**Práctica I.: Configuración básica**

**a) Configure** su máquina virtual de laboratorio con los datos proporcionados por el profesor. **Analice los ficheros** básicos de configuración

**/etc/network/interfaces:** configurar red.

auto lo ens33 ens34 (levanta interfaces)

iface lo inet loopback (inet (configura para ipv4))

iface ens33 inet static

address 10.10.102.236

netmask 255.255.255.0

broadcast 10.10.102.255

network 10.10.102.0

gateway 10.10.102.5

iface ens34 inet static

address 10.10.150.236

netmask 255.255.255.0

broadcast 10.10.150.255

network 10.10.150.0

**/etc/hosts**: resolución de nombres a ips

nombre ip

**/etc/resolv.conf**: se encarga de configurar el sistema de resolución de nombres

**nameserver** ip\_del\_servidor\_dns

**domain** dominio\_predeterminado

**search** dominios\_para\_completar nombres\_cortos

**/etc/nsswitch.conf**:

Fichero de configuracion de las BB de Conmutacion de los Servicios de nombres. Como una interfaz para acceder a datos de bd datos locales como usuarios contraseñas…

**base\_datos** especificacion\_servicio respuesta\_resultado\_busqueda

**/etc/apt/sources.list**: lista de repositorios

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

Una distro es una distribución software basado en el núcleo de Linux. Básicamente son distintos métodos de trabajo que emplean Linux, los cuales contienen diferentes procesos de instalación, diferentes paquetes, bibliotecas, herramientas, sistemas de ventanas, etc.

La distro y versión que tiene la máquina inicialmente es Debian 10. Con la actualización pasa a ser Debian 10 (nosotros ya teniamos debian 10 desde el principio). Para consultar esto podemos realizarlos con el comando: **lsb\_release -a.**

Para actualizar la máquina empleamos los siguientes comandos:

**apt get-update** -> para actualizar los repositorios

**apt get-upgrade** -> para realizar la actualización

Estos comandos necesitan permisos de superusuario.

**c)** 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?. Y a todo esto, ¿qué es un target?. ¿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?

En primer lugar, cuando pulsamos el botón de arranque, se activa la **BIOS** la cual reside en la memoria de sólo lectura. La BIOS se encarga de una inicialización básica del hardware.

A continuación pasa el control del sistema a un **cargador de arranque** almacenado en el almacenamiento persistente del dispositivo (se puede seleccionar la ubicación y prioridad de diferentes cargadores de arranque desde la configuración de la BIOS). El cargador de arranque busca el **kernel** en el dispositivo y lo invoca desde el disco. El kernel, extrae y ejecuta el disco RAM inicial, llamado **initrd**, que es un sistema de ficheros temporal en memoria (**systemd** puede ser usado para manejar servicios en initrd, similar al sistema real). Este busca la ruta del sistema de ficheros y una vez lo encuentra, lo monta. Seguidamente, initrd, pasa el control al gestor de usuarios del sistema, como systemd, almacenado en la ruta del sistema de ficheros, el cual se encarga de comprobar todo el hardware restante, montar todos los sistemas de ficheros necesarios, instalar los drivers que hagan falta e inicializar todos los servicios de configuración. Para realizar esto, systemd, utiliza varios pasos “**targets**” que agrupan determinadas unidades de arranque y sincronización de procesos. Así es como se logrará que se prepare todo y aparezca la pantalla de login.

Definimos target como una **unidad de systemd** que agrupa a otras unidades según las dependencias entre ellas con el objetivo de dar nombres estándar a puntos de sincronización utilizados en dependencias entre unidades. Los targets sustituyen a los **runlevels del antiguo** sistema de inicialización de SysVinit

Systemd **reemplaza** al init heredado de unix(SysVinit) y es un gestor de sistemas y servicios se basa en dependencias.

Target por defecto: **systemctl get-default** // **who -r**

graphical.target aka runlevel5.target (x compatibilidad con sysV)

Cambiar el target: **systemctl set-default <nivel\_ejecución>.target**

Targets del sistema: **systemctl list-units --type=target**

Mostrar servicios: **systemctl list-units --type=service (-all)**

**systemctl status (-all) // systemctl list-dependencies**

TIpos de unidades de systemd: service,socket,device,mount,automount,swap,target...

**STATES DE LAS UNIDADES**

**-ENABLED: se inicia en el arranque**

**-DISABLED: no lo inicia en el arranque**

**-STATIC: no se puede poner enabled, sólo se utiliza para indicar que realiza una acción única o que tiene dependencia de otra unidad y no debe de ejecutarse por si misma.**

**-MASKED:no puede ser iniciado por otro servicio (matarlo pero sin desinstalarlo), disabled sí puede ser iniciado por otro servicio**

systemctl ->snapshot del runtime

systemctl --all

systemctl list-units ( --type=target ) ->snapshot del runtime

systemctl list-units-files ->estado unidades en el arranque

multi-user.target es el que tenemos que poner (runlevel3)

**d)** 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**

Startup finished in 3.314s (kernel) + 20.444s (userspace) = 23.759s

graphical.target reached after 20.415s in userspace

Startup finished in 3.616s (kernel) + 7.285s (userspace) = 10.901s

multi-user.target reached after 7.263s in userspace

**systemd-analyze blame**

14.228s plymouth-quit-wait.service

3.047s ifupdown-pre.service

2.065s dev-sda1.device

1.772s networking.service

967ms udisks2.service

705ms ModemManager.service

626ms apparmor.service

625ms systemd-journald.service

524ms accounts-daemon.service

484ms systemd-logind.service

469ms rsyslog.service

436ms wpa\_supplicant.service

412ms systemd-udev-trigger.service

381ms systemd-udevd.service

368ms keyboard-setup.service

340ms systemd-timesyncd.service

321ms user@117.service

262ms run-vmblock\x2dfuse.mount

229ms upower.service

208ms gdm.service

145ms systemd-modules-load.service

138ms colord.service

135ms ssh.service

114ms user@1000.service

114ms systemd-tmpfiles-setup-dev.service

108ms systemd-sysusers.service

102ms systemd-update-utmp.service

99ms systemd-tmpfiles-clean.service

95ms dev-hugepages.mount

94ms dev-mqueue.mount

93ms sys-kernel-debug.mount

85ms systemd-remount-fs.service

83ms plymouth-start.service

82ms pppd-dns.service

79ms polkit.service

76ms systemd-sysctl.service

69ms systemd-tmpfiles-setup.service

67ms plymouth-read-write.service

66ms open-vm-tools.service

55ms console-setup.service

48ms systemd-random-seed.service

44ms kmod-static-nodes.service

39ms dev-sda5.swap

27ms switcheroo-control.service

24ms systemd-update-utmp-runlevel.service

22ms systemd-journal-flush.service

21ms user-runtime-dir@117.service

16ms sys-fs-fuse-connections.mount

15ms user-runtime-dir@1000.service

14ms systemd-user-sessions.service

9ms rtkit-daemon.service

**e)** 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 qué sirve el **systemd-cgls**?

Fallo algo: **systemctl --failed**

**Journald** recolecta mensajes del kernel y otras fuentes (bajo systemd (de hecho en los estatus aparecen datos de jourland)).

Almacena los logs en binario y son volátiles -> **journalctl**

(

-f (realtime)

-u (unidad systemd)

-k (kernel)

--list-boots (lista de los boots ids y timestamp)

-b [ID][±offset], (mensajes de un determinado boot)

-p PRIORIDAD (0 peores - 7)

)

**systemd-timesyncd:** sincronizar relojes

**systemd-cgls:** muestra el contenido de los grupos de control jerárquicos del kernel en un árbol. (.slice -> jerarquía ; .scope -> agrupan recursos)

**Grupos de control**: limitan, cuentan y aíslan los recursos usados de una colección de procesos.

f) Identifique y cambie los principales parámetros de su **segundo interface** de red (ens34). Configure un segundo **interface lógico**.

***ip de pedro: 10.10.150.236***

***ip de sergio: 10.10.150.233***

2º Interface: ens34

**ip addr** [xinterfaz]: interfaces físicas con sus ips etc.

i**p link set** xinterfaz **up/down**: levanta o tira la interfaz

**ifdown** xinterfaz [-a]: desactiva la interfaz si estaba activa

i**fup** xinterfaz [-a]: activa la interfaz si estaba desactiva

-a -> las interfaces definidas con auto

Añadimos nueva interfaz lógica ens34:0:

/etc/network/interfaces

auto lo ens33 ens34

iface lo inet loopback

iface ens33 inet static

address 10.10.102.233

netmask 255.255.255.0

broadcast 10.10.102.255

network 10.10.102.0

gateway 10.10.102.5

iface ens34 inet static

address 10.10.150.233

netmask 255.255.255.0

broadcast 10.10.150.255

network 10.10.150.0

iface ens34:0 inet static

address 10.10.150.254

netmask 255.255.255.0

broadcast 10.10.150.255

network 10.10.150.0

**systemctl restart NetworkManager.service:** reiniciar servicio de red (este servicio lo desactivamos en la primera clase, si se resetea no activamos la interfaz lógica, necesitamos hacer un ifup)

**systemctl restart networking.service:** reinicia servicio de red (no lo desactivamos)

**ifup** ens34:0

**ifdown** ens34:0

**ifconfig** eth1 "dirIP" : cambiar direccion ip.

**ifconfig** eth1 netmask "mascred" : cambiar mascara de red.

**ifconfig** eth1 broadcast "dirbro" : cambiar direccion broadcast.

**ifconfig** eth1 mtu "XX" : cambiar MTU (unidad maxima de transmision)

**ifquery --state:** interfaces que se levantaron con ifup

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

**ip route show (o list)** // **route -n :** muestra tabla de enrutamiento

Destination Gateway Genmask Flags Metric Ref Use Iface

0.0.0.0 10.10.102.5 0.0.0.0 UG 0 0 0 ens33

10.10.102.0 0.0.0.0 255.255.255.0 U 0 0 0 ens33

10.10.150.0 0.0.0.0 255.255.255.0 U 0 0 0 ens34

169.254.0.0 0.0.0.0 255.255.0.0 U 1000 0 0 ens33 (dir enlace local)

**route add ip/mask:** añade ruta a la tabla

ej -> route add -net 10.10.151.0 netmask 255.255.255.0 gw 10.10.150.254 dev ens34

-p -> si queremos que persista tras el reinicio del sistema

**route del:** elimina ruta de la tabla

ej -> route del -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.1 dev eth0

Podemos añadir una nueva ruta a la red 10.10.200.0 con gateway la máquina 10.10.150.254:

**route add -net 10.10.200.0 netmask 255.255.255.0 gw 10.10.150.254 dev ens34**

h) 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.

La limpieza queda limitada a los servicios que **no son dependientes de otros** ( sino al desactivarlos vuelven a ser activados).

Servicios que NO son dependientes de otros (mostrar sólo enable no static):

**systemctl list-unit-files --type=service | grep enable**

accounts-daemon.service -> permite a los programas el acceso a la información de las cuentas de usuario

anacron.service -> ejecuta aplicaciones y scripts en fechas y horas específicas

apparmor.service -> controla los recursos que se asignan a los programas

autovt@.service -> protege de accesos no autorizados

bluetooth.service -> gestiona servicio de bluetooth, no lo necesitamos

console-setup.service -> establece la fuente de la consola y carga el mapa de teclas

cron.service -> administrador regular de procesos

daemons dbus: encamina mensajes por un bus para comunicaciones entre procesos

dbus-fi.w1.wpa\_supplicant1.service -> bus del autenticador de claves wpa ?? se pude quitar

dbus-org.bluez.service -> bus relacionado con bluetooth, creo que se puede quitar

dbus-org.freedesktop.ModemManager1.service ->bus controla banda ancha móvil, se puede quitar

dbus-org.freedesktop.timesync1.service -> bus relacionado con la sincronización del tiempo

getty@.service -> protege de accesos no autorizados

keyboard-setup.service -> configura distribución del teclado, tarda mucho

ModemManager.service -> controla interfaces de banda ancha de móviles, al no usar móviles se puede quitar

networking.service -> permite levantar o tirar interfaces. No está corriendo en segundo plano. Normalmente todas las interfaces de red son manejadas por NetworkManager mientras networking.service no hace nada. En un servidor es bastante común asignar una dirección IP estática a una o más interfaces editando / etc / network / interfaces, y en este caso esas interfaces ya no son administradas por NetworkManager y son activadas o desactivadas por network.service.

open-vm-tools.service -> servicio de herramientas del VMware, pasa información entre el SO y el host. No se usa se pueden quitar

pppd-dns.service -> reliquia del pasado, servicio para internet de acceso telefónico rsyslog.service -> permite guardar syslogs

ssa.service -> genera el shell de VM de OpenBSD Secure shell ???

ssh.service -> OpenBSD Secure Shell Server

sshd.service -> mismo que ssh.service

switcheroo-control.service -> servicio de Switcheroo Control Proxy (d-bus)

syslog.service -> logging de mensajes del sistema y de programas

systemd-fsck-root.service ->checkean el sistema de archivos

systemd-timesyncd.service -> sincroniza el reloj del sistema con un servidor

udisks2.service -> checkea la ruta de archivos

unattended-upgrades.service -> descarga e instala actualizaciones de seguridad

vgauth.service -> servicio de herramientas VMWare para máquina virtual

wpa\_supplicant.service -> sólo necesario si usas interfaz wifi

systemctl stop accounts-daemon.service

systemctl disable accounts-daemon.service

systemctl stop bluetooth.service

systemctl disable bluetooth.service

systemctl stop dbus-fi.w1.wpa\_supplicant1.service

systemctl disable dbus-fi.w1.wpa\_supplicant1.service

systemctl stop dbus-org.bluez.service -> pone que no está loaded debe de ser por tirar el servicio de bluetooth porque esté está relacionado

systemctl stop dbus-org.freedesktop.ModemManager1.service

systemctl disable dbus-org.freedesktop.ModemManager1.service

systemctl stop ModemManager.service

systemctl disable ModemManager.service

systemctl stop open-vm-tools.service

systemctl disable open-vm-tools.service

systemctl stop pppd-dns.service

systemctl disable pppd-dns.service

systemctl stop vgauth.service

systemctl disable vgauth.service

systemctl stop wpa\_supplicant.service

systemctl disable wpa\_supplicant.service

i) 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.

/lib/systemd/system/storage\_notifyer.service

[Unit]

Description=Storage Notifyer creates a .txt with storage information

After=multi-user.target

[Service]

Type=simple

Restart=on-failure

User=lsi

ExecStart=/usr/bin/python3 /home/lsi/storage\_notifyer.py

StandardOutput=syslog

StandardError=syslog

[Install]

WantedBy=multi-user.target

import shutil

f = open("/home/lsi/storage.txt", "w")

total, used, free = shutil.disk\_usage("/")

#si tiene menos de 1gb

if free<2\*\*30:

f.write("ALERTA! Espacio disponible: %.2f GB\n" % (free/2\*\*30))

else:

f.write("Tienes suficiente espacio: %.2f GB\n" % (free/2\*\*30))

f.close()

After : establece el orden entre las dependencias.

Type: configura inicio de nuestro servicio.

Restart: por defecto systemd no reinicia tu servicio si cae.

WantedBy: target al que pertenece

Una vez creado el script y el servicio -> darle permisos para que el usuario lsi pueda ejecutarlos :

chmod 744 /lib/systemd/system/storage\_notifyer.service

chmod 744 storage\_notifyer.py

(rwx para user r r para el resto)

[**https://www.freedesktop.org/software/systemd/man/bootup.html**](https://www.freedesktop.org/software/systemd/man/bootup.html) **(sysinit.target)**

**sudo systemctl daemon-reload**: recarga suave,toma configuraciones cambiadas del sistema de archivos y genera árbol de dependencia

sudo systemctl start storage\_notifyer.service : iniciar servicio

sudo systemctl stop storage\_notifyer.service: para servicio

sudo systemctl enable storage\_notifyer.service: habilita servicio en el arranque

j) Identifique las conexiones de **red abierta** a y desde su equipo

**netstat -putona**

* ***p*** Muestra las conexiones para el protocolo especificado que puede ser TCP o UDP
* ***u*** Lista todos los puertos UDP
* ***t*** Lista todos los puertos TCP
* ***o*** Muestra los timers
* ***n*** Nos muestra el numero de puerto
* ***a*** Para visualizar todas las conexiones activas del sistema

k) 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**.

**ps**: Imprime los procesos en ejecución y estadísticas asociadas.

**top**: Porcentaje y tiempo de CPU, así como uso de memoria, de procesos e hilos.

**vmstat**: Utilización de la memoria virtual (VM) del sistema.

**free**: Consumo global de la VM.

**netstat -c:** imprime la info solicitada cada segundo.

l) Un primer nivel de filtrado de servicios los 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 Wrappers son listas de control de acceso, trabaja en capa 7. Cosas relevantes:

* Daemon tcpd: software que controla el acceso para los servicios de red.
* /etc/hosts.allow: Este archivo contiene los nombres de hosts que tienen permitido utilizar servicios de red. (Tiene prioridad en caso de conflicto)
* /etc/hosts.deny: Este archivo contiene los nombres de hosts que no pueden utilizar servicios de red

Comodines: ALL, LOCAL, UNKNOWN, KNOWN, EXCEPT;

DEMONIOS: EQUIPOS : deny/spawn/twist

spawn: registros de conexión

twist: sustituye el servicio -> honeypots

[ipv6]

/etc/hosts.allow

#Propia

sshd: 127.0.0.1, 10.10.102.236 :spawn (/bin/echo ‘[ %d ] conectado con %a’ >> /var/log/conexiones.log )

#Compañero

sshd: 10.10.102.233 :spawn (/bin/echo “[ %d ] conectado con %a” >> /var/log/conexiones.log )

#Compañero

sshd: 10.10.102.56 :spawn (/bin/echo “[ %d ] conectado con %a” >> /var/log/conexiones.log )

#Eduroam

sshd:10.20.32.0/255.255.252.0 :spawn (/bin/echo “[ %d ] conectado con %a” >> /var/log/conexiones.log )

#VPN

sshd:10.30.8.0/255.255.248.0 :spawn (/bin/echo “[ %d ] conectado con %a” >> /var/log/conexiones.log )

/etc/hosts.deny

ALL:ALL

:spawn (/bin/echo ‘%a intentando conectarse con [ %d ]’ >> /var/log/conexiones\_deny.log)

Un firewall trabaja a nivel del sistema operativo, no a nivel de aplicación. Con solo TCP wrappers no haces un firewall, deben estar los dos. Política todo cerrado menos ssh.

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

Syslog significa **protocolo de registro del sistema**.

El mensaje de syslog consta de tres partes: **PRI** (un valor de prioridad calculado), **HEADER** (con información de identificación) y **MSG** (el mensaje en sí).

PRI = (Valor de recurso \* 8) + Valor de gravedad -> num menores-mayor prioridad

valor recurso: tipo de mensaje o qué sistema generó el evento.

valor gravedad: debug(=7),info,notice,..,emerg(lopeor = 0)

Rsyslog es un demonio encargado de recolectar los mensajes de servicios que provienen de aplicaciones y el núcleo para luego distribuirlos en archivos de registros

PROBANDO LOG LOCAL

logger: comando para registrar entradas en el log del sistema

-p -> prioridad especificada***<valor\_recurso.valor\_gravedad>***mail.err, user.notice …

**logger -p mail.err “Pueba”**

Se crean 4 archivos en /var/log : mail.err, mail.info, mail.log, mail.warn } configurado así en /etc/rsyslog.conf

**cat /var/log/mail.err**

n) 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?

Enviar paquetes IPV6 encapsulados en IPV4

2002::/16:

ELEMENTOS: 6to4 host(destinatario (2002)), 6to4 router(dual-stack)

10.10.102.233 -> segmento 2002:A0A:66EA/48 -> 1ª dir **2002:A0A:66E9::**1/128

10.10.102.236 -> segmento 2002:A0A:66EC/48 -> 1ª dir **2002:A0A:66EC**::1/128

**/etc/network/interfaces**

auto tun6to4

iface tun6to4 inet6 v4tunnel

address 2002:A0A:66EC::1

netmask 16

endpoint any

gateway ::192.88.99.1

local 10.10.102.236

*\*gateway 192.88.99.1*-> dirección reservada para routers que enrutan datagramas entre ipv4 y ipv6. Dudas de si ponerla porque si hacemos un ping a dicha ip no nos responde, y debería ya que se trata de la gateway.

*\*endpoint*-> dirección del otro extremo del túnel.

**ifup tun6to4**

**ping6 ipv6.google.com** -> deja enviar paquetes (está bien), no los recibe porque el firewall lo bloquea.

Si intentamos hacer un ssh en ipv6:

ssh lsi@2002:a0a:66e9::1

ssh\_exchange\_identification: read: Connection reset by peer

Tal como tenemos configurado el tcp wrapper no permite ssh en ipv6.

Añadimos a hosts.allow

**/etc/hosts.allow**

#Local ipv6

sshd:[::1] :spawn (/bin/echo ‘[ %d ] conectado con %a’ >> /var/log/conexiones.log )

#Propia IPv6

sshd:[2002:a0a:665a::1] :spawn (/bin/echo ‘[ %d ] conectado con %a’ >> /var/log/conexiones.log )

#Comp IPv6

sshd:[2002:A0A:66EC::1] :spawn (/bin/echo ‘[ %d ] conectado con %a’ >> /var/log/conexiones.log )

Para **desactivar el protocolo de ipv6**, añadimos:

Si queremos desactivarlo sin reinicio: en **/etc/sysctl.conf**

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

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

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

**sudo sysctl -p // reinicio**

**sysctl->** se utiliza para establecer algunos parámetros del kernel y que estos se mantengan entre sucesivos arranques del sistema, es decir, que los cambios sean persistentes.

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

Network TIme Protocol

apt install ntp ntpdate

systemctl stop/disable systemd-timesyncd.service

Archivo a configurar: **/etc/ntp.conf**

***stratum***

***uno mas tocho para que yo me actualice***

***Servidor:***

server 127.127.1.1 minpoll 4

fudge 127.127.1.1 stratum 10

restrict 10.10.102.233 mask 255.255.255.255 noquery nopeer

***Cliente:***

server 10.10.102.236 minpoll 4 prefer

fudge 10.10.102.236 stratum 1

restrict 10.10.102.236 mask 255.255.255.255 noquery nopeer

**Restrict**: define las reglas que permiten a los clientes conectarse a su servicio.

Opciones de acceso de control más comunes:

**ignore:** deniega todos los paquetes

**nomodify:** no permite modificar la configuración de ntpd a determinada ip/subred

**noserve**: deniega todos los paquetes menos los ntpq y ntpdc

**notrust**: ignora todos los paquetes ntp que no estén autenticados

**noquery**: no permite conocer el estado ntpd, no permite obtener info del sistema del servidor

**nopeer**: no permite que se produzca una red de pares, P2P.

Tenemos que esperar 10 minutillos

**ntpdate -u ip**

**ntpq -p**

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

**/etc/rsyslog.conf**

**Servidor:**

module(load="imtcp")

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

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

\*.\* ?RemoteLogs

**Cliente:**

$WorkDirectory /rsyslog/work

$ActionQueueType LinkedList

$ActionQueueFileName example\_fwd

$ActionResumeRetryCount -1

$ActionQueueSaveOnShutdown on

\*.\* @@10.10.102.233:514

c)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?

Rsyslog no pide **autenticación** y te pueden petar el disco, se puede restringir ips que puedan mandar: **$AllowedSender TCP, 127.0.0.1, 172.16.1.0/28, 10.0.1.0/29** o firewall, de todas formas se puede **spoofear** la ip. Lo mejor sería meter una forma de autenticación, Algo parecido pasa con NTP. En cuanto al flujo, **nada es cifrado** y podría o **modificarse** o ir **escuchando** todo. En el caso de RSYSLOG es muy peligroso dado que da información muy útil sobre nuestro sistema y si llegan a hackear el servidor tiene info de toda la red. En NTP mucho negacion de servicios (noquery soluciona algo pero ññ)

Resumen **AUTENTICAR**, **CIFRAR**.

d) 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.

Podemos realizar un **apt clean** para borrar todos los paquetes de la caché ya que el gestor guarda una lista de paquetes para no tener que descargarlos de nuevo.

También podemos realizar un **apt autoclean** el cual elimina los paquetes que están obsoletos.

Un **apt autoremove** paraeliminar los paquetes que estén huérfanos con dependencias incumplidas y los cuales no va a necesitar el sistema. La opción **--purge** para purgarlos sino no lo realiza.

Eliminar imágenes de kernels antiguos:

**uname -sr:** muestra la versión del kernel que está corriendo actualmente en nuestro sistema.

**dpkg -l | grep linux-image | awk '{print$2}' :** lista todos los kernels que tenemos instalados.

**apt remove --purge *linux-image*:** elimina la imagen que le indiquemos.

**update-grub2**: al finalizar conviene actualizar el grub y realizar un reinicio de la máquina.

También conviene dejar alguna imagen antigua por si falla alguna actualización, poder volver a dicha imagen.

Podemos eliminar todos los ficheros de log y también los idiomas que vienen instalados por defecto en la máquina.

Si vamos muy apurados de espacio también se puede realizar un borrado de los manuales del sistema.

**apt remove --purge man-db :** elimina dichos manuales.

**rm -r /usr/share/man:** forma bruta

DUDAS: puerta d enlace en tun6to4