## **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:

**sysadmin@UbuntuDesktop:~/Projects$ tar -xvf TarDocs.tar**

1. Command to **create** the Javaless\_Doc.tar archive from the TarDocs/ directory, while excluding the TarDocs/Documents/Java directory:

**tar --exclude=’home/sysadmin/Projects/TarDocs/Java’ -cvvf Javaless\_Docs.tar /home/sysadmin/Projects**

1. Command to ensure Java/ is not in the new Javaless\_Docs.tar archive:

**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 --listed-incremental=snapshot.file -cvvzf logs\_backup\_tar.gz /var/log**

#### **Critical Analysis Question**

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

**Ex: If I had 20 documents to send. I can not compress and extract at the same time. You will just undo your work that was created. You will get an error.**

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

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

**0 6 \* \* 3 tar -cvvzf /auth\_backup.tgz /var/log/auth.log >/dev/null 2>&1**

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

1. Brace expansion command to create the four subdirectories

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

1. Paste your system.sh script edits below:

[Your solution script contents here]

**#!/bin/bash**

**#Free memory output to a free\_mem.txt file**

**free -h > ~/backups/freemem/free\_mem.txt**

**# Disk usage output to a disk\_usage.txt file**

**du -h > ~/backups/diskuse/disk\_usage.txt**

**# List open files to a open\_list.txt file**

**lsof > ~/backups/openlist/open\_list.txt**

**# Free disk space to a free\_disk.txt file**

**df -h > ~/backups/freedisk/free\_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:

**Test the script: sudo ./system.sh**

**Confirm it’s execution: cat disk\_usage.txt**

**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:

[Your logrotate scheme edits here]

**/var/log/auth.log {**

**weekly**

**rotate 7**

**notifempty**

**delaycompress**

**missingok**

**}**

### **Bonus: Check for Policy and File Violations**

1. Command to verify auditd is active:

**Systemctl status auditd**

1. 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]

max\_log\_file = 35

num\_logs = 7

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]

**-w /etc/shadow -p wra -k hashpass\_audit**

**-w /etc/passwd -p wra -k userpass\_audit**

**-w /var/log/auth.log -p wra -k authlog\_audit**

1. Command to restart auditd:

**sudo service auditd restart**

1. Command to list all auditd rules:

**sudo auditctl -l**

1. Command to produce an audit report:

**sudo aureport -m**

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

sudo useradd attacker

1. Command to use auditd to watch /var/log/cron:

sudo auditctl -w /var/log/cron

1. Command to verify auditd rules:

sudo auditctl -l

### **Bonus (Research Activity): Perform Various Log Filtering Techniques**

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

sudo journalctl -b -p emerg..err

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

sudo journalctl -b -u=systemd-journald

1. Command to remove all archived journal files except the most recent two:

sudo journalctl --vacuum-files=2

1. Command to filter all log messages with priority levels between zero and two, and save output to /home/sysadmin/Priority\_High.txt:

/home/sysadmin# journalctl -p emerg..err > /home/student/Priority\_High.txt

1. Command to automate the last command in a daily cron job. Add the edits made to the crontab file below:  
     
    [Your solution cron edits here]

#Automate the last task by creating a cron job that runs daily in the user crontab.

@daily journalctl -p emerg..err > /home/student/Priority\_High.txt