Submission File for Unit 5: Archiving and Logging Data Homework

Please edit and save this file while updating it with the commands and file (configuration and rules) edits you used to solve your homework.

`tar`: Create, extract, compress, and manage tar backup archives

Command to **extract** the `TarDocs.tar` archive to the current directory:

- `tar -xvf TarDocs.tar.gz`

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

- `tar -czvf ~/Projects/Javaless_Doc.tar.gz ~/Projects/TarDocs/Financials
~/Projects/TarDocs/Movies ~/Projects/TarDocs/Pictures ~/Projects/TarDocs/Programs
~/Projects/TarDocs/Documents/c++interviewquestions.pdf
~/Projects/TarDocs/Documents/Design-Patterns
~/Projects/TarDocs/Documents/Google-Maps-Hacks
~/Projects/TarDocs/Documents/IntelliJIDEA_ReferenceCard.pdf
~/Projects/TarDocs/Documents/Music-Sheets`

Ensuring `Java/` is not in the new `Javaless_Docs.tar` archive:

- `tar -tvf ~/Projects/Javaless_Doc.tar.gz`

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

- `tar cvvWf ~/Projects/logs_backup_tar.gz --listed-incremental=snapshot.file --level 0
~/Projects/Javaless_Doc.tar.gz`

`tar` Critical Thinking

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

Because -x extracts contents and -c creates them, so if you used both then it would make no sense. Since you would be both creating an archive, then extract the archive to the exact same location nothing would change.

`cron`: Create, manage and automate various cron jobs

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

#Run command
- `vi backup.sh`

#add to vi
- `tar-cJpf /var/log/auth_backup.tar.gz /var/log/auth.log`

#Run Command
- `crontab -u sysadmin -e`

Add to crontab
- `0 6 * * 3 ~/Projects/backup.sh`

`bash scripting`: Write basic bash scripts

Brace expansion command to create the four subdirectories:

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

Command and file edit to create `system.sh` (you can copy and paste it here):

- `nano system.sh`

- Within the script, have the following:

```
bash
#!/bin/bash
free -mh >> ~/backups/freemem/free_mem.txt
df -mh | awk '{print $1,$2,$3}' >> ~/backups/diskuse/disk_usage.txt
lsof >> ~/backups/openlist/open_list.txt
df -mh | awk '{print $1,$2,$4}' >> ~/backups/freedisk/free_disk.txt
```

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

- `chmod -x system.sh`

Commands to to confirm script's execution:

- `cat ~/backups/freemem/free_mem.txt`

- `cat ~/backups/diskuse/disk_usage.txt`

- `cat ~/backups/openlist/open_list.txt`

- `cat ~/backups/freedisk/free_disk.txt`

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

- `crontab -u sysadmin -e`
- `0 6 * * 3 ~/sysadmin.sh`

`journalctl`: Perform various log filtering techniques

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

- `journalctl -b -1  -p "emerg".."err"`

Command to return `systemd-journald` messages:

- `journalctl | grep 'systemd-journald'`

Comand to prune archived journal files except the most recent 2:

- `journalctl -r | head -n 3`

Bonus Command to filter all log messages with priority levels between 0 and 2, output to `/home/sysadmin/Priority_High.txt`

- `journalctl -b -1  -p "emerg".."crit"`

Bonus 2 Command and file edit to automate the last command in a daily cronjob:

- `crontab -u sysadmin -e`

- Within the `crontab` file, add the following:

   ```bash
   `0 6 * * 3 ~/sysadmin.sh
   ```

`rsyslog`: Priority based log file creation

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

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

- Add within the configuration file:

   ```bash
mail.!info /var/log/mail.log
   ```

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

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

- Add within the configuration file:

   ```bash
   boot.!notice /var/log/boot.log
   ```

`logrotate`: Manage log file sizes

Command and file edit that backs up authentication messages to `/var/log/auth.log`:

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

- Add within the configuration file:

   ```bash
  include /var/log/auth.log
   ```

BONUS ACTIVITY `auditd`: Check for policy and file violations.

Command to verify `auditd` is active:

- `systemctl is-enabled auditd`

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

- `sudo nano /etc/audit/audit.conf`

- Add within the configuration file:

```
num_logs = 7
max_log_file = 30
```

Command and file edit using `auditd` itself to set rules for `/etc/shadow`, `/etc/passwd` and `/var/log/auth.log`:

- `sudo nano /etc/audit/rules.d/audit.rules`

- Add within the `rules` file:

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

Command to restart `auditd`:

- `sudo service auditd restart`

Command to list all `auditd` rules:

- `sudo auditd -l`

Command to produce an audit report:

- `aureport -au`

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

- `auditctl -w /var/log/cron -p wra -k cronchanges`

Command to re-verify `auditd` rules:

- `sudo auditd -l`
