| Cybersecurity |
| --- |
| Module 5 Challenge Submission File |

## Archiving and Logging Data

Make a copy of this document to work in, and then for each step, add the solution command below the prompt. Save and submit this completed file as your Challenge deliverable.

### 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 ~/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 ~/Projects |
| --- |

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

| tar -tvf Javaless\_Docs.tar | grep "Documents/Java" |
| --- |

#### Bonus

1. 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 -cvzf logs\_backup.tar.gz --listed-incremental=snapshot.file /var/log |
| --- |

#### Critical Analysis Question

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

-c is to Create the file  
-x is to eXtract the file  
  
If you use them both at the same time, you’re not really doing anything. You’re creating a .tar and then extracting it at the same time. There is no real reason then to have created the .tar in the first place.

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

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

| crontab -e  \* 6 \* \* 3 tar -cvzf /auth\_backup.tgz /var/log/auth.log |
| --- |

### 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:

| #!/bin/bash  ## Print Free Memory and save to file  free -h > ~/backups/freemem/free\_mem.txt    ## Print Disk Usage and save to file  du -ach > ~/backups/diskuse/disk\_usage.txt  ## https://opensource.com/article/22/7/check-disk-usage-linux  ## Lists all open files and save to file  lsof > ~/backups/openlist/open\_list.txt  ## https://www.thegeekstuff.com/2012/08/lsof-command-examples/  ## Print File System Disk Space Stats and saves to file  df -h > ~/backups/freedisk/free\_disk.txt  ## https://docs.oracle.com/cd/E19683-01/806-4074/6jd68pq2n/index.html |
| --- |

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

| chmod +x system.sh |
| --- |

#### Optional

1. Commands to test the script and confirm its execution:

| sudo ./system.sh cat ~/backups/freemem/free\_mem.txt |
| --- |

#### Bonus

1. Command to copy system to system-wide cron directory:

| sudo cp ~/system.sh /etc/chron.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.

* 1. Add your config file edits:

| ## Weekly, Rotates on 7th log, No Empty, Delay Compression, Skips error for missing logs and continues  ## https://linux.die.net/man/8/logrotate  /var/log/auth.log {  weekly  rotate 7  notifempty  delaycompress  missingok  endscript  } |
| --- |

### 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 mousepad /etc/audit/auditd.conf  NOTE: We are using nano in class and nano is used in the StudentGuide, but I am more comfortable using mousepad as my text editor than nano. As such, my answers will show mousepad instead of nano. |
| --- |

Add the edits made to the configuration file:

| max\_log\_file = 35  num\_logs = 7 |
| --- |

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

| sudo mousepad /etc/audit/rules.d/audit.rules |
| --- |

Add the edits made to the rules file below:

| ## Setting permissions to watch and monitor with a keyname  -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 |
| --- |

1. Command to list all auditd rules:

| sudo auditctl -l |
| --- |

1. Command to produce an audit report:

| sudo aureport -au |
| --- |

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

| sudo useradd attacker  sudo aureport -m |
| --- |

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 -p emerg..err -b |
| --- |

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:

| sudo journalctl -p 0..2 > /home/sysadmin/Priority\_High.txt |
| --- |

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

| sudo mousepad Priority\_High.sh  #!/bin/bash  ## Filter all logs with Priority Levels between 0 and 2, and save them to a file  ## Going to append to the file so that it does not overwrite it daily, but instead adds to it  journalctl -p 0..2 >> /home/sysadmin/Priority\_High.txt  sudo chmod +x Priority\_High.sh  sudo mv ~/PriorityHigh.sh /etc/cron.daily  sudo crontab -e  ## Daily run to filer all logs for Priority Levels 0 through 2  @daily Priority\_High.sh |
| --- |

© 2022 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.