|  |
| --- |
| **Sec:** |

|  |
| --- |
| **Student-ID:** |

|  |
| --- |
| **Name:** |

**Topics: Logs, Networking Setup and Backup**

This lab exercise is to be submitted **at the end** of the lab session !

*With your server having been on the network, we want to explore some logs, and some information about users, access, etc.*

*And finally, what we all have been waiting for: There IS a GUI that we can install!*

Start your machine in Virtualbox.

Move to the directory containing the log files:

**/var/log/**

**ls**

shows you a number of log files. You can also see, that some log files contain numbers, and have the same name. This is called ***log-rotation*** and is done to make the information easier accessible, keeping file sizes smaller, while retaining older log-entries. (ls -l would show you the dates.)

Let us explore some of those files:

**1. auth.log**

This is the file that stores all logons, successful as well as unsuccessful. Let us simulate a failed logon. You can do this over the network (like in the last exercise) or locally. The latter is easier, so we do the easier one: Open another console, and try to log on with your username, but a wrong password. Then go back to your original console, and

**cat auth.log | grep failure | tail -n 1**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This log entry shows the failed login, the username being used, the time, etc. If you get many of these, someone is obviously trying to break into your system.

**2. boot.log**

At boot, the system checked all the file systems (mount points) with fsck and probably found them to be in good order ('clean'). Then it started the services.

**cat boot.log**

shows all of this.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3. mail.log**

[*This is the log file that we used earlier to check if a message was sent* *and accepted* *by the recipient’s mail server.*]

If you wanted, you could send a mail to have another entry in the log file. Have you not send any, you'll only see that the mail server has been started.

In this case, we stop and (re-)start the mail server (MTA) for the track record:

**sudo postfix stop sudo postfix start**

Note the last lines of tail mail.log:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Configure the static IP**

Ethernet interfaces are identified by the system using the naming convention of ethX, where X represents a numeric value. The first Ethernet interface is typically identified as eth0, the second as eth1, and all others should move up in numerical order.

To quickly identify all available Ethernet interfaces, you can use the ifconfig command as shown below:

**ifconfig -a | grep eth**

Currently, the Ubuntu Server installer has set your server to use Dynamic Host Control Protocol(DHCP), which means your IP address is automatically set by the DHCP server. Every server needs a static IP address to solve the DNS issue.

Changing this setting without a GUI will require some text editing, but that’s classic linux, right?

Let’s open up the /etc/network/interfaces file. I’m going to use nano, but you can choose a different editor

**sudo nano /etc/network/interfaces**

or the primary interface, which is usually eth0, you will see these lines:

**auto eth0**

**iface eth0 inet dhcp**

As you can see, it’s using DHCP right now. We are going to change dhcp to static, and then there are a number of options that should be added below it. Obviously you’d customize this to your network.

**auto eth0**

**iface eth0 inet static**

**address aaa.bbb.ccc.ddd #( This is your static IP address)**

**netmask eee.fff.ggg.hhh #( The netmask for your network)**

**network iii.jjj.kkk.lll #( Network address)**

**broadcast mmm.nnn.ooo.ppp #( Broadcast address)**

**gateway qqq.rrr.sss.ttt #( Exit gateway)**

**dns-nameservers uuu.vvv.www.xxx #( DNS server)**

Now we’ll just need to restart the networking components:

**sudo /etc/init.d/networking restart**

**Backup**

Almost every Linux Admin use this utility (for backing up their critical data) which is something call tar, tar allows you easily and quickly backup your files folder or your entire system. Next, you need to schedule backup task using cron job. Backing up your files using tar is very simple using following command:

**cd /**

**mkdir mybackup**

**tar -cvpzf /mybackup/etcbackup.tar.gz /etc**

Now Let’s add tar command in bash script to make this whole backup process automatic. To make this script automatic and run in background we will use cron job. Create file using nano editor and paste below script.

|  |
| --- |
| **#!/bin/bashTIME=`date +%b-%d-%y`          # This Command will add date in Backup File Name.FILENAME=backup-$TIME.tar.gz   # Here i define Backup file name format.SRCDIR=/etc     # Location of Important Data Directory (Source of backup).DESDIR=/mybackup           # Destination of backup file.tar -cpzf $DESDIR/$FILENAME $SRCDIR#END OF SCRIPT** |

Next, open cronab editor utility. Cron jobs allow you to schedule your tasks and run automatically.

**sudo crontab –e**

It has 6 parts see below explanation:



If you want to run the backup for every Monday and Friday at 11.00pm, please add this line at the file crontab

00 23 \* \* 1,5 /bin/bash/scriptfile

**Graphical User Interface (GUI)**

We find a good reason to finally install a GUI: the use of a web browser. A browser usually is a graphic application, so we need to get some graphics up in our system.

The graphics in \*nix is based on the X Windows System of MIT (Massachusetts Institute of Technology).

We start this interface:

**startx**

Ooops. The applications are not yet installed.

Follow the suggestions as lined out in the error message.

Then try again. Graphics should start now, the mouse become visible, and a terminal come up. We are not yet done, though. There is no handling of the frames around the window(s), no background, the window(s) can't be moved, resized, etc. We need a so-called *Windows Manager* to help us manage the windows.

Typing 'exit' in that terminal gets us back to the console.

We use the most light-weight windows manager: *fluxbox*.

Install fluxbox.

startx

brings up the UI now, with fluxbox as windows manager.

With the right mouse clicks you can explore and change the settings of fluxbox.

Next, we want to use a web browser: Chrome.

(Right-click) Applications → Terminal Emulators → Xterm opens a graphical terminal,

where we can install Google's Chrome browser. It is named chromium-browser here. Install this application.

When the installation is finished, you can start Chrome:

chromium-browser

Select a search engine of your preference. Voilà, there it is.

*You cannot yet browse the web, because of the proxy settings of Uniten. :(*

*You would need to start it with* chromium-browser –proxy-server=http://172.20.23.211:5865"

We use the browser to access one of the web-servers of other students. You need to know the IP-address of someone else's virtual machine, again, and you need to know your own IP-address, again.

With a right-click on the desktop of fluxbox (you might want to minimize the Chrome window eventually) and Applications → Terminal Emulators → Xterm you can open a second terminal and find out the IP-address of your server (refer to Lab 9 if in doubt). Note the IP-address of your server here:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now enter the IP-address of your neighbour's machine in the address bar of your browser.

You should see that person's welcome message, that he / she had modified in the previous exercise.

Also, try to open a page that does not exist; for example by simply appending the page name nonsense.html to the URL:

http://172.20.16.43/nonsense.html [Leave the IP-address untouched!]

You receive a notice "The requested URL /nonsense.html was not found on this server"

In that second terminal (you can't use the first one, because you started Chrome in there)

move to the directory that contains the log files of your web server:

**cd /var/log/apache2**

There, you find *access* and *error* logs; access shows all accesses, successful or not. error shows all erroneous accesses; for the example the nonsense.html – page should show up here.

The logs are pretty exhaustive, they tell you from which address your server was queried, also about the web browser software, version, capabilities, etc. Note the **access log** for the nonsense.html page here:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

And last not least: the **error log** for this page (nonsense.html):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Right-click on the fluxbox-desktop and 'Exit' shuts down the GUI, and gets you back to the console, from where you can shut down your server as always.

The configuration file for the proxy is in that same directory: /etc/apt/apt.conf. Which is the current setting for the proxy? (Use cat /etc/apt/apt.conf)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Don't forget to submit your lab sheet!*

*This was the last lab, thanks very much for participating until today!*

*I can only hope that you enjoyed this lab; and even more, I hope that you could learn something useful about computers, operating systems, and system administration.*

*If you think that one or another thing could be improved, just let me know, please, so that I can consider your input for the next semester.*

*Thanks again, and good luck for the finals.*

*(Actually, hard work will give you a better chance than betting on good luck* **;)** *)*