## **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:  
   



1. Command to **create** the Javaless\_Doc.tar archive from the TarDocs/ directory, while excluding the TarDocs/Documents/Java directory:  
   
2. Command to ensure Java/ is not in the new Javaless\_Docs.tar archive:  
   

**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**

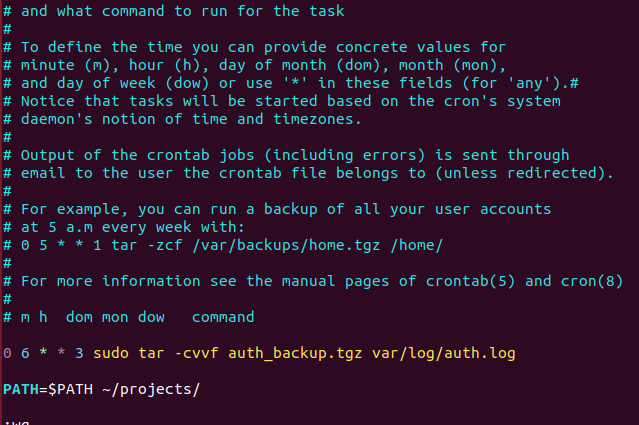
* Why wouldn't you use the options -x and -c at the same time with tar?

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:





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.



### **Step 3: Write Basic Bash Scripts**

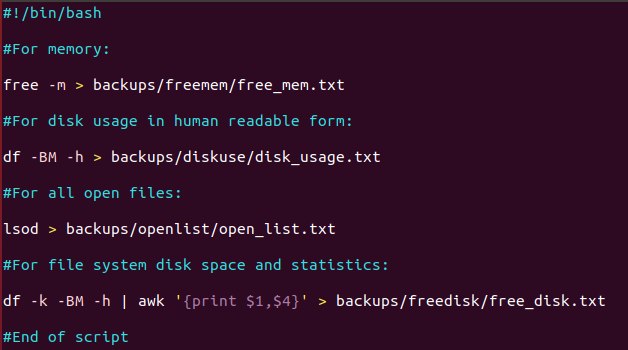
1. Brace expansion command to create the four subdirectories:   
   No directory is specified other than a general “backup” directory, so we will make one in our home directory including all four of the required directories.





Paste your system.sh script edits below:  
  
 #!/bin/bash

1. #For memory:
2. free -m > backups/freemem/free\_mem.txt
3. #For disk usage in human readable form:
4. df -BM -h > backups/diskuse/disk\_usage.txt
5. #For all open files:
6. lsod > backups/openlist/open\_list.txt
7. #For file system disk space and statistics:
8. df -k -BM -h | awk '{print $1,$4}' > backups/freedisk/free\_disk.txt
9. #End of script



Command to make the system.sh script executable:  


**Optional**

* Commands to test the script and confirm its execution:

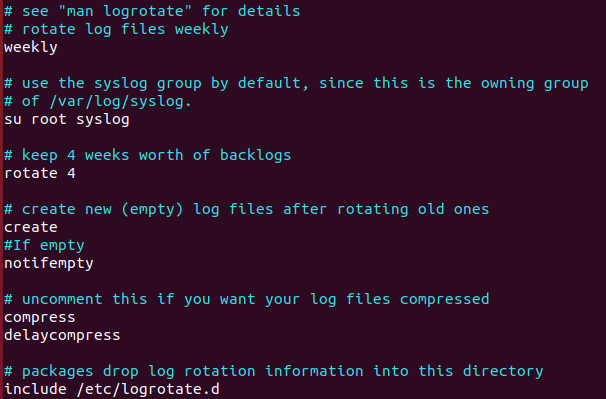
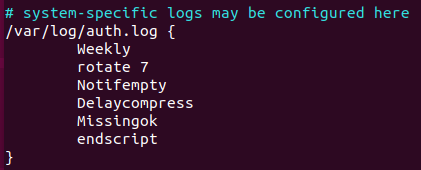


**Bonus**

* 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.  
     
    Configure a log rotation scheme that backs up authentication messages to the /var/log/auth.log.  
   * Add your config file edits below:
2. 
3. 

### **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]