## **Week 5 Homework: Archiving and Logging Data**

This unit's homework is designed to solidify the following concepts and tools:

* Create a tar archive that excludes a directory using the --exclude= command option.
* Manage backups using cron jobs.
* Write bash scripts to create system resource usage reports.
* Perform log filtering using journalctl.
* Perform priority-based log filtering and log file creation using rsyslog.
* Manage log file sizes using logrotate.
* Create an auditing system to check for policy and file violations using auditd.

Please refer to the student guides and slides from this unit's lessons as you work through the assignment. If you get stuck, remember you can use Google and man pages for more information.

### **Scenario**

For this assignment, you will play the role of a security analyst for Credico Inc., a financial institution that offers checking, savings, and investment banking services.

* The company collects, processes, and maintains a large database of private financial information for both consumer and business accounts.
* The data is maintained on a local server.
* The company must comply with the Federal Trade Commission's Gramm-Leach-Bliley Act ([GLBA](https://www.ftc.gov/tips-advice/business-center/privacy-and-security/gramm-leach-bliley-act)), which requires that financial institutions explain their information-sharing practices to their customers and protect sensitive data.

In an effort to mitigate network attacks and meet federal compliance, Credico Inc. developed an efficient log management program that performs:

* Priority-based logging with rsyslog.
* Log size management using logrotate.
* Log auditing with auditd to track events, record the events, detect abuse or unauthorized activity, and create custom reports.

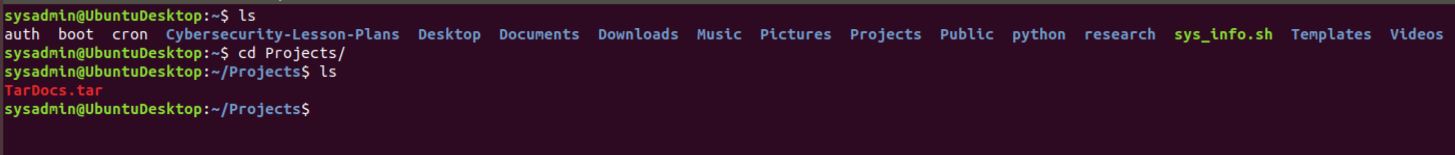
These tools, in addition to archives, backups, scripting, and task automation, contribute to a fully comprehensive log management system.

You will expand and enhance this log management system by learning new tools, adding advanced features, and researching additional concepts.

### **Lab Environment**

To set up your lab environment with the necessary files, complete the following steps.

* Log into your local virtual machine. Use the following credentials:  
  + Username: sysadmin
  + Password: cybersecurity
* Open the terminal within your Ubuntu VM by pressing Ctrl+Alt+T for Windows users or Ctrl+Options+T for Mac users.  
  + Alternatively, press Windows+A (Command+A for Mac users), type "Terminal" in the search bar, and select the terminal icon (not the Xfce Terminal icon).
* Create a directory called Projects in your /home/sysadmin/ directory.
* Download the following file (you can either slack it to yourself or use the Firefox browser in your Ubuntu machine), and move it to your ~/Projects directory before you get started:  
  + [TarDocs.tar](https://drive.google.com/a/2tor.com/file/d/1fRjFS1vOdS7yfKJgpJxR02_UxeT_qI_u/view?usp=sharing)



### **Instructions**

As you solve each step below, please fill out the Submission File. This will be your homework deliverable.

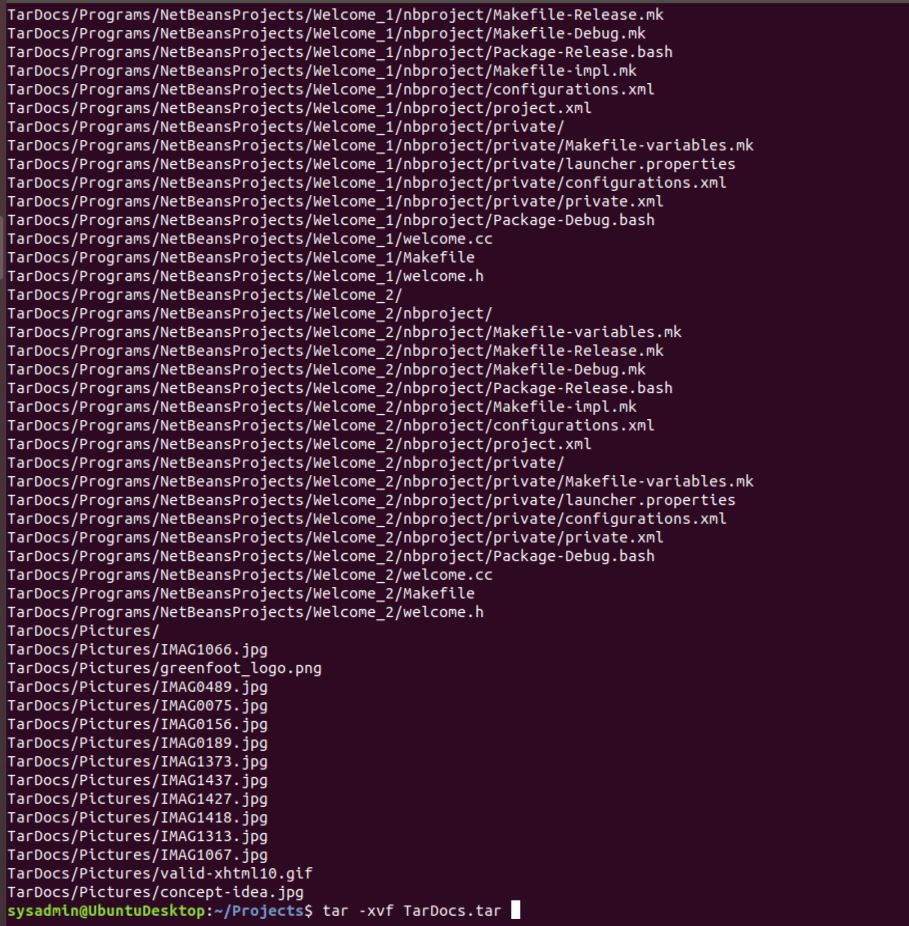
In each of the following sections, you will use and build on your system administration tools and knowledge. Make sure to read the instructions carefully.

#### **Step 1: Create, Extract, Compress, and Manage tar Backup Archives**

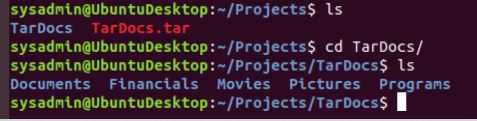
Creating tar archives is something you must do everyday in your role at Credico Inc. In this section, you will extract and exclude specific files and directories to help speed up your workflow.

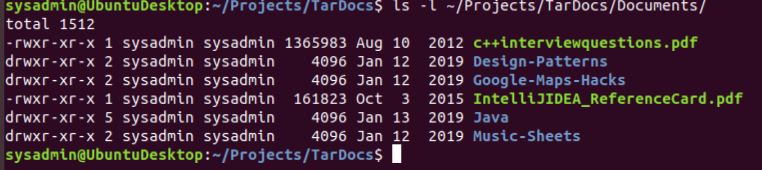
To get started, navigate to the ~/Projects directory where your downloaded TarDocs.tar archive file should be.

1. Extract the TarDocs.tar archive file into the current directory (~/Projects). Afterwards, list the directory's contents with ls to verify that you have extracted the archive properly.  
   * Note that because we want to preserve the directory structure of our archive, we do not have to specify a target directory to extract to.



* + Note that when you run ls you should see a new ~/Projects/TarDocs directory with five new subdirectories under TarDocs/.

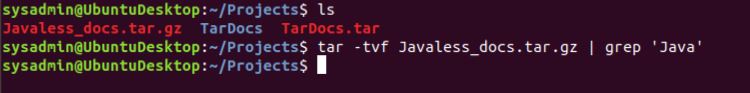


Verify that there is a Java subdirectory in the TarDocs/Documents folder by running: ls -l ~/Projects/TarDocs/Documents/.

1. Create a tar archive called Javaless\_Docs.tar that excludes the Java directory from the newly extracted TarDocs/Document/ directory.
   * If you've executed this command properly, you should have a Javaless\_Docs.tar archive in the ~/Projects folder.



1. Verify that this new Javaless\_Docs.tar archive does not contain the Java subdirectory by using tar to list the contents of Javaless\_Docs.tar and then piping grep to search for Java.



**Bonus**

* Create an incremental archive called logs\_backup.tar.gz that contains only changed files by examining the snapshot.file for the /var/log directory. You will need sudo for this command.

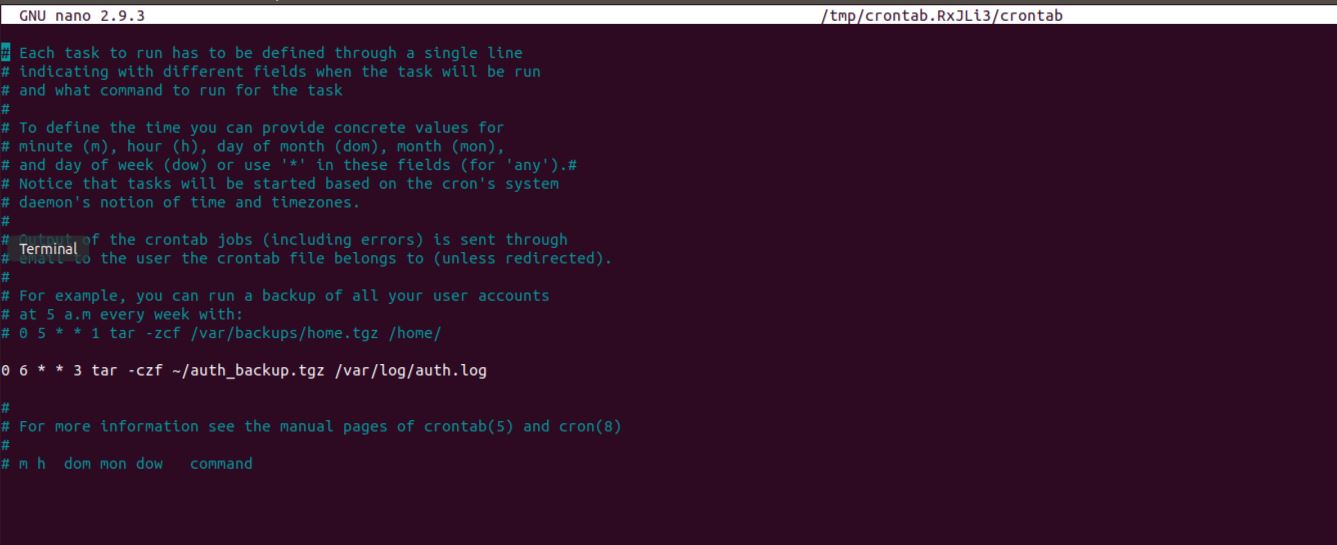
#### **Step 2: Create, Manage, and Automate Cron Jobs**

In response to a ransomware attack, you have been tasked with creating an archiving and backup scheme to mitigate against CryptoLocker malware. This attack would encrypt the entire server’s hard disk and can only be unlocked using a 256-bit digital key after a Bitcoin payment is delivered.

For this task, you'll need to create an archiving cron job using the following specifications:

* This cron job should create an archive of the following file: /var/log/auth.log.
* The filename and location of the archive should be: /auth\_backup.tgz.
* The archiving process should be scheduled to run every Wednesday at 6 a.m.
* Use the correct archiving zip option to compress the archive using gzip.

1. To get started creating cron jobs, run the command crontab -e. Make sure that your cron job line includes the following:  
   * The schedule (minute, hour, etc.) in cron format. - **Hint:** Reference the helpful site [crontab.guru](https://crontab.guru/) as needed.
   * An archive (tar) command with three options.
   * The path to save the archive to
   * The path of the file to archive.



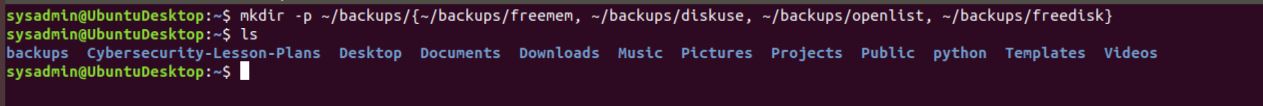
#### **Step 3: Write Basic Bash Scripts**

Portions of the Gramm-Leach-Bliley Act require organizations to maintain a regular backup regimen for the safe and secure storage of financial data.

You'll first need to set up multiple backup directories. Each directory will be dedicated to housing text files that you will create with different kinds of system information.

For example, the directory freemem will be used to store **free memory** system information files.

1. Using brace expansion, create the following four directories:  
   * ~/backups/freemem
   * ~/backups/diskuse
   * ~/backups/openlist
   * ~/backups/freedisk
2. **Note**: Remember that brace expansion uses the following format: ~/exampledirectory/{subdirectory1,subdirectory2,etc}



Now you will create a script that will execute various Linux tools to parse information about the system. Each of these tools should output results to a text file inside its respective system information directory.

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

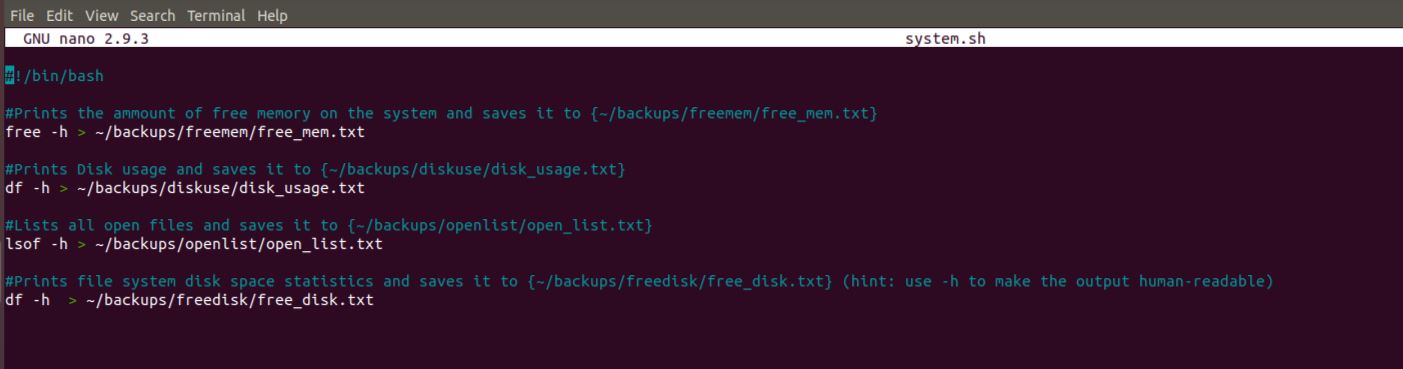
In the above example, the cpu\_usage\_tool command will output CPU usage information into a cpu\_usage.txt file.

To get started with setting up your script up in your home directory, do the following:

* Navigate to your home directory by running: cd ~/
* Run the command nano system.sh to open a new Nano window.

**Note**: If you're unsure how to get started, we included a system.sh starter file. Use that as a guide.

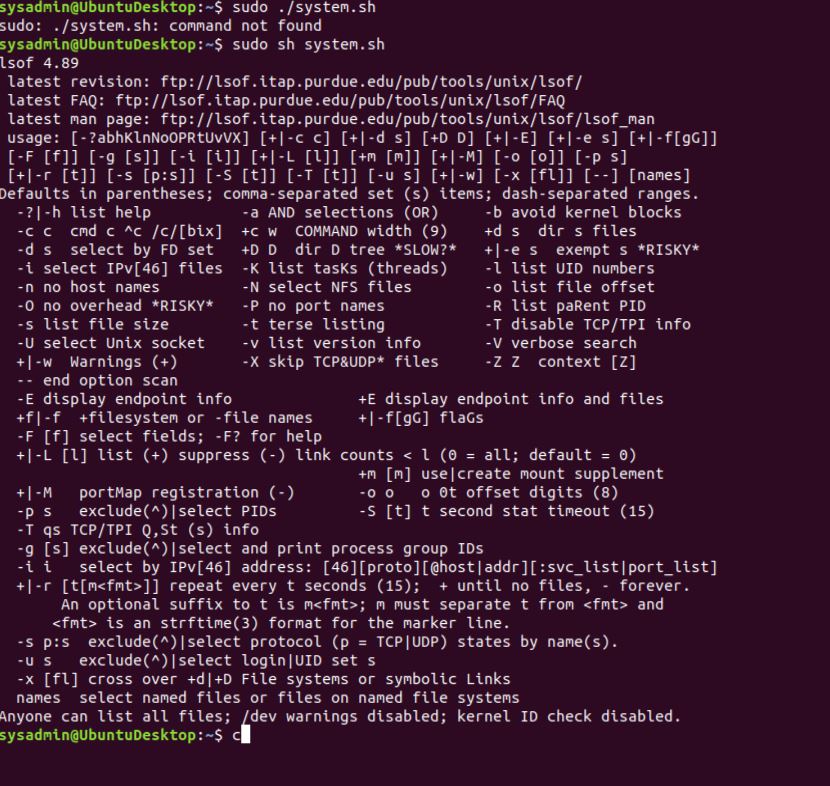
1. Edit the system.sh script file so that it that does the following:  
   * Prints the amount of free memory on the system and saves it to ~/backups/freemem/free\_mem.txt.
   * Prints disk usage and saves it to ~/backups/diskuse/disk\_usage.txt.
   * Lists all open files and saves it to ~/backups/openlist/open\_list.txt.
   * Prints file system disk space statistics and saves it to ~/backups/freedisk/free\_disk.txt.



1. **Note**: For the free memory, disk usage, and free disk commands, make sure you use the -h option to make the output human-readable.
2. Save this file and make sure to change or modify the system.sh file permissions so that it is executable.

You should now have an executable system.sh file within your home ~/ directory.

* Test the script with sudo ./system.sh.
* **Note**: If it appears, ignore the warning: lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1001/gvfs Output information may be incomplete.



**Optional**

* Confirm the script ran properly by navigating to any of subdirectories in the ~/backup/ directory and using cat <filename> to view the contents of the backup files.

**Bonus**

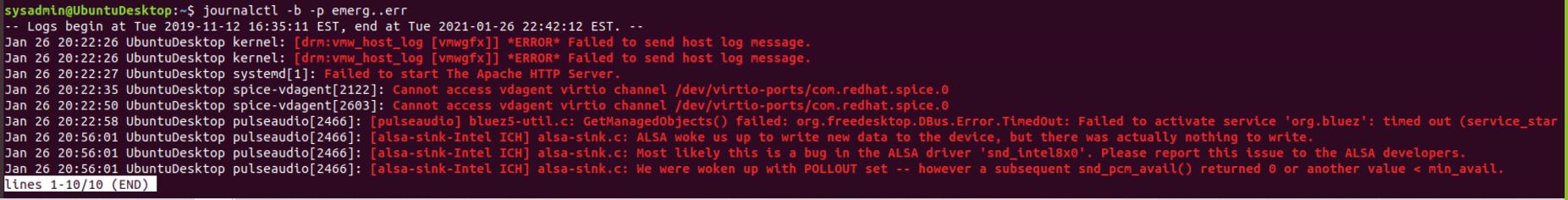
* Automate your script system.sh by adding it to the weekly system-wide cron directory.

#### **Step 4: Perform Various Log Filtering Techniques**

There was a suspicious login from a host on the network during the early morning hours when the office was closed. The senior security manager tasked you with filtering through log files to determine if a system breach occurred.

**Hint:** Remember that journal tracks each log relative to each system boot. Also, keep in mind that you can sort messages by priority, relative boot, and specific units.

1. Use journalctl to perform a log search that returns all messages, with priorities from emergency to error, since the current system boot.



1. Use journalctl to check the disk usage of the system journal unit since the most recent boot. You will likely have to pipe this output to less if it doesn't fit on the screen.
   * The unit you want is systemd-journald.

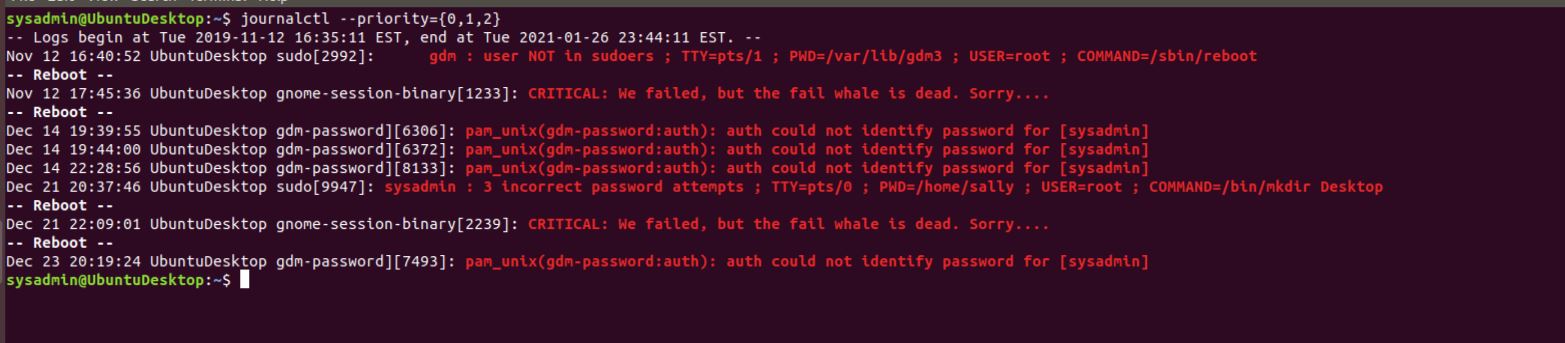


1. Use journalctl to remove all archived journal files except the most recent two.



**Bonus**

* Use journalctl to filter all log messages with priority levels between zero and two, and save the results to a file named Priority\_High.txt in /home/student/ directory.



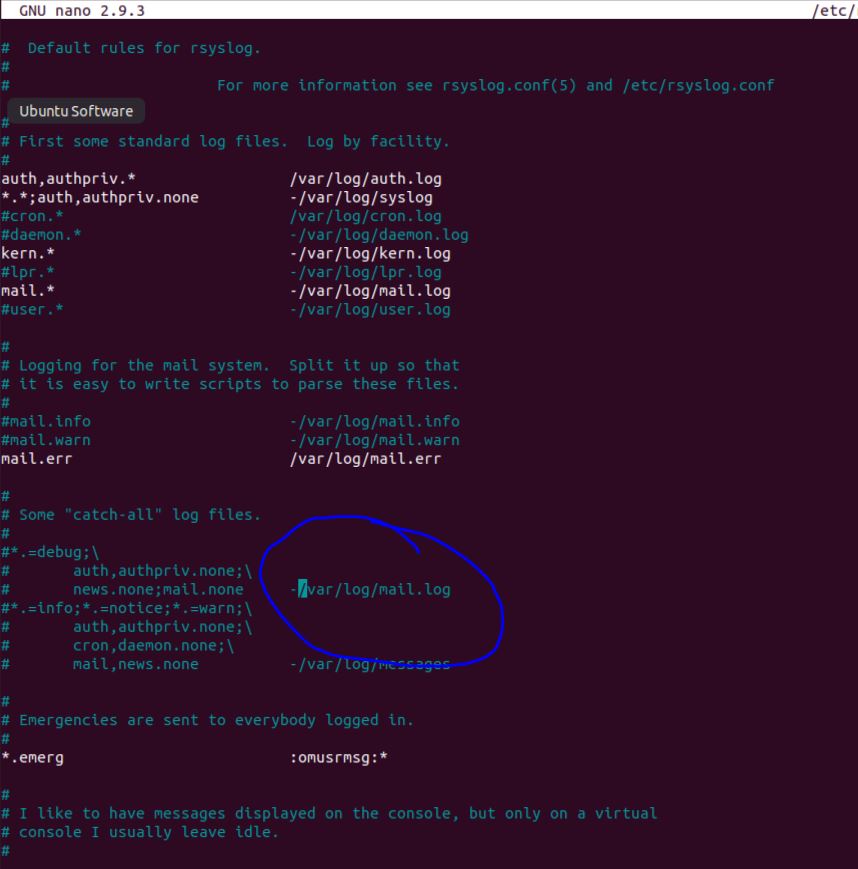
* Automate the last task by creating a cron job that runs daily in the root crontab.  
    
   **Note**: You'll need sudo to run journalctl.

#### **Step 5. Create Priority-Based Log Files**

Your organization is constantly bombarded with spam messages, a type of social engineering attack. To address this, you’ve decided to implement a priority-based log filtering system to monitor access to the mail daemon.

To get started with editing rsyslog configurations from any directory within your terminal, run sudo nano /etc/rsyslog.d/50-default.conf to edit the rsyslog config file.

1. Configure rsyslog to record mail log message for all priorities except debug to the /var/log/mail.log directory.



**Bonus**

* Configure rsyslog to record boot log messages for all priorities except info and debug to the /var/log/boot.log directory.

#### **Step 6. Manage Log File Sizes**

You realize that the spam messages are making the size of the log files unmanageable.

You’ve decided to implement log rotation in order to preserve log entries and keep log file sizes more manageable. You’ve also chosen to compress logs during rotation to preserve disk space and lower costs.

1. Run sudo nano /etc/logrotate.conf to edit the logrotate config file. You don't need to work out of any specific directory as you are using the full configuration file path.
2. Configure a log rotation scheme that backs up authentication messages to the /var/log/auth.log directory using the following settings:  
   * Rotates weekly.
   * Rotates only the seven most recent logs.
   * Does not rotate empty logs.
   * Delays compression.
   * Skips error messages for missing logs and continues to next log.
3. Don't forget to surround your rotation rules with curly braces {}.

