### **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**

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

**tar -cvvf Javaless\_Docs.tar --exclude=’TarDocs/Documents/Java’ TarDocs/**

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 -cvzf logs\_backup\_tar.gz /var/log**

#### **Critical Analysis Question**

* Why wouldn't you use the options -x and -c at the same with tar?
* **-x means instructs tar to extract the files from the zipped file**
* **-c means creates a new archive**
* **Not possible to extract something that has not been created, would create an error**

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

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

**crontab -e**

**0 6 \* \* \*/3 tar -zcf /auth\_backup.tgz /var/log/auth.log**

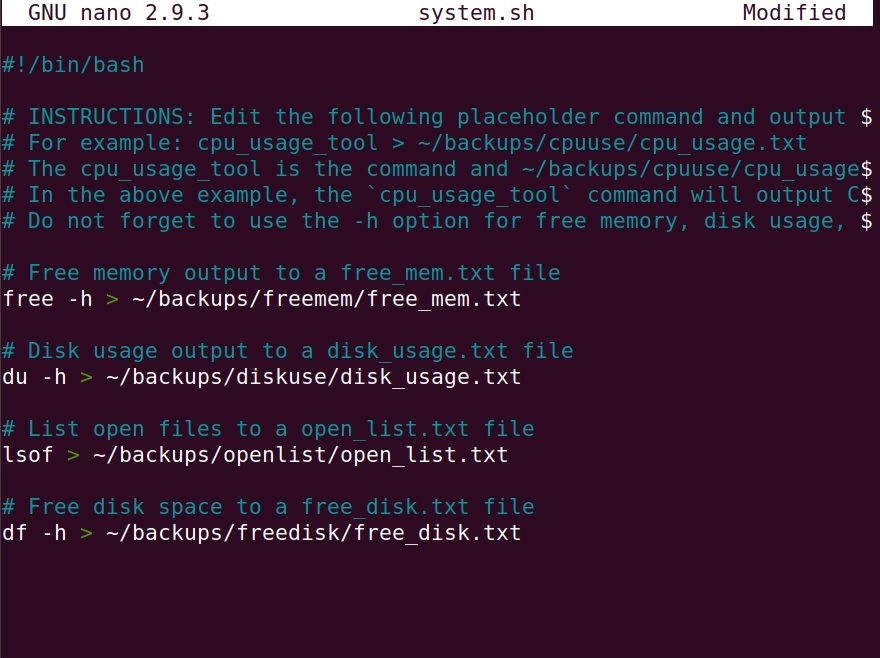
### 

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

1. Brace expansion command to create the four subdirectories:

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

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

1. [Your solution script contents here]  
   
2. Command to make the system.sh script executable:  
   **chmod +x system.sh**

**Optional**

* Commands to test the script and confirm its execution: **./ sh system.sh**
* **cat ~/backups/freemem/free\_mem.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:

**sudo nano /etc/logrotate.conf**

1. [Your logrotate scheme edits here]

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

**weekly**

**rotate 7**

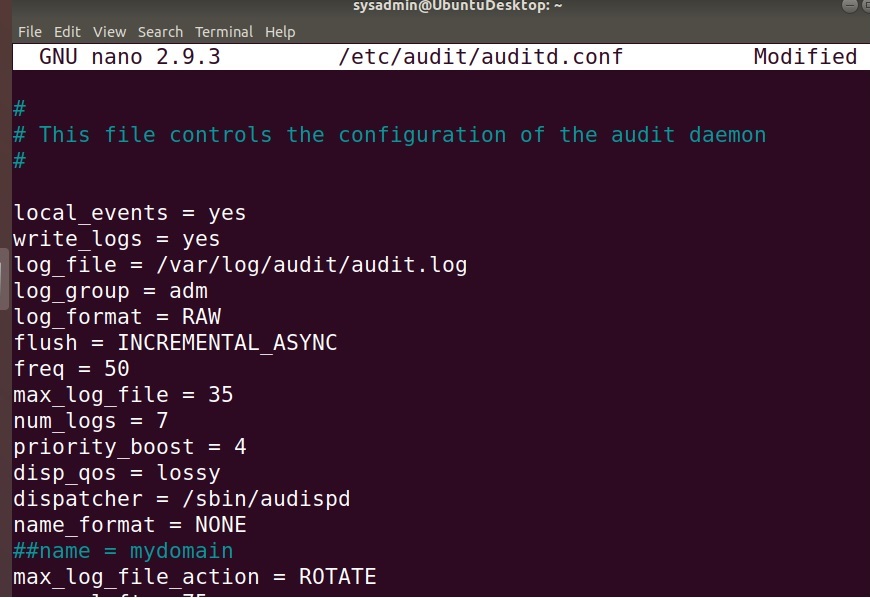
**notifempty**

**delaycompress**

**missingok**

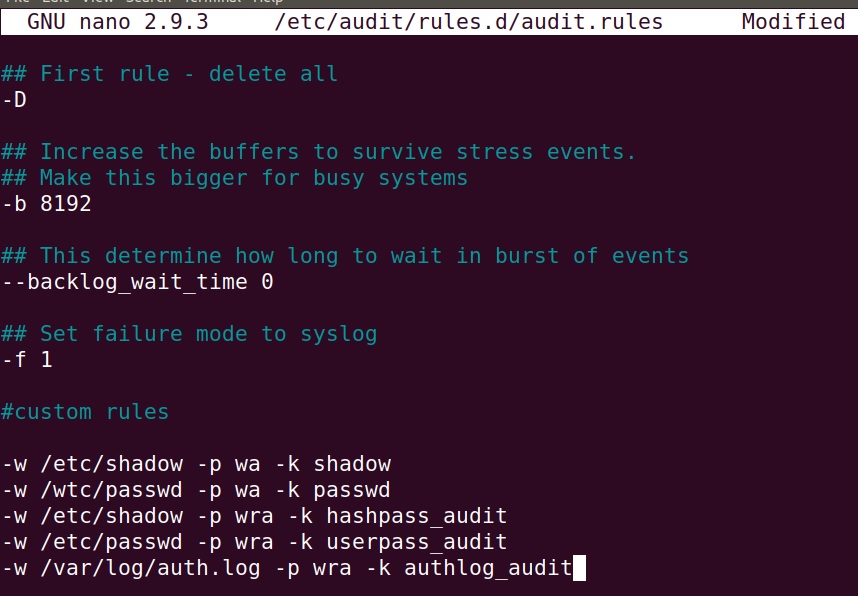
**}**

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

1. Command to verify auditd is active:  
   **systemctl status** **auditd**
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:
3. [Your solution edits here]
4. Command using auditd to set rules for /etc/shadow, /etc/passwd and /var/log/auth.log:

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

* + Add the edits made to the rules file below:

****

1. [Your solution edits here]   
   **Please see screenshot above**
2. Command to restart auditd:  
   **sudo systemctl restart auditd**
3. Command to list all auditd rules:  
   **sudo auditctl -l**
4. Command to produce an audit report:  
   **sudo aureport -au**
5. Create a user with sudo useradd attacker and produce an audit report that lists account modifications: **sudo adduser attacker**

**sudo aureport -au**

1. Command to use auditd to watch /var/log/cron:  
   **sudo auditctl -w /var/log/cron**
2. 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**
2. Command to check the disk usage of the system journal unit since the most recent boot: **sudo journalctl --disk-usage --boot=-0**
3. Command to remove all archived journal files except the most recent two: **sudo journalctl --vaccum-file=2**
4. Command to filter all log messages with priority levels between zero and two, and save output to /home/sysadmin/Priority\_High.txt: **sudo journalctl -p 0..2 > /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: **sudo crontab -e**  
     
    [Your solution cron edits here]

