**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 xvf TarDocs.Tar -C ~/Projects/

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

tar cvf Javaless\_Docs.tar –exclude=TarDocs/Documents/Java Tar Docs/

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:

tar cvvz logs\_backup.tar.gz –listed-incremental=snapshot.snar –level=0 /var/log

**Critical Analysis Question**

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

The reason why you wouldn’t put -x and -c at the same time with tar is because -x extracts the file while -c creates them. Extracting and creating a file is not possible to run at the same time.

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

1. Cron job for backing up the /var/log/auth.log file:  
   \* 6 \* \* 3 tar zvf /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}

1. 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 +x system.sh

**Optional**

* Commands to test the script and confirm its execution:

Test the script:

sudo ./system.sh

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

● cat ~/backups/diskuse/disk\_usage.txt

● cat ~/backups/openlist/open\_list.txt

● cat ~/backups/freedisk/free\_disk.txt

Confirm its execution:

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

cat > ~/backups/diskuse/disk\_usage.txt

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

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

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:

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 {  
 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:

sudo nano /etc/audit/auditd.conf

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

-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:  
   systemctl restart auditd
2. Command to list all auditd rules:  
   sudo auditctl -l
3. Command to produce an audit report:  
   sudo aureport -au
4. Create a user with sudo useradd attacker and produce an audit report that lists account modifications:  
   sudo aureport -m
5. Command to use auditd to watch /var/log/cron:  
    **sudo auditctl -w /var/log/cron**
6. 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 -ef
2. Command to check the disk usage of the system journal unit since the most recent boot:  
   journalctl –disk-usage
3. Command to remove all archived journal files except the most recent two:  
   sudo journalctl –vacuum-files=2
4. Command to filter all log messages with priority levels between zero and two, and save output to /home/sysadmin/Priority\_High.txt:  
   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: (I decided to make it every day at 7am to automate the job)

Sudo nano crontab -e

1. **0 7 \* \* \*** journalctl -p 0..2 > home/sysadmin/Priority\_High.txt