## **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

**:~/Projects$ tar xvvf TarDocs.tar**

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

**tar cfv Javaless\_Docs.tar --exclude "Java" / Projects**

****

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

**tar -tfv Javaless\_Docs.tar | grep -i '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 cvvWf logs\_backup\_tar.gz - -listed-incremental= logs\_backup\_snar.gz - -level=0 /var/log**

#### **Critical Analysis Question**

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

**-c creates a new archive and -x extracts files from an archive. The archive needs to be created first and the files can be extracted.**

### **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 -cpzf /auth\_backup.tgz /var/log/auth.log.tar**

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

1. Brace expansion command to create the four subdirectories:

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

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

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

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

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

**df -h > ~/backups/freedisk/free\_disk.txt**

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

**chmod u+x system.sh**

**Optional**

* Commands to test the script and confirm its execution:

**sudo ./system.sh**

**Received the error message that was mentioned in the assignment to ignore. So I tested the script by do the following:**

**cat backups/freemem/free\_mem.txt**

**cat backups/diskuse/disk\_usage.txt**

**cat backups/openlist/open\_list.txt**

**cat backups/freedisk/free\_disk.txt**

**Bonus**

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

**cp system.sh /usr/bin/crontab**

***If I get this one wrong,can the grader put the answer in the grade comments so that I know what I did wrong?***

### **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 {**

**rotate 7**

**weekly**

**notifempty**

**delaycompress**

**missingok**

**endscript**

**}**

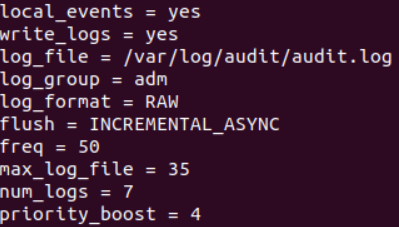
### **Bonus: Check for Policy and File Violations**

1. Command to verify auditd is active: **systemctl**
2. Command to set number of retained logs and maximum log file size:
   * **sudo nano /etc/audit/auditd.conf**

Add the edits made to the configuration file below:

**Max\_log file = 35**

**Num\_log = 7**



1. Command using auditd to set rules for /etc/shadow, /etc/passwd and /var/log/auth.log:

**sudo nano /etc/audit/rules.d/audit.rules**

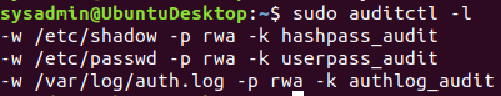
4. Add the edits made to the rules file below:

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

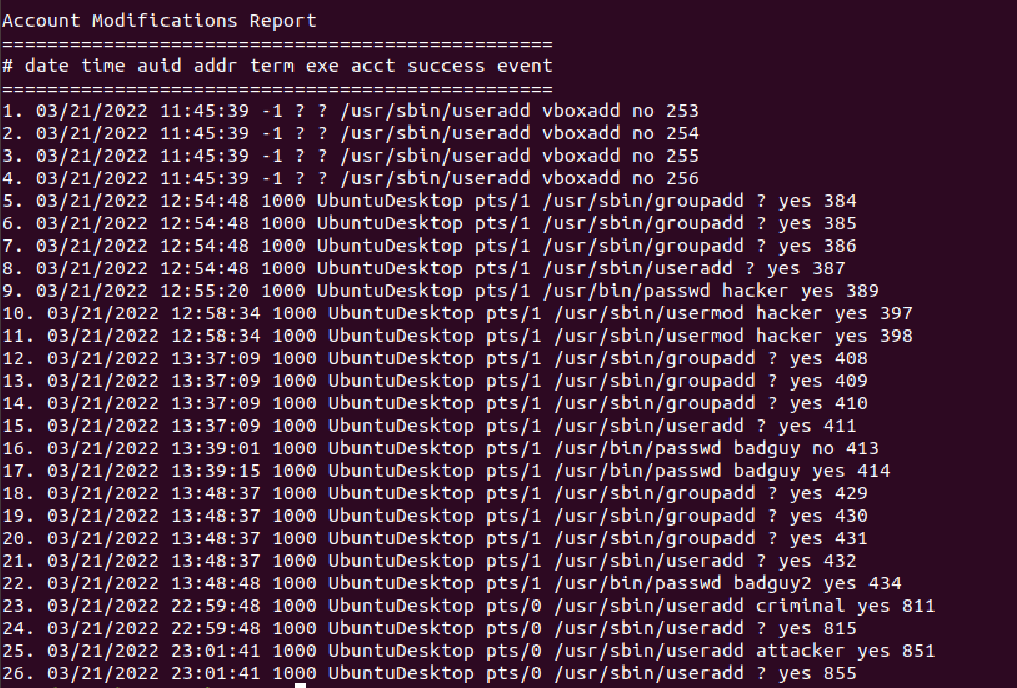
**-w /etc/passwd -p wra -k userpass\_audit**

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

1. Command to restart auditd:**sudo systemctl restart auditd**
2. Command to list all auditd rules:**sudo auditctl -l**

****

1. Command to produce an audit report:**sudo aureport -au**
2. Create a user with sudo useradd attacker and produce an audit report that lists account modifications:



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:

**journalctl -p "emerg".."err"**

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

**journalctl --disk-usage -b -0**

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:

**journalctl -p between 0..2**  /**home/sysadmin/Priority\_High.txt**

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

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.