Exploring Unix Logs on CentOS

Estimated completion time: 45-60 min.

Warning: In the commands given in this instruction, the difference between the number one ('1') and the lower-case letter ell ('l') can be very slight, if anything. The context for the commands should tell you what it ought to be.

Heads up: A simple list of Unix commands is given in an Appendix at the end of the document.

I. Objective

The objective of this exercise is to give the student some hands-on experience with syslog configuration and testing.

II. Getting Started

1. **Boot your Linux system or VM. If necessary, log in and then open a terminal window and cd to the labtainer/labtainer-student directory. The pre-packaged Labtainer VM will start with such a terminal open for you. Then start the lab:**

    labtainer centos-log2

    Note the original terminal displays the paths to two files on your Linux host:

    1) This lab manual

    2) The lab report template

    On most Linux systems, these are links that you can right click on and select "Open Link". **If you chose to edit the lab report on a different system, you are responsible for copying the completed report back to the displayed path on your Linux system before using "stoplab" to stop the lab for the last time.**

2. **Log into CentOS as Joe using password4joe as the password.**

III. Explore

1. Enter the `su` command but **give the <u>wrong</u> password** for root.

2. Enter the **`su`** command again, but this time give the correct password for root, which is **badpassword**. If you did it right, the prompt should end in '#'.

3. Explore the log directory

Change the current working directory to `/var/log`.

List the contents of `/var/log`.

You should see a variety of files and directories. Note that blue names refer to directories. You **may** see a couple of different "extensions" on files:

- `.old`
  This is the "rotated" copy of the log. You should see another file with the same prefix, but without the ".old" extension.
- `-yyyymmdd`
  This is another example of a "rotated" log but with a more useful extension: the date it was rotated. If you see this extension, then you should also see another file with the same prefix, but without the dated extension.

Look at the permissions for the `messages` log.

Record in Item #1 of your report the permission(s) that regular users have with respect to this file.

Most of the files in the log directory are text-based, but there are exceptions to the usual Unix rule. Many will be empty, either because it was recently rotated, or because the associated service is not running, or no auditable event associated with the service has been detected yet.

4. Wrong Password

Login-related records are stored in the text file named `secure`. The most recent records are written at the end of the file.

Open the file and look for your **failed** attempt to **log in** with the username of `Joe` (**not** your failure to 'su' to root).

In Item #2 of your report, record the wording that is used to indicate a failed login attempt.

Note that Item #3 asks a follow-up question.

5. The `su` action.

   With the `secure` log still open, find the entry at the bottom of the file related to your earlier action to **su** to root. Look at the kind of information that is stored about an `su` action.

   In Item #5 of your report, record the information that was recorded about your recent `su` action.

   Exit the editor when you are done looking around.

6. `wtmp` file

   One of the binary files in the log directory is the commonly found `wtmp` file, which requires the use of other tools to extract information from it, such as the `last` command.

   Bring up the man page for the `last` command, by doing the following:

   ```
   man last
   ```

   Read the DESCRIPTION section of the man page to find out what the command does.

   Go to the OPTIONS section of the man page.

   In Item #6 of your report, write in your own words the functionality provided by the `-t` option of `last`.

IV. Reconfigure rsyslog for MARK

In this section you will turn on the MARK feature and restart syslog to accept the change.

1. Open the rsyslog configuration file.

   While still running with root privilege in the terminal, start an editor from the command line (such as `leafpad`) to open `/etc/rsyslog.conf`

   Remember: when the rsyslog daemon reads this file during initialization, anything after a '#' (through the end of the line) is treated as a comment.

2. Enable the Mark feature.

   By default, the insertion of timestamps at a specified frequency is disabled.

   - In the "### MODULES ###" section, find the line that has **$ModLoad immark** and remove the leading '#' to enable this feature.
   - Set the frequency of the timestamps by adding the following line **under** the one you just changed:

     ```
     $MarkMessagePeriod  60
     ```

     Where "60" is the number of seconds between timestamps[1].
   - Save the change and exit the editor.

3. Restart the rsyslog daemon.

   Restarting the rsyslog daemon will cause it to reinitialize and re-read its configuration file (thus making any change effective). Do the following to perform the restart:

   ```
   systemctl restart rsyslog
   ```

4. See the effect of this change in the logs by using the `tail` command in the following fashion:

   ```
   tail –f  /var/log/messages
   ```

   The `tail` command shows the last several lines of a file (as opposed to `head`, which shows the first several lines of a file). The "`-f`" option tells it to wait "forever" and display more lines as they are added at the end of the file.

   You should see a line recording the fact that rsyslogd was stopped, and then a line recording that rsyslogd was started.

   Continue to wait until you see a MARK record appear in the log. After you have seen it (or more than a minute passes), press **Ctl-C** to exit `tail`.

V. Reconfigure and Test rsyslog

In this subsection you will become familiar with the `logger` utility for manually creating syslog entries. A system administrator could use this command to document changes he/she makes to the system, and it can be used to test changes to the syslog configuration. You will make some changes to the syslog rules and then use `logger` to test those changes.

---

1 This setting is very subjective. If you don't put this line in, the default period is 20 minutes. This short interval of 60 seconds works well for our purposes in this lab.

1. Read the DESCRIPTION section of the `man` page for the `logger` utility:

   ```
   man logger
   ```

2. Generate a record in `/var/log/messages` with a priority of "info" by doing the following:

   ```
   logger  -p info  "Hello World"
   ```

   When no facility is specified, as is the case with the above command, the "user" facility is used by default.

3. Reopen the rsyslog configuration file at `/etc/rsyslog.conf`, and scroll down to the "#### RULES ####" section. It might be helpful to expand the window size so nothing wraps around.

   In Item #7 of your report, write the syslog rule that specifies what to do with the record you sent to syslog in step #2 above.

4. Exit the editor.

5. Use `grep` (or some other tool of your choice) to verify that your log entry has been saved in the file you think it should be saved in (per the rule you recorded in item #7 of the report). [If it is not there, then you probably made a mistake. In that case, feel free to reevaluate the rule you chose until you get it right.]

6. Reopen the syslog configuration file and scroll down to the RULES section.

   Add a new syslog rule that puts <u>all</u> messages with a priority of "debug" into a file called `/var/log/mydebug`. The file should <u>only</u> have debug messages. Feel free to refer to the lecture slides and your lecture notes to figure out what to do.

   In Item #8 of your report, write the rule you used to meet the above requirement.

7. Save your changes to the configuration file and then exit the editor.

8. Restart rsyslog again (so your new rule will take effect):
   ```
   systemctl restart rsyslog
   ```
   If your change to `rsyslog.conf` has a syntax error, then it will be reported at the end of `/var/log/messages`.

9. Now that you know how to use `logger`, use it to test the rule you added to `rsyslog.conf` in step #6 above.

   In Item #9 of your report, describe how you used `logger` (and other commands) to test the rule you added in step #6.

10. Do the following to display the permissions associated with the `logger` command:

```
ll /bin/logger
```

It may not be a good idea to let regular users execute `logger`. Change the permissions so that only the root user and the root group can execute it.

In Item #10 of your report, write the command(s) you used to change the permissions on `logger`.

VI. Centralized Logging

Imagine you have several Linux systems to manage. Instead of configuring and reviewing logging on each of the systems, you can define a centralized logger and then forward log messages from each of the systems to that centralized logger. In this section, you will configure your existing "logger" system to accept log messages from remote computers, and you will configure a workstation computer to forward its logs to the logger.

1. Reopen /etc/rsyslog.conf on the logger computer.
2. Find the these entries in the configuration file and uncomment them (remove the "#") to allow accept syslog messages on port 514 via either TCP or UDP:

```
$ModLoad imudp
$UDPServerRun 514

$ModLoad imtcp
$InputTCPServerRun 514
```

3. Then restart your rsyslog on the logger.

```
systemctl restart rsyslog
```

4. On the Linux host (where your issued the "labtainer centos-log2" command) type:

```
moreterm.py centos-log2 workstation
```

5. That will start a new virtual terminal that is connected to a workstation computer. This computer shares a network with your logger. Use "ifconfig" on each computer to view the IP address of each.'
6. On your logger, use "tail" to view your logs:
7. `tail -f /var/log/*`
8. Use "`sudo su`" to elevate privilges on the workstation.
9. Open the /etc/rsyslog.conf file on the workstation and find its "RULES" section. At the end of that section, add this line to direct all messages to your logger:

```
*.* @172.25.0.2
```

10. Now restart rsyslog on the workstation and observe the log messages on your logger.
11. Experiment with different security relevant events such as de-elevating and elevating privileges on the workstation and issuing `logger` commands from the workstation.

VII. More Questions

1. Reopen `/etc/rsyslog.conf` on the logger.

   Referring to `/etc/rsyslog.conf`, answer the questions in items 11 through 13 of your report.

2. Perform extra experimentation if you have not already done so. **Describe your actions and your results in item #14 of your report**.

3. In item #15 of your report, please describe what you learned from this exercise, if anything.

4. In item #16 of your report, describe any recommendations you may have for improving this exercise.

5. After finishing the lab, go to the terminal on your Linux system that was used to start the lab and type:
   ```
   stoplab
   ```

If you modified the lab report on a different system, you must copy that completed file into the directory path displayed when you started the lab, and you must do that before typing “stoplab”. When you stop the lab, the system will display a path to the zipped lab results on your Linux system.

VIII. Deliverable

Provide the zip file to your instructor, e.g., via the Sakai site.

Appendix – Some Unix Commands

`cd` Change directory

```
cd  location
```

With no "location", you will be taken to your home directory.

`chmod` Change the DAC permissions on a file or directory.

```
chmod permissions objectname
```

Consult Lab 1 for examples.

`clear` Erase all the output on the current terminal and place the shell prompt at the top of the terminal.

`grep` Search for a string in the given input.

```
grep string filename
```

The above command will search for "string" within the given file. If the string has spaces in it, then it must be quoted.

`ls` List the contents and/or attributes of a directory or file

```
ls  location
ls  file
```

With no "location" or "file" it will display the contents of the current working directory.

`less` Display a page of a text file at a time in the terminal

```
more file
```

To see another page press the space bar. To see one more line press the Enter key. To quit before reaching the end of the file enter 'q'.

`man` Manual

```
man command
```

Displays the manual page for the given "command". To see another page press the space bar. To see one more line press the Enter key. To quit before reaching the end of the file enter 'q'.

`more` Display a page of a text file at a time in the terminal

```
more file
```

To see another page press the space bar. To see one more line press the Enter key. To quit before reaching the end of the file enter 'q'.

`pwd` Display the present working directory

```
pwd
```

`su` Super user (change to root)

`tail` Display the last 20 lines of a text file on the terminal.

`touch` Change the modification date on the given file. If the file does not exist, it will be created.

```
touch filename
```

If no time is provided, then the modification time will be change to the current time. Given the right options, `touch` can also be used to change the modification to a

specific date/time.