**Lexislación e Seguridade Informática:**

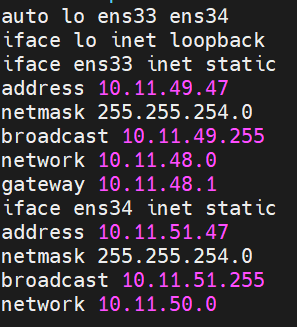
*Práctica I. Configuración básica*

**3.2.7-49.47**

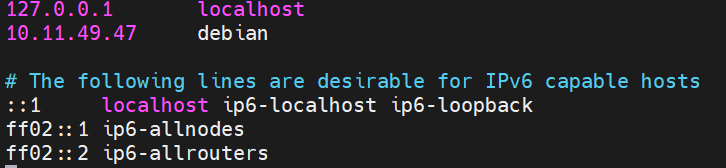
**ssh lsi@10.11.49.47**

1. *Configure su máquina virtual de laboratorio con los datos proporcionados por el profesor. Analice los ficheros básicos de configuración (interfaces, hosts, resolv.conf, nsswitch.conf, sources.list, etc.)*

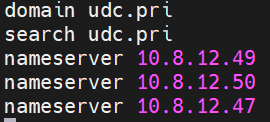
**/etc/network/interfaces**: describe las interfaces de red



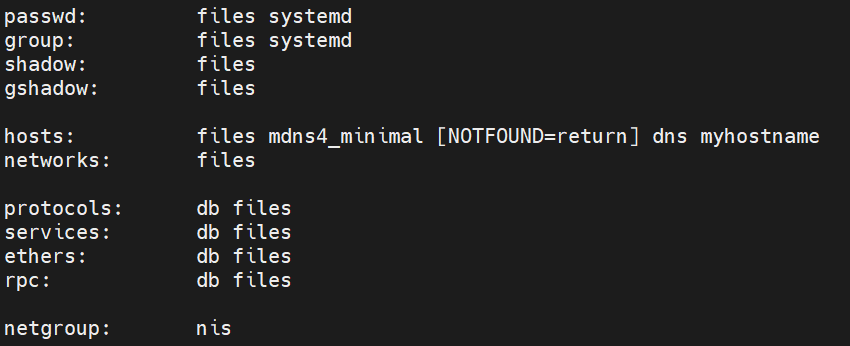
**/etc/hosts**: mapea nombres de host a direcciones ip



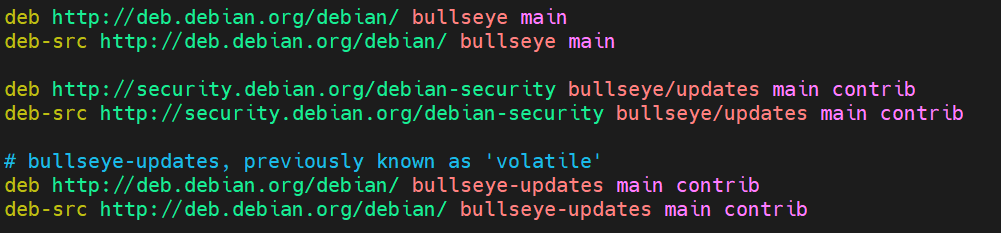
**/etc/resolv.conf**: se encarga de configurar parte del sistema de resolución de nombres de dominio



**/etc/nsswitch.conf:** define el orden de búsqueda de las bases de datos de red. La primera columna es la base de datos. El resto de la línea especifica cómo funciona el proceso de búsqueda. Se puede especificar individualmente la forma en la que trabaja cada base de datos.



**/etc/apt/sources.list:** Fuente de soporte del Sistema Operativo



1. *¿Qué distro y versión tiene la máquina inicialmente entregada? Actualice su máquina a la última versión estable disponible.*

Inicialmente se nos entrega: Debian 10 (Buster)

Tras actualizar: Debian 11 (Bullseye)

Para realizar la actualización:

* Actualizamos Debian 10 al máximo:
  + - sudo apt update
    - sudo apt upgrade
    - sudo apt full-upgrade
    - sudo apt --purge autoremove
* Hacemos reboot
* Modificamos el archive /etc/apt/sources.list sustituyendo “buster” por “bullseye”
* Actualizamos
  + - sudo apt update
    - sudo apt full-upgrade
* Hacemos reboot

Para saber la versión actual: lsb\_release -a

1. *Identifique la secuencia completa de arranque de una máquina basada en la distribución de referencia (desde la pulsación del botón de arranque hasta la pantalla de login). ¿Qué target por defecto tiene su máquina? ¿Cómo podría cambiar el target de arranque? ¿Qué targets tiene su sistema y en qué estado se encuentran? ¿Y los services? Obtenga la relación de servicios de su sistema y su estado. ¿Qué otro tipo de unidades existen?*

Secuencia de arranque de la máquina

(comando **dmesg // systemctl list-dependencies default.target**):

1. La BIOS toma el control, utilizando memoria de solo lectura
2. La BIOS realiza comprobaciones y obtiene parámetros de memoria no volátil
3. La BIOS detecta discos duros y lanza el MBR (Master Boot Record)
4. La BIOS ejecuta el gestor de arranque (habitualmente Grub)
5. El gestor toma el control, busca el kernel, lo carga y ejecuta
6. Se inicia el kernel y monta la partición
7. Se ejecuta el init en RAM, que entrega el control al systemd para empezar el proceso de inicio estándar

Por defecto la máquina tiene el target **graphical.target.**

Para cambiar el target por defecto: **systemctl set-default <nuevo\_target>.**

Ya que no vamos a usar el entorno gráfico, pondremos **multi-user.target** como target por defecto.

Para ver el target por defecto: **systemctl get-default**

Para ver los targets en memoria: **systemctl list-units --type=target**

Para ver los targets instalados: **systemctl list-unit-files --type=target**

Estados: enabled, disabled, static, masked…

Para ver los service en memoria: **systemctl list-units --type=service**

Para ver los service instalados: **systemctl list-unit-files --type=service**

Existen otros tipos de unidad: timers, mount, socket…

Para ver todas las unidades: **systemctl list-units -t help**

1. *Determine los tiempos aproximados de botado de su kernel y del userspace. Obtenga la relación de los tiempos de ejecución de los services de su sistema.*

**systemd-analyze**: Saca los tiempos de arranque userspace + kernel

**systemd-analyze blame**: Saca relación de tiempos de ejecución de los services

1. *Investigue si alguno de los servicios del sistema falla. Pruebe algunas de las opciones del sistema de registro journald. Obtenga toda la información journald referente al proceso de botado de la máquina. ¿Qué hace el systemd-timesyncd?*

Para ver los servicios que fallan al arrancar el sistema: **systemctl list-unit-files –state=failed**

Para obtener información del arranque actual: **journalctl -b**

**systemd-timesyncd:** servicio encargado de sincronizar el reloj a través de la red

1. *Identifique y cambie los principales parámetros de su segundo “interface” de red (ens34). Configure un segundo “interface” lógico. Al terminar, déjelo como estaba.*

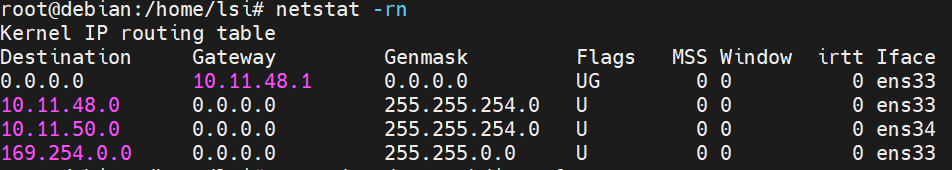
Para modificar el segundo interfaz de red: **nano /etc/network/interfaces.**

Para configurar el segundo interfaz lógico:

**-ifconfig ens34 down**

**-ifconfig ens34:0 10.11.52.0 netmask 255.255.254.0**

**-ifconfig ens34 up**

1. *¿Qué rutas (routing) están definidas en su sistema? Incluya una nueva ruta estática a una determinada red.*

Para ver las rutas definidas en el sistema: **netstat -rn // route -n**

Añadir ruta: **route add -net <destination> netmask <netmask> gw <gateway> dev <iface>**

Borrar ruta: **route del -net <destination> netmask <netmask> gw <gateway> dev <iface>**

Opción -p para mantener al reiniciar el dispositivo

route add -net [192.168.1.0](https://192.168.1.0/) netmask [255.255.255.0](https://255.255.255.0/) dev ens34

ip route add 10.11.52.0/23 via 10.11.50.1

ip route del 10.11.52.0/23 via 10.11.50.1

1. *En el apartado d) se ha familiarizado con los services que corren en su sistema. ¿Son necesarios todos ellos? Si identifica servicios no necesarios, proceda adecuadamente. Una limpieza no le vendrá mal a su equipo, tanto desde el punto de vista de la seguridad, como del rendimiento.*

Para ver los service instalados: **systemctl list-unit-files --type=service**

Para ver los activos: **systemctl list-unit-files --type=service --state=enabled**

Para desactivar: **systemctl disable <service> // systemctl mask <service>**

Services desactivados:

- bluetooth.service (Bluetooth)

- avahi-daemon.service (Busca conexiones con otras máquinas)

- NetworkManager.service (Gestiona comunicaciones automáticamente)

*-* ModemManager.service (Lineas G)

- open-vm-tools.service (Vmware)

- accounts-daemon.service (Obtiene y manipula info de la cuenta de usuario)

- wpa\_supplicant.service (sin Wifi no es útil)

- switcheroo-control.service (relacionada con GPUs)

- vgauth.service (autenticación en Vmware)

- plymouth.service (carga animación al inicio)

- anacron.service (teniendo cron es eliminable)

- cups-browsed.service (relacionado con impresoras)

- cups.service (impresoras)

- unattended-upgrades.service (actualizaciones automáticas)

- apparmor.service (controla gestión de recursos a programas)

- ssa.service (solo es usado en el primer encendido de la máquina)

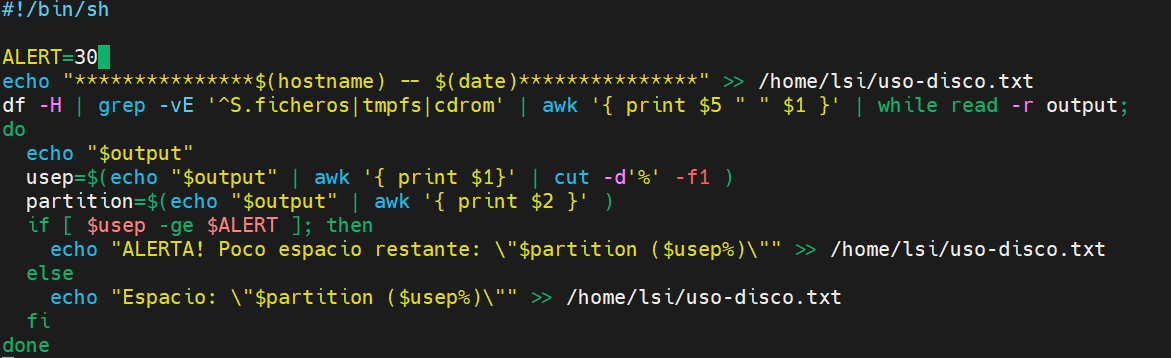
- keyboard-setup.service (teclado)

- e2scrub\_reap.service

- udisks2.service (relacionado con ruta archivos y con graphical.target)

1. *Diseñe y configure un pequeño “script” y defina la correspondiente unidad de tipo service para que se ejecute en el proceso de botado de su máquina.*

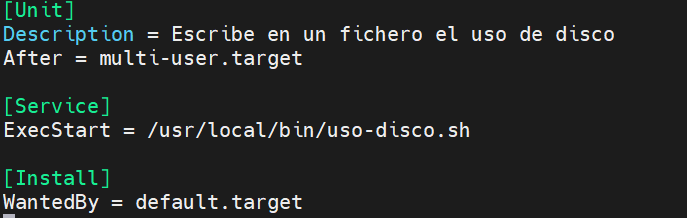
Creamos el script en el directorio /usr/local/bin (directorio para ejecutables no manejados por el paquete de distribución)



El servicio lo creamos en /lib/systemd/system

/lib/systemd/system/: ahí está el conjunto de targets y services del sistema

/etc/systemd/system/ -> aquí hay enlaces simbólicos a los de /lib/systemd/system/



* Description: descripción del servicio (systemctl status)
* After: momento de ejecución. Una vez la red está activa (network.target)
* ExecStart: ruta al script
* WantedBy: target en el que se instala

Cambiamos permisos del script: **chmod 744 uso-disco.sh**

Cambiamos permisos del servicio: **chmod 664 uso-disco.service**

Rehacemos el árbol: **systemctl daemon-reload**

Habilitamos servicio**: systemctl enable uso-disco.service**

1. *Identifique las conexiones de red abiertas a y desde su equipo.*

Para ver las conexiones abiertas: **netstat -neta**

Para mostrar los sockets**: netstat -a**

1. *Nuestro sistema es el encargado de gestionar la CPU, memoria, red, etc., como soporte a los datos y procesos. Monitorice en “tiempo real” la información relevante de los procesos del sistema y los recursos consumidos. Monitorice en “tiempo real” las conexiones de su sistema.*

Información tiempo real de procesos y recursos: **systemd-cgtop // top**

Información tiempo real de conexiones: **netstat -netac**

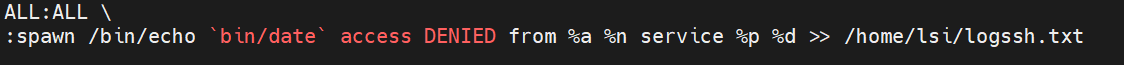
1. *Un primer nivel de filtrado de servicios lo constituyen los tcp-wrappers. Configure el tcp-wrapper de su sistema (basado en los ficheros hosts.allow y hosts.deny) para permitir conexiones SSH a un determinado conjunto de IPs y denegar al resto. ¿Qué política general de filtrado ha aplicado? ¿Es lo mismo el tcp-wrapper que un firewall? Procure en este proceso no perder conectividad con su máquina. No se olvide que trabaja contra ella en remoto por ssh.*

Tcp-wrapper no es lo mismo que firewall, ya que los tcp-wrappers trabajan a nivel de aplicación mientras los firewalls lo hacen a nivel de Sistema Operativo.

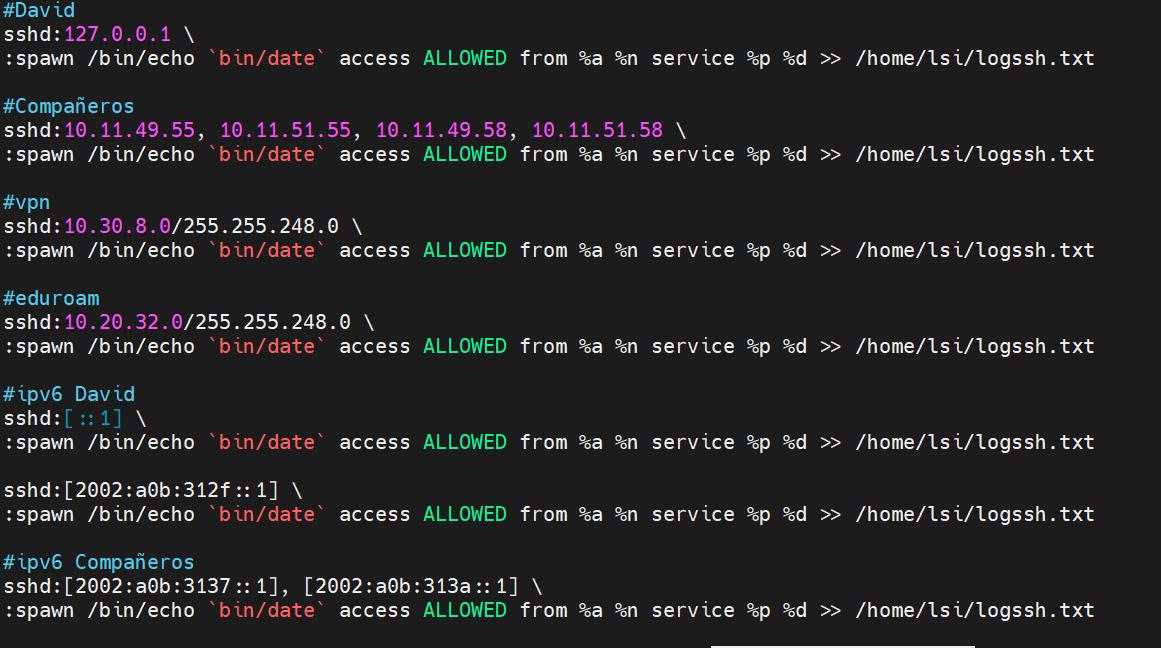
Para filtrar el acceso de servicios trabajaremos con dos archivos:

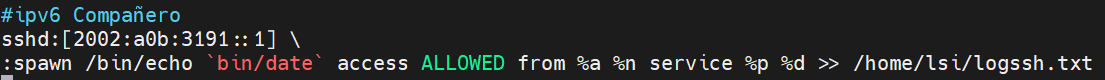
- /etc/hosts.allow

- /etc/hosts.deny

En/etc/hosts.deny 

En /etc/hosts.allow



**

1. *Existen múltiples paquetes para la gestión de logs (syslog, syslog-ng, rsyslog). Utilizando el rsyslog pruebe su sistema de log local.*

**rsyslog** es un programa de logging de mensajes que implementa el protocolo básico de syslog y le agrega filtros. Puede reenviar los logs via UDP y TCP (usando puerto 514).

Su configuración está en /etc/rsyslog.cong

En /var/log están los archivos de log

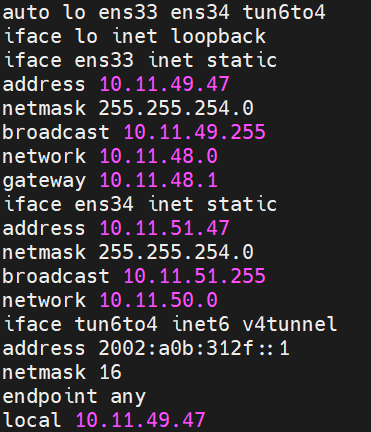
Enviamos un mensaje **logger -p mail.err “Prueba”** y buscamos en /var/log/mail.err

1. *Configure IPv6 6to4 y pruebe ping6 y ssh sobre dicho protocolo. ¿Qué hace su tcp-wrapper en las conexiones ssh en IPv6? Modifique su tcp-wapper siguiendo el criterio del apartado h). ¿Necesita IPv6? ¿Cómo se deshabilita IPv6 en su equipo?*

10.11.49.47 -> 2002:a0b:312f::1

Modificamos el hosts.allow para incluir la dirección ipv6 anterior.

Modificamos el fichero /etc/network/interfaces:



**ifup tun6to4**

Para hacer ssh: **ssh -6 lsi@ipv6**

Para hacer ping: **ping6 ipv6**

Para tumbar ipv6 tenemos que poner lo siguiente en el archivo /etc/sysctl.conf:

net.ipv6.conf.all.disable\_ipv6 = 1

net.ipv6.conf.default.disable\_ipv6 = 1

net.ipv6.conf.lo.disable\_ipv6 = 1

Para volver a levantarlo, comentamos estas líneas

Aplicar cambios: **sysctl -p**

1. *En colaboración con otro alumno de prácticas, configure un servidor y un cliente NTP.*

Yo:servidor

Instalar NTP: **sudo apt install ntp // sudo apt install ntpdate**

**Systemctl disable systemd-timesyncd.service**

En el servidor:

En el fichero **/etc/ntp.conf**:

server 127.127.1.1 minpoll 4

fudge 127.127.1.1 stratum 10

//Comentamos los otros

restrict 10.11.49.55 mask 255.255.255.255 noquery nopeer

restrict 127.127.1.1 mask 255.255.255.255 noserve nomodify

En **/etc/hosts.allow**:

ntpd: 10.11.49.55 ESTO NO SE HACE!!!!!!!!!!!!

En el cliente:

En el fichero **/etc/ntp.conf**:

server 10.11.49.47 minpoll 4

restrict 10.11.49.47 mask 255.255.255.255 noquery nopeer

Actualizamos: **systemctl restart ntp**

Probamos:

**date +%T -s 1**

**ntpdate -u 10.11.49.47**

1. *Cruzando los dos equipos anteriores, configure con rsyslog un servidor y un cliente de logs.*

Yo:Cliente

En el servidor:

En el fichero **/etc/rsyslog.conf**

# provides TCP syslog reception

module(load="imtcp")

input(type="imtcp" port="514")

$AllowedSender TCP, 127.0.0.1, 10.11.49.47

$template remote, "/var/log/%FROMHOST-IP%/%PROGRAMNAME%.log"

:inputname, isequal, "imtcp" ?remote

&stop

En **/etc/hosts.allow**:

rsyslogd: 10.11.49.47

En el cliente:

En el fichero /etc/rsyslog.conf

\*.\* action(

type="omfwd"

queue.type="LinkedList"

queue.filename="cola\_logs"

queue.maxFileSize=”1m”

action.resumeRetryCount="-1"

queue.saveonshutdown="on"

target="10.11.49.55"

port="514"

protocol="tcp"

)

**Systemctl stop syslog.socket syslog.service rsyslog.service**

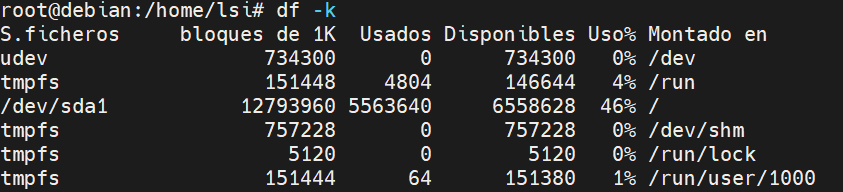
Actualizamos: **systemctl restart syslog.socket syslog.service ryslog.service**

1. *Haga todo tipo de propuestas sobre los siguientes aspectos.: ¿Qué problemas de seguridad identifica en los dos apartados anteriores? ¿Cómo podría solucionar los problemas identificados?*

NTP utiliza UDP por lo tanto los datagramas son fácilmente falsificables

Rsyslog es critico porque si se consigue acceder a el se puede obtener información muy valiosa. Además un ataque llenando de logs nuestra máquina podría hacerla caer. Para solventar esto podríamos utilizar un firewall

1. *En la plataforma de virtualización corren, entre otros equipos, más de 200 máquinas virtuales para LSI. Como los recursos son limitados, y el disco duro también, identifique todas aquellas acciones que pueda hacer para reducir el espacio de disco ocupado.*

**

Borrar paquetes en cache: **apt clean**

Borrar paquetes obsoletos: **apt autoclean**

Borrar paquetes con dependencias incumplidas: **apt autoremove**

Borrar manuales: **apt remove –purge man-db**

Ver kernel: **uname -a // uname -sr**

Listar kernels instalados: **dpkg -l | grep linux-image | awk ‘{print$2**}’

Eliminar imagen: **apt remove –purge <linux-image>**

Dejo instalada la 5.10.0-16-amd64 y la 4.19.0.21-amd64 y borro:

* Linux-image-4.19.0-9-amd64
* Linux-image-amd64

