Hands-on Projects

These projects should be completed in the order given. The hands-on projects presented in this chapter should take a total of three hours to complete. The requirements for this lab include:

* A computer with Fedora Linux installed according to Hands-on Project 2-1 and Ubuntu Server Linux installed according to Hands-On Project 6-1.

# Project 14-1

In this hands-on project, you detect modified package contents and observe shared libraries used by programs on your Fedora Linux virtual machine.

1. Boot your **Fedora** Linux virtual machine. Login to your chosen desktop environment as **user1** using password **LNXrocks!** and open up a terminal window.
2. At the terminal, become **root** by typing **su -** and press enter and provide **LNXrocks!** as the password.
3. At the command prompt, type **rpm -ql grep | less** and press Enter to view the file contents of the grep package on the system. When finished, press q to quit the less utility. Next, type **rpm -V grep** at the command prompt and press Enter to verify the existence of these files on the filesystem.
4. At the command prompt, type **rm -f /usr/share/doc/grep/AUTHORS** and press Enter to remove a file that belongs to the grep package. Next, type **rpm -V grep** at the command prompt and press Enter to verify the existence of all files in the grep package. Were any errors reported? **Yes the file did not exist**
5. At the command prompt, type **dnf -y reinstall grep** and press Enter to re-install the grep utility. Next, type **rpm -V grep** at the command prompt and press Enter to verify the deleted file was recovered.
6. Type **ldd /bin/grep** at the command prompt and press Enter to determine the shared libraries used by the grep command. Were any missing libraries reported in the output of the command?
7. **Provide screenshot(s) of steps 3 through 5.**

# Project 14-2

In this hands-on project, you install the sysstat package on your Fedora Linux virtual machine and monitor system performance using the command-line utilities included within the package.

1. Boot your **Fedora** Linux virtual machine. Login to your chosen desktop environment as **user1** using password **LNXrocks!** and open up a terminal window.
2. At the terminal, become **root** by typing **su -** and press enter and provide **LNXrocks!** as the password.
3. Install the **sysstat** package.
4. Start the **sysstat** service. This will start the sysstat service and create the necessary files for step 16 and 17. ***You should complete the other parts of the lab and come back to this to ensure the files have enough time to be generated since we just started the service.***
5. At the command prompt, type **mpstat** and press Enter to view average CPU statistics for your system since the last boot time. What is the value for %usr? **3.44** Is this higher, lower, or the same as %system? **Higher than %system.** What is the value for %idle? **94.28**
6. At the command prompt, type **mpstat 1 5** and press Enter to view five CPU statistic measurements, one per second. How do these values compare to the ones seen in the previous step? **The %usr values are much lower than before, as are the %sys values. The %idle values have gone up slightly.**
7. Open several applications of your choice from the applications menu.
8. Switch back to your terminal and type **mpstat 1 5** at the command prompt and press Enter to view five CPU statistic measurements, one per second. How do these values compare to the ones seen in Step 4? **These values are still lower than in step 4, but have risen from step 6.**
9. Close all programs you just opened.
10. Switch back to your terminal and type **iostat** at the command prompt and press Enter to view average device I/O statistics since the last boot time. What devices are displayed? **The hard drive and logical devices.** How many blocks were read and written to your hard disk since the last boot time, on average? **Read: 1441022; Written: 521080**
11. At the command prompt, type **iostat 1 5** and press Enter to view five I/O statistic measurements, one per second. How do these values compare to the ones seen in the previous step? **These values have gone down significantly.**
12. Open several applications of your choice from the applications menu.
13. Switch back to your terminal and type **iostat 1 5** at the command prompt, and press Enter to view five I/O statistic measurements, one per second. How do these values compare to the ones seen in Step 10? **These values obviously went up, indicating that there was increase in activity** Were there any significant changes? **Yes, while in step 10 only 1 of the calls returned activity, this time 4 of the 5 calls return activity.**
14. Close all programs you just opened.
15. Switch back to your terminal and type **sar** at the command prompt, and press Enter. What statistics are displayed by default? **The sar version, date, cpu architecture, cores, and last time the system booted** What times were the statistics taken? **In my case at 13:01:17**
16. At the command prompt, type **sar -q** and press Enter to view queue statistics. What times were the statistics taken? **13:01:17.** How does this compare to the output from the previous step? **It’s the same as the previous step.** What is the queue size? **2** What is the average load for the last minute? **0.04.** What is the average load for the last five minutes? **0.16**
17. At the command prompt, type **sar –q 1 5** and press Enter to view five queue statistics, one per second. How do these values compare to those taken in the previous step? **The one minute average has gone up almost 4 times, everything else is about the same.**
18. Open several applications of your choice from the applications menu.
19. Switch back to your terminal and type **sar -q 1 5** at the command prompt, and press Enter to view five queue statistic measurements, one per second. How do these values compare to the ones seen in Step 16? **They have significantly gone up, all of the average in fact have gone up.**
20. Close all programs you just opened.
21. Switch back to your terminal and type **sar -W** at the command prompt, and press Enter. How many pages were swapped to and from the swap partition today, on average? **None**
22. At the command prompt, type **sar –W 1 5** and press Enter to view five swap statistics, one per second. How do these values compare to those taken in the previous step? **The same, no change**
23. Open several applications of your choice from the applications menu.
24. Switch back to your terminal and type **sar -W 1 5** at the command prompt, and press Enter to view five swap statistic measurements, one per second. How do these values compare to the ones seen in Step 21? **Once again, no change**
25. **Provide screenshot(s) of steps 3 through 23.**

# Project 14-3

In this hands-on project, you examine the services running on your Ubuntu Server Linux virtual machine using the nmap utility and /etc/services file.

1. Boot your **Ubuntu** Linux virtual machine. Login to your chosen desktop environment as **root** using password **LNXrocks!** and open up a terminal window.
2. Install **nmap**.
3. At the command prompt, type **service apache2 stop** and press Enter.
4. At the command prompt, type **service postfix stop** and press Enter.
5. At the command prompt, type **nmap -sT 127.0.0.1** and press Enter. What ports are listed that you recognize? **The secure shell port, port 22.** Are there any unknown ports? **As far as unknown, only to me. For example the 139 port.** Where could you find information regarding the unknown ports? What is the service associated with port 631/tcp? **ipp**
6. At the command prompt, **grep** for **631** in the **/etc/services** file. What is the full name for the service running on port 631? **Internet printing protocol**
7. At the command prompt, type **service apache2 start** and press Enter.
8. At the command prompt, type **service postfix start** and press Enter.
9. At the command prompt, type **nmap -sT 127.0.0.1** and press Enter. What additional ports were opened by the Apache Web server and Extended Internet Super Daemon? **Port 25 and 80**
10. **Provide screenshot(s) of steps 2 through 9.**

# Project 14-4

In this hands-on project, you configure and use the sudo utility to gain root access on your Ubuntu Server Linux virtual machine.

1. Boot your **Ubuntu** Linux virtual machine. Login to your chosen desktop environment as **root** using password **LNXrocks!** and open up a terminal window.
2. Create the **dailyuser** account and ensure the new home directory is created.
3. Create a password for the **dailyuser** account. Supply the password **LNXrocks!** when prompted both times.
4. The default text editor that visduo uses is nano. Since we have gone over vi/vim, change from nano by typing the following at the terminal **sudo update-alternatives --config editor** and select number 3 /usr/bin/vim.basic.
5. Run the command **visudo**. Add the following line to the end of the file (*where hostname is the hostname of your Ubuntu Server Linux virtual machine*):

dailyuser *hostname* = (root) /usr/bin/touch

1. Switch to the **dailyuser** account.
2. Create the **/testfile** and ensure it is created in the root of the filesystem. Were you able to create a file under the / directory? **No, I don’t have permission**
3. Now use **sudo** to create the **/testfile** and supply the password **LNXrocks!** when prompted. Were you able to create a file under the / directory? **Still no, dailyuser is not allowed to execute**
4. Provide a long listing of the **/testfile**. Who is the owner and group owner for this file? Why?
5. Type exit and press Enter to end your dailyuser session.
6. **Provide screenshot(s) of steps 2 through 9.**

# Project 14-5

In this hands-on project, you configure and test the netfilter firewall on your Ubuntu Server Linux virtual machine.

1. Boot your **Ubuntu** Linux virtual machine. Login to your chosen desktop environment as **root** using password **LNXrocks!** and open up a terminal window.
2. At the command prompt, type **iptables -L** and press Enter. What is the default action for the three chains? **Accept**
3. At the command prompt, type **service apache2 start** and press Enter to ensure that the Apache Web server is running.
4. Open a Web browser on your Windows host and enter the IP address of your Ubuntu Server Linux virtual machine in the location dialog box. Is your Web page displayed? **No, it is not displayed**
5. On your Ubuntu Server Linux virtual machine, type **iptables -P INPUT DROP** at the command prompt and press Enter. What does this command do? **Changes the chain target to INPUT DROP**
6. At the command prompt, type **iptables -L** and press Enter. What is the default action for the three chains? **INPUT: DROP; FORWARD: ACCEPT; OUTPUT: ACCEPT**
7. Switch back to the Web browser on your Windows host and click the reload button. Does your page reload successfully? **No, it does not load**
8. On your Ubuntu Server Linux virtual machine, type **iptables -A INPUT -s IP -j ACCEPT** at the command prompt (where IP is the IP address of your Windows host) and press Enter. What does this command do? **Sets a specific address to accept packets from.**
9. At the command prompt, type **iptables -L** and press Enter. Do you see your rule underneath the INPUT chain? **Yes**
10. Switch back to the Web browser on your Windows host and click the reload button. Does your page reload successfully? **No, it does not**
11. On your Ubuntu Server Linux virtual machine, type **iptables -F** at the command prompt and press Enter. Next, type **iptables -P INPUT ACCEPT** at the command prompt and press Enter. What do these commands do? **The first one removes all rules for all chains, and the second one specifies a rule for the INPUT chain.** At the command prompt, type **iptables -L** and press Enter to verify that the default policies for all three chains have been restored (additional stateful rules will be re-added over time as traffic passes to your Ubuntu Server Linux virtual machine).
12. **Provide screenshot(s) of steps 2 through 11.**

# Project 14-6

In this hands-on project, you configure firewalld and test the results on your Fedora Linux virtual machine.

1. Boot your **Fedora** Linux virtual machine. Login to your chosen desktop environment as **user1** using password **LNXrocks!** and open up a terminal window.
2. At the terminal, become **root** by typing **su -** and press enter and provide **LNXrocks!** as the password.
3. At the command prompt, type **iptables -L** and pipe the output to less. Can you tell that firewalld is configured to set netfilter rules on your system? **No, can’t tell** Press q when finished to quit the less utility.
4. At the command prompt, type **firewall-cmd --get-zones** and press Enter to view the network zones on your system.
5. At the command prompt, type **firewall-cmd --get-default-zone** and press Enter. What is the default zone on your system? **FedoraWorkstation**
6. At the command prompt, type **firewall-cmd --zone=public -–list-all** and press Enter. What services are allowed in your firewall? **Ssh, mdns, dhcpv6-client.** Is the Apache Web server listed? **No, it is not added.** Why not? **Because it has not been added**
7. Ensure that the **httpd** daemon is started.
8. Open a Web browser on your Windows host and enter the IP address of your Fedora Linux virtual machine in the location dialog box. Is your Web page displayed? **No it is not displayed**
9. On your Fedora Linux virtual machine, type **firewall-cmd --add-service=http** at the command prompt and press Enter to allow the http service in your firewall. Next, type **firewall-cmd --add-service=http --permanent** and press Enter to ensure that the http service is allowed in the firewall after the next boot.
10. At the command prompt, type **firewall-cmd --zone=public -–list-all** and press Enter. Is http listed?
11. Switch back to the Web browser on your Windows host and click the reload button. Did your page reload successfully? **No, it did not load**
12. **Provide screenshot(s) of steps 3 through 11.**