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

**sudo 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 -cvvzf logs\_backup.tar.gz /var/logs**

**Critical Analysis Question**

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

**Because if the archive hasn’t been created, there’s no files to be able to extract from it. Both of these commands contradict each other and cannot be used at the same time.**

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

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

**0 6 \* \* 3 tar -czf /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:
2. #!/bin/bash

**#!/bin/bash**

**# INSTRUCTIONS: Edit the following placeholder command and output filepaths**

**# For example: cpu\_usage\_tool > ~/backups/cpuuse/cpu\_usage.txt**

**# The cpu\_usage\_tool is the command and ~/backups/cpuuse/cpu\_usage.txt is the filepath**

**# In the above example, the `cpu\_usage\_tool` command will output CPU usage information $**

**# Do not forget to use the -h option for free memory, disk usage, and free disk space**

**# 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 -h > ~/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:

**sudo chmod +x system.sh**

**Optional**

* Commands to test the script and confirm its execution:

**Bonus**

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

**# see "man logrotate" for details**

**# rotate log files weekly**

**weekly**

**# use the syslog group by default, since this is the owning group**

**# of /var/log/syslog.**

**su root syslog**

**# keep 4 weeks worth of backlogs**

**rotate 7**

**# create new (empty) log files after rotating old ones**

**create**

**# uncomment this if you want your log files compressed**

**#compress**

**# packages drop log rotation information into this directory**

**include /etc/logrotate.d**

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

**rotate 7**

**weekly**

**notifempty**

**delaycompress**

**missingok**

**}**

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

1. Command to verify auditd is active: **sudo** **systemctl status auditd**
2. Command to set number of retained logs and maximum log file size:
   * Add the edits made to the configuration file below:

**#**

**# This file controls the configuration of the audit daemon**

**#**

**local\_events = yes**

**write\_logs = yes**

**log\_file = /var/log/audit/audit.log**

**log\_group = adm**

**log\_format = RAW**

**flush = INCREMENTAL\_ASYNC**

**freq = 50**

**max\_log\_file = 35**

**num\_logs = 7**

**priority\_boost = 4**

**disp\_qos = lossy**

**dispatcher = /sbin/audispd**

**name\_format = NONE**

**##name = mydomain**

**max\_log\_file\_action = ROTATE**

**space\_left = 75**

**space\_left\_action = SYSLOG**

**verify\_email = yes**

**action\_mail\_acct = root**

**admin\_space\_left = 50**

**admin\_space\_left\_action = SUSPEND**

**disk\_full\_action = SUSPEND**

**disk\_error\_action = SUSPEND**

**use\_libwrap = yes**

**##tcp\_listen\_port = 60**

**tcp\_listen\_queue = 5**

**tcp\_max\_per\_addr = 1**

**##tcp\_client\_ports = 1024-65535**

**tcp\_client\_max\_idle = 0**

**enable\_krb5 = no**

**krb5\_principal = auditd**

**use\_libwrap = yes**

**##tcp\_listen\_port = 60**

**tcp\_listen\_queue = 5**

**tcp\_max\_per\_addr = 1**

**##tcp\_client\_ports = 1024-65535**

**tcp\_client\_max\_idle = 0**

**enable\_krb5 = no**

**krb5\_principal = auditd**

**##krb5\_key\_file = /etc/audit/audit.key**

**distribute\_network = no**

[Your solution edits here]

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:

**## First rule - delete all**

**-D**

**## Increase the buffers to survive stress events.**

**## Make this bigger for busy systems**

**-b 8192**

**## This determine how long to wait in burst of events**

**--backlog\_wait\_time 0**

**## Set failure mode to syslog**

**-f 1**

**#auditd rules**

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

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

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

[Your solution edits here]

1. Command to restart auditd: **sudo systemctl restart auditd**
2. Command to list all auditd rules: **man auditd**

**--help**

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

**sudo aureport -m**

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

**-w /var/log/cron -p war -k cron\_file**

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:
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]