**Student Name: Benjamin Lee Student ID: 13248113 Tutorial: 1-04 9:00-11:00 Friday Tutor Name: Yingying Yang Due Date: 10/9/21**

Benjamin Lee 13248113

31338 Network Servers

Learning Journal Part 1

**Table of Contents**

[General Description 2](#_Toc82155638)

[Intent 2](#_Toc82155639)

[Week 1 2](#_Toc82155640)

[Lecture 1 2](#_Toc82155641)

[Lab 1a 2](#_Toc82155642)

[Lab 1b 3](#_Toc82155643)

[Lab 1c 5](#_Toc82155644)

[Week 2 7](#_Toc82155645)

[Lecture 2 7](#_Toc82155646)

[Lab 2a 7](#_Toc82155647)

[Lab 2b 9](#_Toc82155648)

[Lab 2c 10](#_Toc82155649)

[Lab 2d 12](#_Toc82155650)

[Week 3 13](#_Toc82155651)

[Lecture 3 13](#_Toc82155652)

[Lab 3a 13](#_Toc82155653)

[Lab 3b 18](#_Toc82155654)

[Week 4 19](#_Toc82155655)

[Lecture 4 19](#_Toc82155656)

[Lab 4a 19](#_Toc82155657)

[Lab 4b 20](#_Toc82155658)

[Lab 4c 23](#_Toc82155659)

[Week 5 29](#_Toc82155660)

[Lecture 5 29](#_Toc82155661)

[Lab 5a 29](#_Toc82155662)

[Lab 5b 35](#_Toc82155663)

[Reflection 38](#_Toc82155664)

# General Description

In this learning journal, my aim is to note-take my journey to develop a solid understand of networking and improve my skills to manage network servers in both CentOS/Linux and Windows. The purpose of this journal is to keep a log/record of all my system administration tasks, efforts and methods when learning this subject.

# Intent

My intention in this subject is to learn the new methods of administrating these operating systems and update my configuration skills. I had some issues with VMware Fusion on my MAC due to some networking issues so I had to use my old PC for the VM lab work instead. Using the learning materials were very rewarding and I learnt a lot from doing the labs and figuring out what to do by myself. Supplementary materials of the lecture slides were also helpful while I was progressing.

# Week 1

Week 1 Topic: System Start-up

## Lecture 1

The lecture discusses POST, linux kernel, BIOS and UEFI differences, GRUB2 and the boot process as well as log files.

### Lab 1a

**Task 1: Navigate through the VMware Lab**

In this task, since we are in lockdown we are downloading the two VM’s ‘centos64’ and ‘Windows Server’ online. Due to some technical issues on my MAC using VMware Fusion, where ethernet1 is disabled and I could not enable it even when I tried configuring a new network adapter, I installed the two VM’s on my old PC via ***VMware workstation 16*** instead and it seems to be working fine after this.

In this task, I also changed my centos vmx configuration file to add the following line: ***mks.win32.useInjectedMagic = “FALSE”***

**The Configuration of my VM for centos:**

* 2GB of Memory, 2 Processors, 20GB Hard Disk, NAT network adapter and VMNet2 network adapter

**The Configuration of my VM for Windows Server:**

* 4GB of Memory, 2 Processors, 60GB Hard Disk, NAT network adapter and VMNet2 network adapter

**Task 2: Logging in as root and shutting down your Linux image**

We are already superuser as root so we don’t need to do the command ‘**su**’.

**Ifconfig command** shows ens33 and ens37, providing the configurations of the network interface. The **ls** **command** shows the contents of a directory.

**shutdown -h now** - It is important to know this command as in the real world, a system administrator would use this command to power off a linux server or machine in a safe and secure manner.

### Lab 1b

**Task 1: Boot single-user shell only, using boot loader**

In this task, we want to edit the GRUB2 boot loader to enter the single user mode. To do this we can use the following steps:

1. Boot up the centos machine
2. Wait till the grub menu screen appears
3. The grub menu should show multiple kernels and a timer. Once the timer ends which should be around 5 seconds, the kernel highlighted will be booted. We should get something like this:

Text

Description automatically generated

1. We want to edit the centos linux kernel (1st option) so press e to edit
2. Replace ro with the line: rw init=/sysroot/bin/sh
3. After editing the line, press **Ctrl-X** to boot machine with new parameters

Remember that single-user mode is usually for maintenance and can be used for password recovery of the root password if lost or forgotten.

**Task 2: Explore and modify system start-up scripts**

In task 2 we learn about targets or run levels.

We can use the command **systemctl get-default** to verify the current default target:Text

Description automatically generated

As seen above, the output shows *graphical.target* is the default. Using **runlevel**, we can see *graphical.target* is in **level 5** [Full Multi-User Graphical Mode]. However, we can edit this via the **systemctl isolate** command. There are four main targets we can change to:

* **emergency.target** – used for emergency system recovery
* **rescue.target** – Used for system recovery but requires root password
* **multi-user.target** – multi-user mode with no graphical login
* **graphical.target** – full graphical mode

From my lab results changing the others yield:

* **systemctl isolate multi-user.target** – Changes runlevel to 5 and 3. Full multi-user graphical mode and full multi-user text mode.
* **systemctl isolate rescue.target –** Changes runlevel to 3 and 1. Full multi-user text mode and single-user mode.
* **systemctl isolate emergency.target –** Changes runlevel to N and N. I am guessing this means ‘Not used’ but I am not sure.

**systemctl list-unit-files** command shows targets and services known as units. Some examples include cups.path [**Enabled**], anaconda.service [**static**], arp-ethers.service [**disabled**] and many more.

SSHD is enabled by default and httpd is disabled by default by using the command **syststemctl is-enabled** as seen below. We can also check if a service is active (started) or inactive (stopped) via the command **systemctl is-active.**

Text

Description automatically generated

Text

Description automatically generated

We can also use these commands to interact and stop/start services:

* systemctl start <servicename>
* systemctl stop <servicename>
* systemctl enable <servicename>
* systemctl disable <servicename>

For example, after disabling and stopping httpd by using some of these commands we should get the following:

**[root@localhost ~]# systemctl stop httpd**

**[root@localhost ~]# systemctl disable httpd**

**[root@localhost ~]# systemctl is-enabled httpd**

disabled

**[root@localhost ~]# systemctl is-active httpd**

inactive

**Task 3: Examine system log information**

The **dmesg** command shows network protocols and driver information in the kernel ring buffer. Using **journalctl --dmesg** shows kernel message logs from boot.

The file **/var/log/messages** shows general system message logs.

The file **/var/log/secure** logs information on successful/unsuccessful login attempts and authentication failures. Tracks SSH logins, sudo logins and much more. This could be useful for tracking activity of valid users as well as successful logins.

The commands below are usually used for filtering logs to look for specific information such as time, service and log type.

**journalctl --since “10 minutes ago”**

**journalctl --since “2020-01-01” --until “yesterday”**

**journalctl -u multi-user.target**

**journalctl -u sshd.service**

**journalctl -u sshd.service --since “10 minutes ago”**

**journalctl -p err**

**journalctl -p warning**

**journalctl -p err --since “1 hour ago”**

Text

Description automatically generatedText

Description automatically generated

### Lab 1c

**Task 1: Startup**

Changed the hostname in Windows using Server Manager to benlee as you can see below after reboot. Also checked time zone, updates and network settings which seemed fine on my PC. When I did this on my MAC via VMware Fusion I couldn’t connect Ethernet1, and DHCP couldn’t be enabled no matter what I tried. I switched to my old PC and installed my VM’s on there and that fixed the problem.

Graphical user interface, text, application

Description automatically generated

**Task 2: Server Management**

Device manager can be found in **Tools** à **Computer Management** à **Device Manager**. We can find various things in here such as Monitors, graphics cards, ports, processors, disk drives, network adapters and much more. This is different to linux as windows has the ability to show hardware. Linux cannot do that.

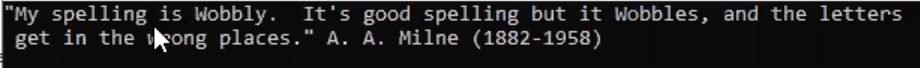
**IP Address Windows: 192.168.192.129**

**IP Address Linux: 192.168.192.128**

**Gateway is 192.168.192.2 and the subnet mask is 255.255.255.0**

To add Telnet Client and Simple TCP/IP Services: **Server Manager** à **Manage** à **Add Roles and Features** 🡪 **Skip Installation Type, Server selection, Server roles 🡪 In features tick Simple TCP/IP Services and Telnet 🡪 Install.**

**Make sure Simple TCIP/IP Services is enabled by going to Tools 🡪 Services 🡪 Enabling Simple TCP/IP Services. Check Status is ‘running’.**

Typing **telnet localhost 13** shows time and date and if telnet is working, **telnet localhost 17** shows some random quotes.

I tried to telnet and ping to the Windows virtual machine but it didn’t work. Was a firewall issue, after disabling the firewall I was able to telnet and ping.

Text

Description automatically generated

**Task 3: Command line**

**net start** command provides a list of started services such as DHCP Client, DNS Client, COM+ Event System etc.

Using **net stop “Simple TCP/IP Services”** I got the following output:

The Simple TCP/IP Services service is stopping.

The Simple TCP/IP Services service was stopped successfully.

After stopping the service telnet doesn’t work anymore as the service has been stopped.

Adding the one rule: netsh advfirewall firewall add rule name="TCP Port 17" dir=inaction=allow protocol=TCP localport=17, and I was able to telnet from my host station.

**Task 4: Documentation**

We can disable IE Enhanced Security Configuration by going to **Server Manager** à **Local Server** 🡪 **IE Enhanced Security Configuration**

Graphical user interface, text, application, email

Description automatically generated

It should be on for both administrators and users. We can disable the Security configuration by clicking ‘Off’ for both administrators and users.

# Week 2

Week 2 topic: System Documentation, Hardware and kernel configuration System installation and management

## Lecture 2

This lecture discusses documentation for windows and linux, kernel modules, hardware configuration, understanding filesystems, software packages and contains useful information on viewing processes

### Lab 2a

**Task 1: Using and configuring man pages**

Entering **mandb -cqs &** in terminal we get this:

**[root@benjamin ~]# mandb -cqs &**

[1] 3018

Running **man 1 passwd** and **man 5 passwd** we can see the difference is that:

* **man 1 passwd –** Shows 1st section of **passwd** man pages: ‘User Utilities’

Text

Description automatically generated

* **man 5 passwd –** Shows 5th section of **passwd** man pages: ‘Password file’ and is part of the Linux Programmer’s Manual'

Text

Description automatically generated

**man grep** and **info grep** are similar as they both display man pages of **grep**, however they both have there own differences.

* **man grep** displays how grep can be used with a description and various options, outputs and prefixes.
* **info grep** displays interactive man pages, showing how to use grep and includes a **menu** of different sections that the user can navigate towards on various commands.

**whatis passwd**

**[root@benjamin ~]# whatis passwd**

openssl-passwd (1ssl) - compute password hashes

passwd (1) - update user's authentication tokens

passwd (5) - password file

**apropos passwd**

**[root@benjamin ~]# apropos passwd**

chgpasswd (8) - update group passwords in batch mode

chpasswd (8) - update passwords in batch mode

fgetpwent\_r (3) - get passwd file entry reentrantly

getpwent\_r (3) - get passwd file entry reentrantly

gpasswd (1) - administer /etc/group and /etc/gshadow

grub2-mkpasswd-pbkdf2 (1) - Generate a PBKDF2 password hash.

htpasswd (1) - Manage user files for basic authentication

ldappasswd (1) - change the password of an LDAP entry

lpasswd (1) - Change group or user password

openssl-passwd (1ssl) - compute password hashes

* **whatis** displays ONLY *man page descriptions*
* **apropos** ***returns more results*** as it provides and searches the *names and descriptions* in man pages

**Task 2: Finding installed package documentation**

* Going into directory /usr/share/doc/sed\*/ we can view the contents of the FAQ file via the command zless sedfaq.txt.gz

Text

Description automatically generated

**Task 3: Finding installed package documentation**

* HowtoDocument Linux Clock: <http://tldp.org/HOWTO/Clock.html>
* Main Linux FAQ: <https://tldp.org/HOWTO/META-FAQ.html>
* Linux Sys Admin Guide: <https://tldp.org/LDP/sag/html/index.html>

### Lab 2b

**Task 1: Enable Linux networking**

To enable the NetworkManager service use command **systemctl enable NetworkManager**

To start the service use command **systemctl start NetworkManager**

You can check if NetworkManager service is active via the command **systemctl is-active NetworkManager. It should state – ‘active’**

Pinging uts.edu.au shows that the network is working

Text

Description automatically generated

**Task 2: Updates for Linux**

We can use the command **gnome-software** to start the Gnome Software Manager. All my packages seem up to date. Updates of packages can also be fetched and installed using yum. XXXXX can be a package name of our choice such as ‘dhcp-server’ (yum install dhcp-server)

**yum check-update**

**yum search *XXXXX***

**yum install *XXXXX***

**yum update *XXXXX***

**yum remove *XXXXX***

**Task 3: Updates for Windows**

My results from using the commands in powershell:

* **Get-WindowsUpdate** – Shows 5 updates to be installed

Text

Description automatically generated

* **Get-WULastResults** – Shows last time updates were checked. Also shows Installation success date: 22/7/20 12:04am and last search success date: 8/9/21 7:51pm
* **Get-WUHistory** – Shows my history of updates. Updates that have been installed and some that failed

Text

Description automatically generated

* **Get-WURebootStatus** – Shows reboot status to install newer updates. In this case no, the rebootrequired says ‘false’

### Lab 2c

**Task 1: Viewing process information in Linux**

The **ps** command shows running processes.

**[root@benjamin ~]# ps**

PID TTY TIME CMD

3032 pts/0 00:00:00 bash

3335 pts/0 00:00:00 ps

* **ps -ef** shows every process running in the current shell of our linux system
* **ps -ef --forest** shows every running process with another column that displays a process tree. The tree shows parent and child processes as well as the locations of commands/services within the file system.
* Processes near the top of the list when using the **top** command are gnome-shell and sssd\_kcm. Since I am not at UTS I use the VM for the next 2 parts for memory and swap.
  + There is 1960.1 MiB of physical memory installed where 255 MiB of it is free
  + There is 2048 MiB of Swap installed and 1931MiB of it free

**Task 2: Viewing and changing process priorities in Linux**

**From dd, PID is 3674 and has a default nice value of 0 and priority of 20.**

* After killing the **dd** process via the command **kill 3674,** going back to the first terminal shows ‘terminated’.
* Entering the command **nice –n 15 dd if=/dev/zero of=/dev/null**, using **top** shows dd now has a nice value of 15. The new PID is 3818.
* Running the command **renice -20 3818** makes **dd** the **highest priority**. Nice value of -20 and priority of 0.
* **top** while using command **rpm -qa** allowed me to get a nice value of 0 and priority of 20
* r**enice 19 35064** makes dd have a new priority of 39 and a nice value of 19.

**Task 3: Job control in Linux**

* Pressing **Ctrl-Z** in the first console window tells me that **dd** has ‘**stopped**’.
* **bg 1** makes **dd** run in the background. ‘1’ is the job number. To bring **dd** back to the foreground use **fg 1**. After using **fg 1,** we can see anything we input is echoed to the screen. We can use **Ctrl-C** to kill the process

Text

Description automatically generated

**Task 4: Viewing processes in Windows**

To view processes in Windows we can use the Task Manager. Going to the details tab, we can select various columns that we want to appear in the table by right clicking the column heading and selecting ‘**Select Columns’** as seen in the picture below:

Graphical user interface, application

Description automatically generated

**Get-Process** command in PowerShell shows running processes, however, the processes aren’t updated in real time unlike Task Manager. To show all Window processes 50MB or greater in size use command: **Get-Process | Where-Object {$\_.WorkingSet -gt 50000000}**

Table

Description automatically generated

**Task 5: Killing a process in Windows**

In this exercise we try to kill **cmd.exe**. Killing **cmd.exe** in Task Manager makes **cmd.exe** disappear. Using commands **Stop-Process –ID 4804** and **taskkill /PID 4726** in PowerShell have the same effect.

### Lab 2d

**Task 1: Linux disk partitioning**

The /boot device is not managed by LVM. LVM doesn’t manage /boot as it would make the OS unbootable.

Running **swapoff -a,** we can see via the **top** command that the Swap MiB goes to 0 as we close the partition. Running **swapon -a** returns it back to 2048 MiB due to us reopening the partition.

Text

Description automatically generated

**Task 3: Windows Server Disk Management**

Graphical user interface, application, Word

Description automatically generated

* As seen in the picture above, I have 3 fixed partitions/volumes. I found this by going through **Server Manager🡪 File and Storage Devices 🡪 Volumes 🡪 Disks**. The main volume is 59.4GB with 42.8GB free space remaining which is our main OS hard drive. Unfortunately, I don’t have any spare USB’s. My last one corrupted doing my digital forensics assignment.
* To Manage disks we can create and format disk partitions by going into **Computer Management🡪 Storage 🡪 Disk Management.** In my case, I created a new VHD called ‘POGGERS’ with NTFS file system and 2GB in size. You can check image below. To create a new VHD right click disk Management and select create VHD.

Graphical user interface, application, Word

Description automatically generated

# Week 3

Week 3 Topic: Networking and services

## Lecture 3

Lecture discusses Network layers and addresses, how to configure networks on Windows and Linux, security via firewalls and iptables and NTP troubleshooting.

### Lab 3a

**Task 1: Querying your network configuration**

* **Ifconfig (Linux) and ipconfig (Windows):** Used to display network configuration information such as IP, Default Gateway, netmask, broadcast address etc.
* **route -n (Linux):** Shows Kernel IP routing table
* **route print -4 (Windows):** Shows IPv4 routing table
* We can also find DNS Suffix for linux using the command **cat /etc/resolv.conf**

Using the commands above to fill table below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Linux: ens33** | | **Windows: Ethernet0** | |
| **DNS Suffix** | 192.168.192.2 | **DNS Suffix** | 192.168.192.2 |
| **IP Address** | 192.168.192.128 | **IP Address** | 192.168.192.129 |
| **Subnet Mask** | 255.255.255.0 | **Subnet Mask** | 255.255.255.0 |
| **Default Gateway** | 192.168.192.2 | **Default Gateway** | 192.168.192.2 |

**Task 2: Designing the network**

|  |  |  |  |
| --- | --- | --- | --- |
| **Linux: ens37** | | **Windows: Ethernet1** | |
| **DNS Suffix** | 10.0.2.1 | **DNS Suffix** | 10.0.2.1 |
| **IP Address** | 10.0.2.1 | **IP Address** | 10.0.2.2 |
| **Subnet Mask** | 255.255.255.0 | **Subnet Mask** | 255.255.255.0 |
| **Default Gateway** | 10.0.2.1 | **Default Gateway** | 10.0.2.1 |

**Task 3: Command-line configuration**

In this task, DHCP should be configured for the VM’s. My tasks is to configure ens37 in centos and Ethernet1 in Windows Server to setup our own private network.

**3a: Linux server**

**ifconfig** is used to configure a temporary IP address of 10.0.2.1. Default gateway is configured via **route** command to 10.0.2.1

**[root@benjamin ~]# ifconfig ens37 10.0.2.1 netmask 255.255.255.0**

**[root@benjamin ~]# route add default gw 10.0.2.1**

Viewing the routing table with **route -n.**

Text

Description automatically generated

**Pinging ens33 works but ens37 no**, probably due to multiple default gateways. We should keep the ens33 default gateway for internet.

**3b: Windows server**

We should Disable Ethernet 0 and enable Ethernet1. To configure a static IP on Windows, navigate to **Server Manager** à **Local Server** à **Ethernet1**

Right click **Ethernet1** à **Properties** à **Internet Protocol Version 4 (TCP/IPv4)** à **Properties** and configure the interface to have IP 10.0.2.2, sub netmask 255.255.255.0 and Default gateway of 10.0.2.1. We can check if everything is correct via ipconfig and pinging 10.0.2.1 which is pingable.

A screenshot of a computer

Description automatically generated

Text

Description automatically generated

**Task 4: Setting up the firewall**

**4a: Windows server**

In this task, we must disable the firewall as it might be blocking ICMP requests. To do this we must navigate to **Control Panel 🡪 Systems and Security🡪 Windows Defender Firewall 🡪 Allow apps to communicate through Windows Firewall 🡪 Make sure file and printer sharing is selected** **for both private and public**

**Graphical user interface, text, application, email

Description automatically generated10.0.2.2 should now be pingable**

Text

Description automatically generated

**4b: Linux**

Using the command **systemctl status firewalld** in centos to verify that the firewall is active. We want to now turn on the echo reply on in **firewall-config**. Run the command and make sure ens37 is public. After this go to **Zones 🡪 ICMP Filter 🡪 Make sure echo-reply is unchecked so we can ping!**

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

**Task 5: Modifying Linux network configuration files**

We now want to Modify the value of ONBOOT from no to yes on interface33. We can do this from the following command: **vim /etc/sysconfig/network-scripts/ifcfg-ens33**. Once we are in the file, make sure BOOTPROTO=dhcp, DEFROUTE=yes, ONBOOT=yes, NAME and DEVICE are ens33 and that there should be no IP and netmask within the file.

**[root@benjamin ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens33**

TYPE=Ethernet

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=dhcp

DEFROUTE=yes

IPV4\_FAILURE\_FATAL=no

IPV6INIT=yes

IPV6\_AUTOCONF=yes

IPV6\_DEFROUTE=yes

IPV6\_FAILURE\_FATAL=no

IPV6\_ADDR\_GEN\_MODE=stable-privacy

NAME=ens33

UUID=9a5b99db-9450-44c5-aece-fbfb20f28e7d

DEVICE=ens33

ONBOOT=yes

We should now Configure interface ens37 by copying the network configuration of ens33 and editing it via the command: **cp /etc/sysconfig/network-scripts/ifcfg-ens33 /etc/sysconfig/network-scripts/ifcfg-ens37 🡪** then enter **vim /etc/sysconfig/network-scripts/ifcfg-ens37.** Make sure Device and Name are ens37, there is no UUID line, BOOTPROTO=none, IPADDR=10.0.2.1 and NETMASK=255.255.255.0, DEFROUTE=no, ONBOOT=yes

**[root@benjamin ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens37**

TYPE=Ethernet

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=none

IPV4\_FAILURE\_FATAL=no

IPV6INIT=yes

IPV6\_AUTOCONF=yes

IPV6\_DEFROUTE=yes

IPV6\_FAILURE\_FATAL=no

IPV6\_ADDR\_GEN\_MODE=stable-privacy

NAME=ens37

DEVICE=ens37

ONBOOT=yes

IPADDR=10.0.2.1

NETMASK=255.255.255.0

For the changes to take effect use the command **ifdown ens37** then **ifup ens37.** Check everything is correct via **ifconfig** and **route -n.** 10.0.2.2 should be pingable. Reboot after this for changes to be permanent!

Text

Description automatically generated

### Lab 3b

**Task 1: Set the time zone**

Entering command **date** shows the incorrect time. Entering **date -u** however, shows the correct time. In this case we want to correct our time zone by changing /etc/localtime into a symbolic link to the correct file for our time zone under the directory /usr/share/zoneinfo/

1. Ls /usr/share/zoneinfo/Australia – to list Australian Timezone’s such as Sydney and Perth
2. Now use command: ln -s /usr/share/zoneinfo/Australia/Sydney /etc/local/time
3. If it fails to create a link use command: mv /etc/localtime /etc/localtime.bak
4. ln -s -f /usr/share/zoneinfo/Australia/Sydney /etc/local/time

**Task 2: Maintain correct time with chrony**

Enable chronyd using the commands: **systemctl enable chronyd, systemctl start chronyd**

After enabling chronyd use the following commands **chronyc sources** and **chronyc tracking** to verify chronyd is working:

Text

Description automatically generated

**Now add server time.uts.edu.au:**

**[root@benjamin ~]# chronyc add server time.uts.edu.au**

200 OK

Now check chronyc sources again. There should be 5 sources now instead of 4 from before. The new one for me is **B1L2-c7206.gw.uts.edu.au with 0ns.** To test if synchronization is working or not we will change the date to 1999 New Year’s Eve by entering command **date 123123591999.00**. It does not reset date/time back to 2021. To reset we must use command systemctl restart chronyd.

**Task 3: Windows NTP**

To Change NTP Server in Windows navigate to **Server Manager 🡪 Local Server 🡪 Click on Time Zone 🡪 Internet Time tab 🡪 Change Settings to 2.pool.ntp.org**

Graphical user interface, application

Description automatically generated

# Week 4

Week 4 topic: Dynamic networking

## Lecture 4

In lecture 4 we are introduced to DHCP, its operations, how DHCP is configured and network configuration of DHCP servers.

### Lab 4a

**Task 1: Linux command-line configuration**

We first have to test network connectivity. Pinging [www.google.com](http://www.google.com), it seems that my network connectivity is fine. Typing **ifconfig ens33** and I can see that my ip is still 192.168.192.128. Using command **nmcli** and both ens33 and ens37 are connected!

Typing command **vim /etc/sysconfig/network-scripts/ifcfg-ens37** we want to convert ens37 from static to DHCP. Make sure BOOTPROTO=dhcp and IPADDR and Netmask have a # to be commented out. After this, Reboot.

**[root@benjamin ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens37**

TYPE=Ethernet

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=dhcp

IPV4\_FAILURE\_FATAL=no

IPV6INIT=yes

IPV6\_AUTOCONF=yes

IPV6\_DEFROUTE=yes

IPV6\_FAILURE\_FATAL=no

IPV6\_ADDR\_GEN\_MODE=stable-privacy

NAME=ens37

DEVICE=ens37

ONBOOT=yes

#IPADDR=10.0.2.1

#NETMASK=255.255.255.0

**Task 3: Windows configuration**

Enabled Ethernet 0 again. To convert Ethernet1 for Windows from static to dynamic i used the command:

**netsh interface ip set address "Ethernet1" dhcp**

Using the command **netsh interface ip show config** shows me the configuration of both interfaces eth0 and eth1. As seen below, eth1 is a bit different as I used dhcp to dynamically assign the ip address earlier from the command above **netsh interface ip set address "Ethernet1" dhcp**

Text

Description automatically generated

### Lab 4b

**Task 1: Install DHCP server on Linux**

To make the Linux virtual machine operate as a DHCP server the following command needs to be used:

**[root@benjamin ~]# yum install dhcp-server**

**Task 2: Configure DHCP server on Linux**

Task 2 focuses on configuring the DHCP server on Linux. We use the following table for ens37 from the lab manual:

|  |  |
| --- | --- |
| **Subnet** | 10.0.2.0/24 |
| **Gateway** | 10.0.2.1 |
| **DNS** | 10.0.2.1 |
| **Reserved space for servers** | 10.0.2.2 à 10.0.2.127 |
| **Dynamically allocated space for workstations etc.** | 10.0.2.128 à 10.0.2.254 |

To configure the DHCP server we need to read the sample of the dhcpd.conf file located in /usr/share/doc/dhcp-server/dhcpd.conf.example and edit the real config file /etc/dhcp/dhcpd.conf

We now Edit the /etc/dhcp/dhcpd.conf file via **vim /etc/dhcp/dhcpd.conf** and enter the following contents:

subnet 10.0.2.0 netmask 255.255.255.0 {

range 10.0.2.128 10.0.2.254;

option domain-name-servers 10.0.2.1;

option domain-name "online.uts.edu.au";

option routers 10.0.2.1;

default-lease-time 60;

max-lease-time 600;

}

Before we start the DHCP server on ens37 we need to make sure the static address is 10.0.2.1 with netmask 255.255.255.0. Since I changed it from Lab 4a, I change it back using the command **vim /etc/sysconfig/network-scripts/ifcfg-ens37**

**[root@benjamin ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens37**

TYPE=Ethernet

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=none

IPV4\_FAILURE\_FATAL=no

IPV6INIT=yes

IPV6\_AUTOCONF=yes

IPV6\_DEFROUTE=yes

IPV6\_FAILURE\_FATAL=no

IPV6\_ADDR\_GEN\_MODE=stable-privacy

NAME=ens37

DEVICE=ens37

ONBOOT=yes

IPADDR=10.0.2.1

NETMASK=255.255.255.0

I then used the commands **nmcli con reload ens37** and then **nmcli con up ens37** to reload the interface. **I** then used the command **ip a** to make sure ens37 configurations were correct and back to static IP 10.0.2.1

**Task 3: Start and monitor the DHCP server on Linux**

We start dhcp service using **systemctl start dhcpd.** To monitor the DHCP I used another tab with the command **tail -f /var/log/messages.**

Typing command **cat /var/lib/dhcpd/dhcpd.leases** we can see that there is no leases as of yet.

**Task 4: Set up Windows DHCP client & test the DHCP server**

We know disable and enable Ethernet1 on Windows. We change the settings to ‘Obtain an IP Address Automatically’ and ‘Obtain DNS server address automatically’. After disabling and enabling Ethernet1 on Windows, an address should be leased correctly. We can use command **ipconfig** to check Ethernet1 starting IP address is 10.0.2.128 as in the file **/etc/dhcp/dhcpd.conf** the range should start from 10.0.2.128 as well as the DNS Suffix being **online.uts.edu.au**

Text

Description automatically generated with low confidence

We can now use the following command **cat /var/lib/dhcpd/dhcpd.leases** to find the log information of a lease existing now in the linux VM:

Text

Description automatically generated

Text

Description automatically generated

The command **arp -I ens37** confirms our MAC address on the ens37 interface.

We can also now use the command **tail -f /var/log/messages** to find the request/response sequence of the WindowsServer by the following messages: **DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, DHCPACK** as seen in the image below.

Text

Description automatically generated

**Task 5: Set up reserved addresses**

In this task we will reserve an IP address for Windows. We first find our Window Server MAC address via the command **arp -I ens37.** In my case it is **00:0c:29:d7:90:c7**

We then modify the DHCP server configuration file by editing the file **/etc/dhcp/dhcpd.conf** with vim and add host entry as seen below:

**[root@benjamin ~]# vim /etc/dhcp/dhcpd.conf**

host WinServer {

hardware ethernet 00:0c:29:d7:90:c7;

fixed-address 10.0.2.20;

}

For the changes to take effect, restart the service via the command **systemctl restart dhcpd**.

In the Windows Server VM use the command **ipconfig /renew.** Pinging 10.0.2.1 now (the linux vm) we get reply’s so everything is working correctly.

Using the command **arp -I ens37** we can now see 2 IP’s, 10.0.2.128 and 10.0.2.20 which is mapped to the MAC address 00:0c:29:d7:90:c7.

For the old lease using command **cat /var/lib/dhcpd/dhcpd.leases** we can still see that the lease is 10.0.2.128. The DHCP server still thinks the fixed-address assignment is not considered by the dhcpd process and will hence not record it.

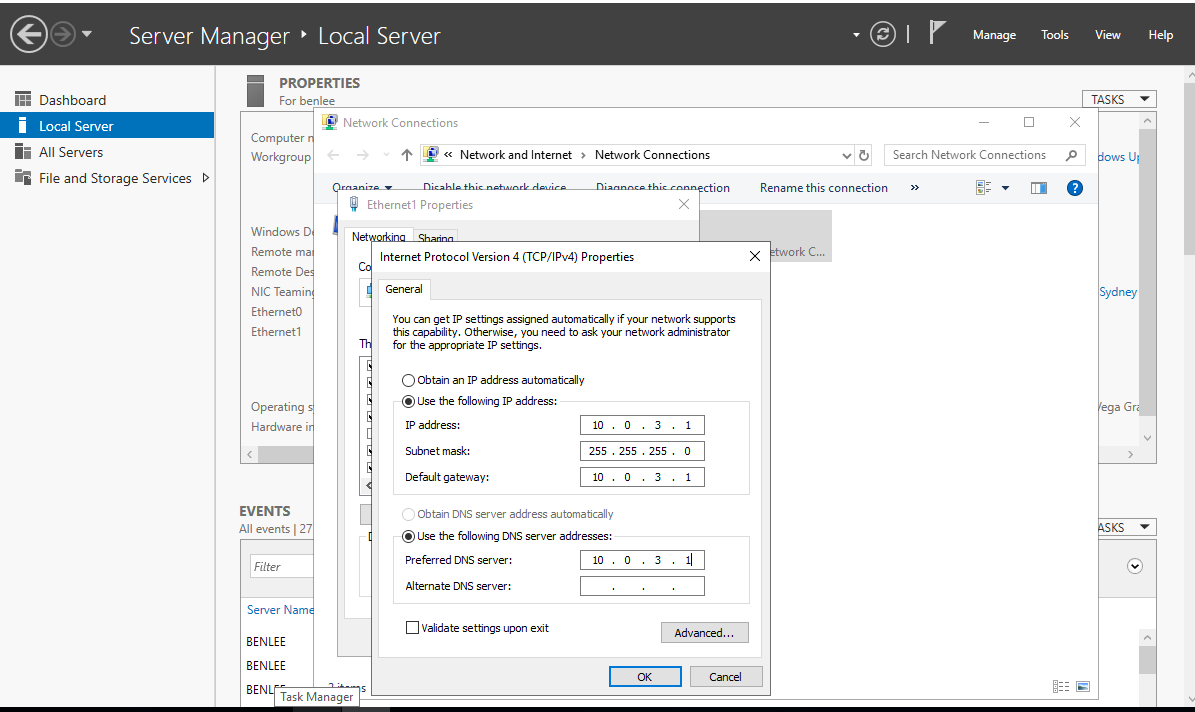
### Lab 4c

**Task 1: Design our network**

In this task we want to design our own subnet. The subnet should be configured to 10.0.3.0/24 instead of 10.0.2.0/24 which we have done from pervious activities. In Windows, DHCP Range is called ‘scope’.

**Task 2: Configure the Windows Server network interface with a static IP address**

To configure DHCP on Windows we must first configure Ethernet1 with static address 10.0.3.1. See picture below.



**Task 3: Install the DHCP role on Windows Server**

We must now setup the DHCP server role from the Server Manager. Do this via **Server Manager** à **Manage** à **Add Roles and Features** 🡪 **In Installation click role-based 🡪 Select Server 🡪 Select DHCP Server in Server Roles and include management tools 🡪 Skip Features and DHCP Server and Install**. At the End you should have something like this when installing from picture below

Graphical user interface, text, application

Description automatically generated

We should now see a new **DHCP option in the Server Manager**. I see a warning called ‘Configuration required for DHCP Server... ‘. Selecting ‘More’ I completed the DHCP configuration process. After this iright-click the server in the Servers menu and choose ‘**DHCP Manager’**. Right click IPv4 and select ‘New Scope’ and follow the wizard instructions.

We then create a name and description which should be ‘winrange’ for the name, and ‘windows range description’ for the description.

Graphical user interface, application

Description automatically generated

We now add an IP Range of 10.0.3.129 🡪 10.0.3.254. Make sure length is 24 with subnet mask of 255.255.255.0 🡪After this leave, everything blank for the **add exclusions and delay** section 🡪Click Next 🡪 For lease Duration make it 1 minute 🡪 Configure DHCP Options as yes 🡪Add 10.0.3.1 to the router and click next 🡪 Make parent domain uts.edu.com and remove the extra IP address 192.168.192.2 and click next 🡪 For WINS Servers leave blank 🡪 Activate Scope.

Once completed it should look something like this:

Graphical user interface, text, application

Description automatically generated

To change anything right click the scope and select properties

Graphical user interface, text, application

Description automatically generated

**Task 4: Monitoring the DHCP server**

To monitor the DHCP server right-click on **IPv4** à **Display Statistics**. This will show information on DHCP traffic such as server up and start time, Acknowledgements, requests and much more.

Text

Description automatically generated with medium confidence

We can find logs and lease database for DHCP role in C:\WINDOWS\system32\dhcp. To view locked files go to **View 🡪 Options 🡪 Change Folder 🡪 Uncheck ‘Hide extensions for known file types’**

When going through **Event viewer** via **Server Manager 🡪 Tools 🡪 Event Viewer 🡪 Windows Logs 🡪 System** we can find a lot of dhcp warning logs by finding ‘find…’ in the right hand toolbar and searching for ‘dhcp’

Graphical user interface, application

Description automatically generated

**Task 5: Set up Linux client and testing**

In this task make sure dhcpd is disabled via command **systemctl stop dhcpd.** We should change the network card configuration via the command **vim /etc/sysconfig/network-scripts/ifcfg-ens37** to change ens37 to dynamic. Make BOOTPROTO=dhcp and comment out IPADDR and NETMASK.

**[root@benjamin ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens37**

TYPE=Ethernet

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=dhcp

IPV4\_FAILURE\_FATAL=no

IPV6INIT=yes

IPV6\_AUTOCONF=yes

IPV6\_DEFROUTE=yes

IPV6\_FAILURE\_FATAL=no

IPV6\_ADDR\_GEN\_MODE=stable-privacy

NAME=ens37

DEVICE=ens37

ONBOOT=yes

#IPADDR=10.0.2.1

#NETMASK=255.255.255.0

* After this we disable and reenable ens37 via the command **nmcli con reload ens37 and then nmcli con up ens37.**
* With the command **ifconfig ens37** we see **IP 10.0.3.129.**
* Using **tail -f /var/log/messages** we can see we have registered a new address and many other interesting logs as seen in the picture below.

Text

Description automatically generated

* In **/var/lib/NetworkManager** by cd’ing into the directory, we can see 2 internal networks cards.
  + They have addresses 10.0.3.129 and 192.168.192.128 when we ‘cat’ there info
* Using **cat /etc/resolv.conf** we can see the DNS 10.0.3.1
  + Pinging 10.0.3.1 seemed to be successful from the centos VM as well as 10.0.3.129 from the Windows Server VM
* Checking the log file DhcpSrvLog-Thu in C:\WINDOWS\system32\dhcp shows the address 10.0.3.129 was leased with domain name **uts.edu.com**

**Task 6: Set up a reserved address**

In this task, we try to set up a reserved address to the linux VM. We first find the MAC address of ens37 via the command **ifconfig ens37.** In my case, the MAC address is 00:0c:29:e6:9e:dc

We then go to DHCP Manager in the Windows Server. Under scope find ‘Reservations’ and right click to add ‘new reservation’. Input reservation name ‘linux’ and set the ip address 10.0.3.30 alongside the mac address with format **00-0c-29-e6-9e-dc or 000c29e69edc** and add.

Graphical user interface

Description automatically generated

Use command **ifconfig ens37** on linux to verify that we have obtained the new address 10.0.3.30 and the old IP has expired. It didn’t work the first time so I had to use **nmcli con up ens37** to refresh the configuration. The result is below!

Text

Description automatically generated

# Week 5

Week 5 topic: Managing users and groups

## Lecture 5

The lecture discusses user and group management policies, /etc/passwd and etc/shadow contents, how to manage user accounts by adding, deleting or modifying them, system issues, workgroups and domains and how to generate/create passwords

### Lab 5a

**Task 1: Creating users with command-line tools**

In this task, we try to create a user with username: peter and full name: Peter Griffin with default uid and gid. We must also use his preferred login shell ‘zsh’.

Creating a user with the username peter with system defaults and full name Peter Griffin, the user prefers Z shell. Using **man useradd** we can see various options. We can set a password using the useradd command **useradd -p**. However, we should not use this command as it will create an encrypted password that can be visible by all users listing processes. This can be bad for privacy reasons so we don’t use it.

We should start by adding the user with the **useradd** command seen below. We use **useradd -c** for a short description ‘Peter Griffin’. We use **-s** to rename the zsh user login shell to peter.

**[root@benjamin ~]# useradd -c "Peter Griffin" -s /usr/bin/zsh peter**

We can now modify the password using the **passwd** command to set a password for Peter. I made the password ‘**oogabooga12345**’ in my lab exercise.

**[root@benjamin ~]# passwd peter**

Changing password for user peter.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

Using command **ls /home/peter** we can see there is nothing in the directory at the moment.

We now want to supply a README file in the user home directory. We do this by modifying the skeleton directory /etc/skel . We can create a README file in the /etc/skel directory via the **touch command.**

**[root@benjamin ~]# touch /etc/skel/README**

We now create a user called ‘Stewie Griffin’ with username: stewie and preferred shell ‘bash’. The password I used for stewie is ‘**chillin445’**. We can type the command **ls /home/stewie** after user has been created to verify the README file exists in stewie’s directory. We can do this via the following commands below:

**[root@benjamin ~]# useradd -c "Stewie Giffin" -s /bin/bash stewie**

**[root@benjamin ~]# passwd stewie**

Changing password for user stewie.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

**[root@benjamin ~]# ls /home/stewie/**

README

We now create a 3rd user brian in the default group ‘users’ with **UID 200**. We can do this via the **-u** option in the **useradd command** for the UID and **-g** for changing the default group**.** Bash shell should be default. Username should be brian and his full name is ‘Brian Griffin’. The password I used is ‘**rubydelight13’**

**[root@benjamin ~]# useradd -c "Brian Giffin" -u 200 -g users -s /bin/bash brian**

**[root@benjamin ~]# passwd brian**

Changing password for user brian.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

We can test that we can log in via SSH. In this example I will ssh into brian’s account. Using command **id** we can see that Brian is confirmed to be part of group ‘users’ with UID 200.

**[root@benjamin ~]# ssh brian@localhost**

**[brian@benjamin ~]$ id**

uid=200(brian) gid=100(users) groups=100(users)

**Task 2: Configuring groups and assigning users to groups**

We now create the group family and assign Brian and Stewie to the group. We must go back to root first however with command **su -u.** After making stewie and brian members of the group, we can check they are in the group via the **tail /etc/group** command which should show **family:x:1002:stewie,brian.** We can verify they are they part of the group by using command **id** again.

**[root@benjamin ~]# groupadd family**

**[root@benjamin ~]# usermod -G family brian**

**[root@benjamin ~]# usermod -G family stewie**

**[root@benjamin ~]# ssh brian@localhost**

**[brian@benjamin ~]$ id**

uid=200(brian) gid=100(users) groups=100(users),1002(family)

Being logged in as Brian we now create a file called ‘a’ in the home directory. Group should be ‘users’. Running the **ps** command the PID is 6302.

**[brian@benjamin ~]$ touch a**

**[brian@benjamin ~]$ ls –la a**

-rw-r--r-- 1 brian users 0 Sep 10 05:35 a

**[brian@benjamin ~]$ ps**

PID TTY TIME CMD

6302 pts/2 00:00:00 bash

6357 pts/2 00:00:00 ps

We now change the current group using the command **newgrp family.** We create a new file CALLED b. Using command **ls -la b** we can see user is brian but group is now in the ‘family’ group.

**[brian@benjamin ~]$ newgrp family**

**[brian@benjamin ~]$ id**

uid=200(brian) gid=1002(family) groups=1002(family),100(users)

**[brian@benjamin ~]$ touch b**

**[brian@benjamin ~]$ ls –la b**

-rw-r--r-- 1 brian family 0 Sep 10 05:36 b

The **newgrp** command changes current real group to the named group. Using the **man pages**, **newgrp** reinitializes user environment even though user has logged in. Using the **ps** command we can see 3 instances: bash, bash, ps. To reverse the effects of **newgrp** you can use a -, so the environment can be changed. For example **newgrp -family**

**Task 3: Modifying user account settings**

In this task we want to modify peter’s account to expire in 5 days. Make sure we are back in root via the **su –** command. We first use **usermod** to modify peter’s account. **-e** changes the date the user account will be disabled. We can see the modifcations via the command **chage -1 peter.**

**[root@benjamin ~]# usermod -e 2021-09-15 peter**

**[root@benjamin ~]# chage -l peter**

Last password change : Sep 09, 2021

Password expires : never

Password inactive : never

Account expires : Sep 15, 2021

Minimum number of days between password change : 0

Maximum number of days between password change : 99999

Number of days of warning before password expires : 7

Commands below show me changing Stewie’s account for his password to expire every 5 days.

**[root@benjamin ~]# chage -M 5 stewie**

**[root@benjamin ~]# chage -l stewie**

Last password change : Sep 9, 2021

Password expires : Sep 14, 2021

Password inactive : never

Account expires : never

Minimum number of days between password change : 0

Maximum number of days between password change : 5

Number of days of warning before password expires : 7

We can lock brian’s account by using command **usermod -L brian**. Using **ssh brian@localhost** command we get a **‘Permission denied, please try again’** error. Using command **usermod -U brian** unlocks brian’s account and we can login fine again.

**Task 4: Creating a user without using command-line tools**

In this task we create a user ‘lois’ **WITHOUT** **useradd command.** We makre sure lois full name is ‘Lois Griffin’ with bash shell. We use command **vipw**  to add lois as a user manually and **vigw** to modify his group manually as well.

**[root@benjamin ~]# vipw**

lois:x:1003:1003:Lois Griffin:/home/lois:/bin/bash

**[root@benjamin ~]# vigr**

lois:x:1003:

We now change password for lois using **passwd** command. The password I chose was ‘**poyostick36’.** We now create a home directory with skeleton files and then we copy the skeleton files to /home/lois via the command **cp -av** **/etc/skel/ /home/lois** where **-a** is for all files and **-v** is for verbose log information.

**[root@benjamin ~]# passwd lois**

**[root@benjamin ~]# cp -av /etc/skel/ /home/lois**

* Check home directory of lois via command **ll -d /home/lois.** We want to change the properties of the group via the commands **chown -R lois:lois /home/lois** and **chmod 700 /home/lois.** Image below shows these commands as well as the **cp -av** command from task above.

Text

Description automatically generated

* Checks:
  + **Does lois have a home directory in the right location?** 
    - Lois’ home directory is in the right location
  + **Who is the user owner and group owner of lois' home directory?** 
    - The user owner and group of lois home directory is now ‘lois’, but as seen in picture above it was originally ‘root’ using the **ll -d /home/lois** command
  + **What files are present in lois' home directory?** 
    - README file is the only one
  + **How did you ensure that you copied the dot files to lois' home directory?** 
    - Use command **cp -av /etc/skel /home/lois**
  + **What are the ownerships of these files?** 
    - The ownership of these files should be root. To change ownership use **chown -R lois:lois /home/lois** command

**Task 5: Using the GUI**

Graphical user interface

Description automatically generated

We see all users except for Brian. This is most likely because we modified his UID to 200 making him an admin.

**Task 6: Using Windows Server GUI to add users and groups**

In this task we create users and groups. We do this by going to **Server Manager🡪 Local Server 🡪 Tools menu 🡪 Computer Management 🡪 Local User’s and Groups 🡪 Select Users and then right click to create new users peter, stewie, brian, lois**

Graphical user interface, application

Description automatically generated

We then do the same thing with the groups and add there group (**family**) and assign them as members to the group

Graphical user interface, application

Description automatically generated

**Task 7: Using Windows Server command line to add users and groups**

In this task we use powershell to create a new user called ‘joe’ with full name ‘Joe Swanson’ via the command: New-LocalUser -Name "joe" -FullName "Joe Swanson"

* **Get-LocalUser** shows current users
* **Disable-LocalUser joe** disables joe’s account
* **Enable-LocalUser joe** renables joe’s account

Text

Description automatically generated

* **net user bonnie /add /fullname:"Bonnie Swanson"** – creates a user named Bonnie Swanson
* **net user –** shows user accounts of the windows machine
* **net user bonnie –** shows information on the user ‘bonnie’ including name, password expiry dates, account activity and much more
* **wmic useraccount where "name='bonnie' " –** shows the Account Type of the user account ‘bonnie’ as well as password information such as password expiry, if the password is changeable, SID etc.

Text

Description automatically generated

### Lab 5b

**Task 1: Executing root commands safely via sudo in Linux**

**Visudo** command edits the file /etc/sudoers. This file allows non-adminstrative or non-root users to perform administrative tasks via **sudo**.

We first use **visudo** and add the following configuration underneath **##Allow root to run any commands anywhere**:

**[root@benjamin ~]# visudo**

peter ALL=/sbin/ifconfig

Looking at the file /etc/sudoers from **visudo** we can see many commented-out permissions such as User Aliases, Host Aliases, Storage, Delegating permissions and much more.

%users ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom might be necessary to allow members of the users group to mount and unmount the cdrom as root. Unfortunately I do not know why it only matches /mnt/cdrom

We now login as Peter via ssh or su and test the command **sudo** to disable and reenable ens33.

**[root@benjamin ~]# su – peter**

**[peter@benjamin ~]~% id**

uid=1000(peter) gid=1000(peter) groups=1000(peter)

**[peter@benjamin]~% sudo ifconfig ens33 down**

We trust you have received the usual lecture from the local System

Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.

#2) Think before you type.

#3) With great power comes great responsibility.

[sudo] password for peter:

**[peter@benjamin]~% sudo ifconfig ens33 up**

[sudo] password for peter:

* If an administrator wanted to grant all users the ability to disconnect or reconnect the ens33 port we should add in **visudo**: peter ALL=(ALL) ALL instead of peter ALL=/sbin/ifconfig
* Executing **visudo** with the rule above peter ALL=(ALL) ALL , we should be able to run the commands again as root without the need to enter a password again.
* If we run the command: sudo su – , it allows the user to switch to root requiring user’s current password. It is a bad thing, as we shouldn’t allow normal users to have administrative privileges that they don’t need in the real world as it can be a security risk.

**Task 2: Executing admin commands in Windows Server**

We can run programs as another user via the **runas** command in Windows.

We can execute the cmd shell as administrator via the command **runas /user:Administrator cmd.exe** while logged in as a normal user.

Doing this prompts me with an admin password. I then get a new command prompt with cmd title: **Administrator: cmd.exe (running as BENLEE\Administrator)**

Entering the command **runas /user:peter cmd.exe,** I get another command prompt asking me for peter’s password. The cmd title now shows: **cmd.exe (running as BENLEE\peter)**

**Task 3: Notifying users on system issues or computer usage policies (Linux – the old way)**

In this task we need to edit the file **/etc/motd** that shows a message of the day when a user logs out and logs back in.

We then edit 3 other files as seen below alongside **/etc/motd**:

* **/etc/issue, /etc/issue.net, /etc/sshd\_config**

**[root@benjamin ~]# vim /etc/motd**

Morning >:D

**[root@benjamin ~]# vim /etc/issue**

33333333333333333333333

\S

Kernel \r on an \m

44444444444444444444444

**[root@benjamin ~]# vim /etc/issue.net**

55555555555555555555555

\S

Kernel \r on an \m

66666666666666666666666

**[root@benjamin ~]# vim /etc/ssh/sshd\_config**

# no default banner path

Banner /etc/issue.net

**[root@benjamin ~]# systemctl restart sshd**

Using SSH to log into peter we get the contents of the /etc/issue.net file due to the message of the day.

**[root@benjamin ~]# ssh peter@localhost**

55555555555555555555555

\S

Kernel \r on an \m

66666666666666666666666

peter@localhost's password:

Morning!

Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Sep 1 07:20:12 2021

* Pressing Ctrl-Alt-F3 for text-based login, when I try to login to one of my users peter we do see **/etc/motd** displayed with ‘Morning!’ but no **/etc/issue.net or /etc/issue**
* For graphical login using Ctrl-Alt-F1, and logging into peter no message of the day or **/etc/issue.net or /etc/issue** appears
* Using SSH /**etc/issue.net or /etc/issue** does appear as well as **/etc/motd**

**Task 4: Notifying users on system issues or computer usage policies (Linux – the graphical way)**

In this task, we edit 2 files via **vim**.

* /etc/dconf/profile/gdm

**[root@benjamin ~]# vim /etc/dconf/profile/gdm**

user-db:user

system-db:gdm

file-db: /usr/share/gdm/greeter-dconf-defaults

* Make directory by command **mkdir /etc/dconf/db/gdm.d**
* /etc/dconf/db/gdm.d/01-banner-message

**root@benjamin ~]# vim /etc/dconf/db/gdm.d/01-banner-message**

[org/gnome/login-screen]

banner-message-enable=true

banner-message-text=’Greetings wonderful Linux user!’

* We then update the dconf database via command **dconf database**. Restart the gdm service by using command: **systemctl restart gdm.** We should get something like this as seen in picture below:

Graphical user interface, application

Description automatically generated

**Task 5: Notifying users on system issues or computer usage policies (Windows Server)**

In this task, we can display a message of the day on the login screen. We can do this by going to **Server Manager 🡪 Tools 🡪 Local Security Policy 🡪 Local Policies 🡪 Security Options 🡪 Edit Interactive logon: Message text for users attempting to log on AND interactive logon: Message title for users attempting to log on**

* For ‘Message text for users attempting to log on’ I put ‘Sunrise Bahammas’
* For ‘Message title for users attempting to log on’ I put ‘Hello Friend’

We now open powershell and run command gpupdate /force to update system policies.

* It should state ‘Computer Policy update has completed Successfully. User Policy update has completed successfully.’

We now log out and log back in to test. Result is shown below.

Graphical user interface

Description automatically generated

# Reflection

I haven’t had this much fun doing lab work for other subjects since forever. I learnt a lot doing this learning journal and the labs as they were super interactive and fun to complete. Looking at the videos and having the tutor provide some guidance in the tutorial really helped me understand the concepts and commands when writing the report.

At the start I had some trouble with my Virtual Machine’s network configuration not working. For some reason using my macbook with VMware Fusion, I wasn’t able to fix the problem of DHCP and internet not being available for the Network Interface Ethernet1. I tried configuring the settings of VMNet2 but nothing seemed to work. Even updating my Vmware version didn’t work. I even emailed the tutor for some advice but it still didn’t seem to work as well for some reason. I had to reinstall everything on my old PC with Vmware Workstation and by some miracle, the network configurations were working properly. I have no idea why. Maybe a software compatibility issue I am not sure.

The labs starting from week3-4 were the most challenging in my opinion because we were configuring DHCP and using some network commands. The labs themselves were very rewarding once completed and I enjoyed taking my time going through them. I didn’t really have any other challenges that were unsolved except for the /mnt/cdm question in Week 5 Lab 5b Task 1 where it asks: Why does it only match /mnt/cdrom, and not any device?

Other than that everything else was good. I definitely could have done a bit better if I started my journal a bit earlier for some of the later weeks.

Overall, I am satisfied with the amount of effort I put into this learning journal.

# References

How To Check The Runlevel In Linux - OSTechNix 2020, OSTechNix. viewed 31 July 2020, <https://ostechnix.com/check-runlevel-linux/>.

Ubuntu Manpage: mandb - create or update the manual page index caches 2020, Manpages.ubuntu.com. viewed 31 August 2020, <http://manpages.ubuntu.com/manpages/trusty/en/man8/mandb.8.html>.