## 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:

```
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:

```
sudo tar cvf Javaless_Docs.tar --exclude ./Java ./
```

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

```
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 tar cvzf 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`?

--- Because one X is used to extract and C is to create tars.

### Step 2: Create, Manage, and Automate Cron Jobs

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

```
0 18 * * 3 sudo tar czf auth_backup.tgz -P /var/log/auth.log
```

### Step 3: Write Basic Bash Scripts

1. Brace expansion command to create the four subdirectories:

```
mkdir ./backups/{freemem,diskuse,openlist,freedisk}
```

2. Paste your `system.sh` script edits below:

```bash
#!/bin/bash

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

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

# List open files to a open_list.txt file
sudo lsof -u sysadmin >  ~/Projects/backups/diskuse/open_list.txt

# Free disk space to a free_disk.txt file
sudo df -h > ~/Projects/backups/diskuse/disk_usage.txt
```

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

```
Sudo +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: Perform Various Log Filtering Techniques

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

```
sudo journalctl -p err -b -0
```

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

```
sudo journalctl -u systemd-journald -b -0
```

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

```
sudo journalctl --vacuum-files=2
```

Bonus

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

sudo journalctl -p crit -b -0 > /home/sysadmin/priority_high.txt

- Command to automate the last command in a daily cronjob:

crontab -e

- Add the edits made to the crontab file below:

   ```bash
   [sudo journalctl -p crit -b -0 > /home/sysadmin/priority_high.txt]
   ```

### Step 5. Create Priority-Based Log Files

1. Command to record all mail log messages, except for debug, to `/var/log/mail.log`:

sudo nano /etc/rsyslog.d/50-default.conf

   - Add the edits made to the configuration file below:

   ```bash
[mail.*                 -/var/log/mail.log
mail.info               -/var/log/mail.info
mail.warn                 -/var/log/mail.warn
mail.err                /var/log/mail.err
*.=debug;\
     mail.none     -/var/log/debug]
Or

[mail.!=debug /var/log/mail.log]
   ```

Bonus

- Command to record all boot log messages, except for info and debug, to `/var/log/boot.log`:

sudo nano /etc/rsyslog.d/50-default.conf

local7.notice   -var/log/notice.log

   - Add the edits made to the configuration file below:

   ```bash
   [    local7.*                -/var/log/boot.log
        local7.notice           -var/log/notice.log
]
   ```

### Step 6. 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:

   ```bash
[/var/log/auth.log {
        rotate 7
        weekly
        missingok
        compress
        delaycompress
        notifempty
        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 auditd.conf

   - Add the edits made to the configuration file below:

   ```bash
   [max_log_file = 35
num_logs = 7
]
   ```

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:

```bash
[-w /etc/passwd -p wra -k userpass_audit
-w /etc/shadow -p wra -k hashpass_audit
-w /var/log/auth.log -p wra -k authlog_audit
]
```

4. Command to restart `auditd`:

```
systemctl restart auditd
```

5. Command to list all `auditd` rules:

```
auditctl -l
```

6. Command to produce an audit report:

```
aureport -au
```

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

```
1. 08/15/2020 16:09:05 1000 Emilianos_Ubuntu pts/2 /usr/sbin/groupadd ? yes 232
2. 08/15/2020 16:09:05 1000 Emilianos_Ubuntu pts/2 /usr/sbin/groupadd ? yes 233
3. 08/15/2020 16:09:05 1000 Emilianos_Ubuntu pts/2 /usr/sbin/groupadd ? yes 234
4. 08/15/2020 16:09:05 1000 Emilianos_Ubuntu pts/2 /usr/sbin/useradd ? yes 237
5. 08/15/2020 16:09:11 1000 Emilianos_Ubuntu pts/2 /usr/bin/passwd attacker yes
```

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

```
-w /var/log/cron -p wra -k cron_audit
```

9. Command to verify `auditd` rules:

```
aureport -m
```
