**Week 5 Homework Submission File: Archiving and Logging Data**

**Step 1: Create, Extract, Compress, and Manage tar Backup Archives**

1. Command to **extract** the TarDocs.tar archive to the current directory:
   * tar -xvvf TarDocs.tar
2. Command to **create** the Javaless\_Doc.tar archive from the TarDocs/ directory, while excluding the TarDocs/Documents/Java directory:
   * tar cvvWf Javeless\_Doc.tar –exclude “TarDocs/Documents/Java” TarDocs/
3. Command to ensure Java/ is not in the new Javaless\_Docs.tar archive:
   * tar tvf Javaless\_Doc.tar | grep “Documents/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:
  + cd /var/log
  + tar --verbose --verbose --create --gzip --listed-incremental=/backup/snapshot.file --file=/backup/logs\_backup\_tar.gz

**Critical Analysis Question**

* Why wouldn't you use the options -x and -c at the same time with tar?
  + There would be a conflict when trying to do both. -c creates a directory that files will be placed vs. -x that extracts and writes files. You could use them in different lines but never on the same line code with the tar command.

**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 cvvf auth\_backup.tgz var/log/auth.log

**Step 3: Write Basic Bash Scripts**

1. Brace expansion command to create the four subdirectories:  
   sudo mkdir -p backups/{freemen,diskuse,openlist,freedisk}
2. Paste your system.sh script edits below:

#!/bin/bash  
echo 'free -h > ~/backups/freemem/free2\_mem.txt'  
echo 'du -h > ~/backups/diskuse/disk2\_usage.txt'  
echo 'lsof | wc -l > ~/backups/openlist/open2\_list.txt'  
echo 'df -h > ~/backups/freedisk/free2\_disk.txt'

1. Command to make the system.sh script executable:  
   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 { missingok weekly rotate 14 notifempty delaycompress}

**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:

[Your solution edits here]

1. 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:

[Your solution edits here]

1. Command to restart auditd:
2. Command to list all auditd rules:
3. Command to produce an audit report:
4. Create a user with sudo useradd attacker and produce an audit report that lists account modifications:
5. Command to use auditd to watch /var/log/cron:
6. 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]