Exemplar: Filter with grep

Activity overview

Previously, you learned about tools that you can use to filter information in Linux. You’re also familiar with the basic commands to navigate the Linux file system by now.

In this lab activity, you’ll use the grep command and piping to search for files and to return specific information from files.

As a security analyst, it’s key to know how to find the information you need. The ability to search for specific strings can help you locate what you need more efficiently.

Scenario

In this scenario, you need to obtain information contained in server log and user data files. You also need to find files with specific names.

Here’s how you’ll do this: **First**, you’ll navigate to the logs directory and return the error messages in the server_logs.txt file. **Next**, you’ll navigate to the users directory and search for files that contain a specific string in their names. **Finally**, you’ll search for information contained in user files.

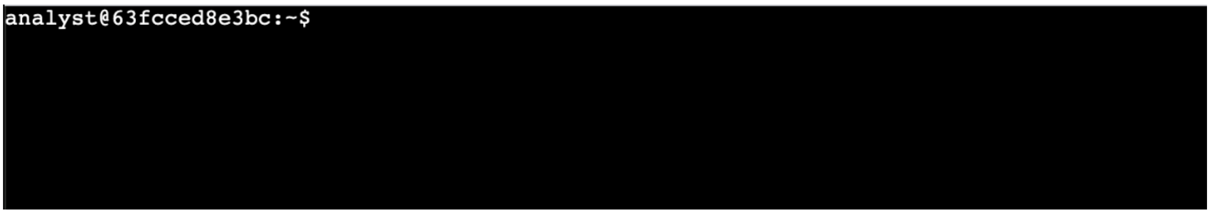
With that in mind, you’re ready to practice what you've learned.

Note:** The lab starts with your user account, called analyst, already logged in to a Bash shell. This means you can start with the tasks as soon as you click the **Start Lab** button.Disclaimer:** For optimal performance and compatibility, it is recommended to use either **Google Chrome** or **Mozilla Firefox** browsers while accessing the labs.

Start your lab

You'll need to start the lab before you can access the materials. To do this, click the green “Start Lab” button at the top of the screen.

After you click the **Start Lab** button, you will see a shell, where you will be performing further steps in the lab. You should have a shell like this:



```
analyst@63fcced8e3bc:~$
```

When you have completed all the tasks, refer to the End your Lab section that follows the tasks for information on how to end your lab.

Task 1. Search for error messages in a log file

In this task, you must navigate to the /home/analyst/logs directory and report on the error messages in the server_logs.txt file. You'll do this by using grep to search the file and output only the entries that are for errors.

1. Navigate to the /home/analyst/logs directory.

The command to complete this step:

```
cd logs
```

2. Use grep to filter the server_logs.txt file, and return all lines containing the text string error.

Note: *If you enter a command incorrectly and it fails to return to the command-line prompt, you can press **CTRL+C** to stop the process and force the shell to return to the command-line prompt.*

The command to complete this step:

```
grep error server_logs.txt
```

This grep command will filter server_logs.txt file, and return a list of the lines that match the text string error.

Note: *The first argument passed to grep is the string you're searching for, and the second argument is the name of the file you're searching through.*

How many error lines are there in the server_logs.txt file?

Two

Eight

Three

Six

Submit

Answer: There are six entries in the server_logs.txt file that include the error string.

Click **Check my progress** to verify that you have completed this task correctly.

Search for error messages in a log file

Check my progress

Task 2. Find files containing specific strings

In this task, you must navigate to the /home/analyst/reports/users directory and use the correct Linux commands and arguments to search for user data files that contain a specific string in their names.

1. Navigate to the /home/analyst/reports/users directory.

The command to complete this step:

cd /home/analyst/reports/users

2. Using the pipe character (|), pipe the output of the ls command to the grep command to list only the files containing the string Q1 in their names.

The command to complete this step:

ls | grep Q1

How many files in the /home/analyst/reports/users subdirectory contain “Q1” in their names?

Two

Five

Three

One

Submit

Answer: There are three files in the reports/users directory that have Q1 in their names.

Note: *Piping sends the standard output of one command to the standard input of another command for further processing. In the example, the output of the grep command is piped to the ls command and the output displayed in the shell.*

3. List the files that contain the word access in their names.

The command to complete this step:

```
ls | grep access
```

How many files in the /home/analyst/reports/users directory contain “access” in their names?

Three

Four

None

Five

Submit

Answer: There are four files in the reports/users directory that have the text string access in their names.

Click **Check my progress** to verify that you have completed this task correctly.

Find files containing specific strings

Check my progress

Task 3. Search more file contents

In this task, you must search for information contained in user files and report on users that were added and deleted from the system.

1. Display the files in the /home/analyst/reports/users directory.

The command to complete this step:

```
ls
```

2. Search the Q2_deleted_users.txt file for the username jhill.

The command to complete this step:

```
grep jhill Q2_deleted_users.txt
```

Did you find the username jhill in the Q2_deleted_users.txt file?

No

Yes

Submit

Answer: Yes, the user jhill is listed in the Q2_deleted_users.txt file.

3. Search the Q4_added_users.txt file to list the users who were added to the Human Resources department.

The command to complete this step:

```
grep "Human Resources" Q4_added_users.txt
```

Note: *In order for grep to interpret a string of two or more words correctly, you must enclose it in quotes ("Human Resources").*

How many users were added to the Human Resources department in quarter 4?

One

Five

Two

Three

Submit

Answer: Two new users were added to the Human Resources department in quarter 4.

Click **Check my progress** to verify that you have completed this task correctly.

Search more file contents

Check my progress

Conclusion

Great work!

You now have practical experience in using grep to:

- search for specific information contained in files, and
- find files containing specific strings that were piped into grep.

You’re well on your way to using fundamental tools in Linux to filter the information you need.

End your lab

Before you end the lab, make sure you’re satisfied that you’ve completed all the tasks, and follow these steps:

1. Click **End Lab**. A pop-up box will appear. Click **Submit** to confirm that you're done. Ending the lab will remove your access to the Bash shell. You won’t be able to access the work you've completed in it again.
2. Another pop-up box will ask you to rate the lab and provide feedback comments. You can complete this if you choose to.
3. Close the browser tab containing the lab to return to your course.
4. Refresh the browser tab for the course to mark the lab as complete.