| 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:

| Sudo tar xvf TarDocs |
| --- |

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

| Sudo tar cvf Javaless\_Docs.tar --exclude=’java/\*’ Documents |
| --- |

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

| Sudo tar -f Javaless\_Docs.tar --delete 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 --level=0 /var/log |
| --- |

#### Critical Analysis Question

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

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

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

| Step 1: cp -vp /var/log/auth.log /var/log/auth\_backup.tgz/  Step 2: sudo tar cvvWf auth\_backup.tar auth.log  Step 3: sudo tar cvzf auth\_backup.tgz auth\_backup.tar |
| --- |

### Step 3: Write Basic Bash Scripts

1. Brace expansion command to create the four subdirectories:

| Mkdir backups/{freeman,diskuse,openlist,freedisk} |
| --- |

1. Paste your system.sh script edits:

| #!/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 into a `cpu\_usage.txt` file.  # 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/freemen/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 > ~/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:

| Chmod +x system.sh |
| --- |

#### Optional

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

| Sudo ./system.sh  Ls backups/freemen/  Ls backups/diskuse/  Ls backups/openlist/  Ls backups/freedisk/ |
| --- |

#### Bonus

1. 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.

* 1. Add your config file edits:

| /var/log/btmp {  Missingok  Monthly  Create 0660 root utmp  Rotate 1  /var/log/auth.log {  Missingok  Weekly  Notifempty  Rotate 7  Compress  delaycompress |
| --- |

### Bonus: Check for Policy and File Violations

1. Command to verify `auditd` is active:

| Systemctl status auditd.service |
| --- |

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:

| 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 |
| --- |

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 |
| --- |

Add the edits made to the rules file below:

| -D  -b 8192  --backlog\_wait\_time 0  -f 1  -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.service |
| --- |

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 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:

| Journalctl -p emerg..err -b -0 |
| --- |

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

| Sudo journalctl -u systemd-journald -b -0 | less |
| --- |

1. Command to remove all archived journal files except the most recent two:

| Sudo journalctl --vacuum-time=”4 days” |
| --- |

1. Command to filter all log messages with priority levels between zero and two, and save output to /home/sysadmin/Priority\_High.txt:

| Sudo cat /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 nano journal\_filter\_priority\_0\_2.sh |
| --- |

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