar.gz -C /documents

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

   a. Sudo tar -cvpzf mybackup.tar.bz2 --exclude=/mnt /

```
drwx------   2 root       root        16K Nov 12  2019 lost+found
drwxr-xr-x   3 root       root       4.0K Nov 12  2019 media
drwxr-xr-x   2 root       root       4.0K Aug  5  2019 mnt
-rw-r--r--   1 root       root       219M Jan 14 01:25 mybackup.tar.bz2
drwxr-xr-x   8 root       root       4.0K Oct  2 15:23 opt
```

c. Command to ensure `Java/` is not in the new `Javaless_Docs.tar` archive:
   a) sudo tar -xvpzf mybackup.tar.gz2 -C /recover

```
sysadmin@UbuntuDesktop:/recover$ ls
boot  etc  initrd.img  opt  proc  run  srv  vagrant
```

No mnt file

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 -cvpzf logs_backup_tar.gz /var/ snapshot.file

Critical Analysis Question

- Why wouldn't you use the options `-x` and `-c` at the same with `tar`?
  - -x is to extract and -c is to create; if a file tar file has not been created cannot be extracted

---

Step 2: Create, Manage, and Automate Cron Jobs

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

```
# m h  dom mon dow   command
59 20 * * 1 /bin/sh /var/log/auth.log/logs.sh
```

```
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                                    logs.sh

touch /home/user/Desktop/file.txt
```

```
:~$ sudo chmod u+x logs.sh
```

Step 3: Write Basic Bash Scripts

1. Brace expansion command to create the four subdirectories:
2. Paste your `system.sh` script edits below:
3. `#!/bin/bash`
   `[Your solution script contents here]`

```
sysadmin@UbuntuDesktop:~$ nano hmwrk.sh
```

```
Activities   Terminal ▾                    Mon 04:04
                                  sysadmin@UbuntuDesktop: ~
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                          hmwrk.sh

#!/bin/bash

# My first simple script to practice bash "hello world: It works" from Cyberbootcamp

echo "hello world; it works my very first script - test script"

                                  [ Read 8 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Linter   ^_ Go To Line
```

```
-rw-r--r-- 1 sysadmin sysadmin     141 Jan 14 19:07 hmwrk.sh
```

4. Command to make the `system.sh` script executable:
   1. chmod u+x hmwrk.sh

```
-rwxr--r-- 1 sysadmin sysadmin     141 Jan 14 19:07 hmwrk.sh
```

Optional

- Commands to test the script and confirm its execution:

```
sysadmin@UbuntuDesktop:~$ bash hmwrk.sh
hello world; it works my very first script - test script
```

Bonus

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

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:

   `[Your logrotate scheme edits here]`

```
  GNU nano 2.9.3                    /etc/logrotate.conf

# see "man logrotate" for details
# rotate log files everyday
daily

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

# keep 6 weeks worth of backlogs
rotate 6

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

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

# packages drop log rotation information into this directory
include /var/log/auth.log
```

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:
    -

```
sysadmin@UbuntuDesktop:~$ dpkg --get-selections | grep auditd
sysadmin@UbuntuDesktop:~$ apt-cache search auditd
auditd - User space tools for security auditing
snoopy - execve() wrapper and logger
sysadmin@UbuntuDesktop:~$ sudo apt-get install auditd
```

```
sysadmin@UbuntuDesktop:~$ service auditd status
● auditd.service - Security Auditing Service
   Loaded: loaded (/lib/systemd/system/auditd.service; enabled; vendor preset: enabled)
```

`[Your solution edits here]`

3. 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:

```
sysadmin@UbuntuDesktop:~$ sudo auditctl -w /etc/passwd -p wrxa -k passwd1-filter
```

[Your solution edits here]

4. Command to restart `auditd`:

```
sysadmin@UbuntuDesktop:/etc$ service auditd start
```

5. Command to list all `auditd` rules:

```
sysadmin@UbuntuDesktop:/etc$ nano /etc/audit/audit.rules
```

6. Command to produce an audit report:

```
sysadmin@UbuntuDesktop:/etc$ service auditd start
```

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

```
attacker:x:1012:1014::/home/attacker:/bin/sh
```

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

```
467  /etc/init.d/auditd
468  systemctl start auditd.service
```

9. Command to verify `auditd` rules:

```
sysadmin@UbuntuDesktop:/$ aureport

Summary Report
======================
Error opening config file (Permission denied)
NOTE - using built-in logs: /var/log/audit/audit.log
Range of time in logs: 01/14/2021 03:16:50.207 - 01/14/2021 20:06:33.445
Selected time for report: 01/14/2021 03:16:50 - 01/14/2021 20:06:33.445
Number of changes in configuration: 163
Number of changes to accounts, groups, or roles: 6
Number of logins: 1
Number of failed logins: 0
Number of authentications: 19
Number of failed authentications: 3
Number of users: 6
Number of terminals: 10
Number of host names: 3
Number of executables: 25
Number of commands: 21
Number of files: 4
Number of AVC's: 0
Number of MAC events: 0
Number of failed syscalls: 5
Number of anomaly events: 1
```

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]`