P1

Comentario	
Nota	2.25/4
Tags	Acabado

> **a) 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.)

Desabilitamos el servicio avahi para hacer la iPv6 pública NO accesible.

→Avahi es un sistema que facilita el descubrimiento de servicios de una red local a través de los protocolos DNS.

```
systemctl stop avahi-daemon.socket
systemctl stop avahi-daemon.service
systemctl stop NetworkManager.service
systemctl disable avahi-daemon.socket
systemctl disable avahi-daemon.service
systemctl disable NetworkManager.service
```

Luego entramos en */etc/network/interfaces* con el comando:

```
nano /etc/network/interfaces
```

Para asi poder editar el fichero de **interfaces**

→Configuración de las interfaces de red (nombre y configuración (en caso de ser estática se indica la dirección IP, la de la red en la que está conectada y la de broadcast, junto a la máscara de subred y la puerta de enlace)

```
auto lo ens33 ens34 tun6to4
iface lo inet loopback
iface ens33 inet static

address 10.11.48.48
```

```
netmask 255.255.254.0
broadcast 10.11.49.255
network 10.11.48.0
gateway 10.11.48.1

iface ens34 inet static
address 10.11.50.48
netmask 255.255.254.0

iface tun6to4 inet6 v4tunnel
address 2002:a0b:3030::1
netmask 16
endpoint any
local 10.11.48.48
```

Ejecutamos:

```
systemctl restart networking.service
```

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

La **distro** es la distribución GNU/Linux.

Con el comando vemos la información específica sobre el LSB (*Linux Standard Base*) y la distribución.

```
lsb_release -a
```

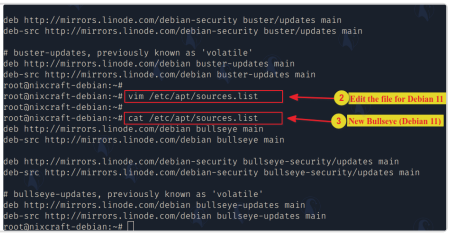
```
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 10 (buster)
Release:        10
Codename:       buster
```

Hay que actualizar a **Debian 11** siguiendo estas instrucciones:

How to upgrade Debian 10 to Debian 11 Bullseye using the CLI

I have Debian 10.10 installed on the AWS EC2 and Linode server. How do I upgrade Debian 10 Buster to Debian 11 Bullseye using the apt command or apt-get command safely? How can I upgrade Debian 10 to Debian 11 using

https://www.cyberciti.biz/faq/update-upgrade-debian-10-to-debian-11-bullseye/



> **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?. ¿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 completa de arranque (debe ser ejecutada desde root):

```
journalctl -b
```

1. . El hardware carga la BIOS. Esta hace comprobaciones de hardware y carga el MBR(Búsqueda de sector de arranque) que se encuentra en el primer sector del disco. Entonces la BIOS le pasa el control.
2. El MBR carga y ejecuta el GRUB (cargador de arranque) ya que busca el nucleo en el DD lo carga y ejecuta.
3. El GRUB elige el SO que se iniciará y carga el Kernel. Es decir se inicializa el núcleo y monta la partición raíz.
4. Se ejecuta el proceso init o initramfs, fichero raíz en ram. Init configura todos los serviciox y estructuras que no sean del SO. Initramfs entraga el control a systemd.
5. Systemd inicia grupos de procesos y servicios en paralelo organizados en targets. Una vez finalice la inicialización, el SO quedará iniciado.

Target:

Archivo de unidad especial que describe el estado del sistema o punto de sincronización. Los targets agrupan un conjunto de servicios que definen el estado de mi sistema.

¿Qué target por
defecto tiene su máquina?.

```
systemctl get-default
```

En nuestro caso tenemos *graphical.target*

¿Cómo podría cambiar el target de arranque?

Tendríamos que romper el enlace simbólico y crear uno nuevo. Se hace con el siguiente comando:

```
systemctl set-default multi-user.targets
```

Qué targets tiene su sistema y en qué estado se encuentran?

```
systemctl list-units --type=target
```

Y los services?

```
systemctl list-units --type=services
```

Relación de servicios de su sistema y su estado.

```
systemctl list-units - -type=service -all
systemctl list-unit-files
```

```
systemctl list-units ---type=service | grep enabled
```

¿Qué otro tipo de unidades existen?.

Sockets, targets, timers, ...)

> **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
systemd-analyze blame //para cada servicio
```

> **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?.**

→Miramos los servicios que fallan:

`systemctl –state=failed`

→Opciones del registro journald:

`journalctl`:muestra todos los logs del sistema.

`journalctl -f`: muestra los mensajes que se van generando en tiempo real.

`journalctl -r:` revertir la salida para que las entradas más recientes se muestren primero.

`journalctl -b:` muestra los logs del botado de la maquina.

→El comando `systemd-timesyncd` es un demon que se ha añadido para sincronizar el reloj del sistema a través de la red, utilizando el protocolo SNTP.

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

→ Con `ip a s` vemos la informacion relacionada con los interfaces.

→ Nosotras ya tenemos configurada una segunda interfaz de red con ens34.

Para crear ahora un interfaz lógico ejecutamos:

```
ifconfig ens34:0 10.11.48.253 netmask 255.255.254.0
```

Para borrar lo que acabamos de hacer:

`Ip a ls ens34 | grep inet` **(grep es para filtrar) : con esto vemos las subredes asociadas a ens34**

`Ip a del 10.11.48.253/23 dev ens34:0` **: asi borramos la ens34:0**

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

Este comando los muestra las tablas de enrutamiento del sistema.

```
route -n
```

Para ver las rutas (routing) hacemos:

```
ip route show
```

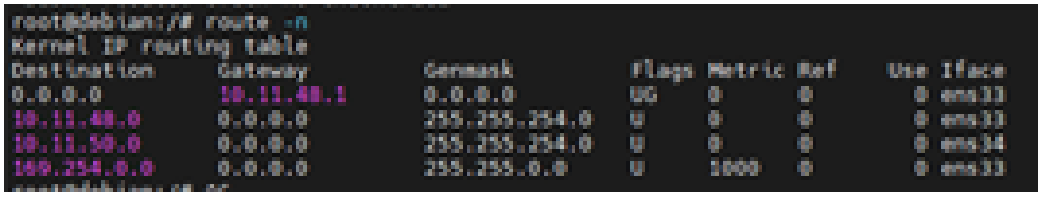
Para añadir rutas estáticas:

```
route add -net <IP> netmask <netmask> gw <gateaway>
route add -net <Red> netmask <netmask> dev <dispositivo>
route add <IP> <ens34> #esto te la añade con mascara por defecto
```

Ejemplos reales:

```
route add -net 10.11.120.0 netmask 255.255.254.0 gw 10.11.48.1
```

Salida por pantalla:




```
default via 10.11.48.1 dev ens33 onlink
10.11.48.0/23 dev ens33 proto kernel scope link src 10.11.48.48
10.11.50.0/23 dev ens34 proto kernel scope link src 10.11.50.48
10.11.120.0/23 via 10.11.48.1 dev ens33
169.254.0.0/16 dev ens33 scope link metric 1000
```

Hacemos **ping** para comprobar que la red de la Ip que añadimos esta activa:

ping IP → si no recive nada, no está activa.

Comprobamos que aparece la nueva ruta en la tabla de enrutamiento y luego borramos lo hecho:

```
route -n
route del -net 10.11.120.0 netmask 255.255.254.0 gw 10.11.48.1
```

▼ **PRUEBA CASO DE USO REAL**

Comprobamos que ***ens34*** no tiene salida a internet:

1. Cogemos una ip de una página web de internet.
2. Bajamos html con `wget marca.com`
3. La añadimos a la route table como ens34.
4. Volvemos a intentar bajar el html: ya no nos dejará.e

> **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.**

En el aparatado a) ya hemos tirado 2 servicios:

→ avahi-daemon.service:Se supone que avahi-daemon.service proporciona detección de red sin configuración y hace que sea muy fácil encontrar impresoras y otros hosts en su red. Siempre lo desactivo y no me lo pierdo
→ NetworkManager.service: es un servicio que administra las conexiones y los informes de cambios en la red y una aplicación gráfica de escritorio que permite al usuario manipular las conexiones de red. El subprograma nmcli proporciona una funcionalidad similar en la línea de comandos. GNOME está obsoleto, por lo que podremos borrar el servicio

1. Miramos el tiempo de arranque y consumo de los procesos que se ejecutan:

```
systemd-analyze blame
systemctl list-unit-files --state=enabled
```

2. Nos centramos en los primeros servicios de la lista que son los que más consumen:

```
systemd-analyze blame | head
```

3. Comprobamos qué servicios no son necesarios.
4. Eliminamos los servicios escogidos:

```
systemctl stop nombre.service
systemctl disable nombre.service
```

Los servicios eliminados son:

Bluetooth : porque no lo usamos.

ModemManager.service : Controla cosas de 3G y 4G, como no tenemos banda ancha movil no lo necesitamos.

pppd-dns.service : "es una reliquia del pasado oscuro" wtf xd Para internet de acceso telefonico.

wpa_supplicant.service: contola el roaming y la asociación/autenticación IEEE 802.11 del controlador wland.

switcheroo-control.service:es un servicio de D-Bus para verificar la disponibilidad de GPU dual.

ssa.service:

→ Hacer apt autoremove

> **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.**

Crear un fichero con y en:

```
nano /usr/local/bin/scripti
```

AÑADIR:

```
#!/bin/bash
echo hola
logger -p mail.err "hola otra vez"
```

Darle permisos con:

```
chmod +x /usr/local/bin/scripti
```

Crear el servicio y añadir:

```
[Unit]
Description=Lanza servicio
After=syslog.target

[Service]
Type=forking
ExecStart=/usr/local/sbin/scripti

[Install]
WantedBy=multi-user.target
```

Este unit file iniciará el script indicado con la opción ExecStart bajo el usuario usuario.

Utilizamos el servicio forking, que es un servicio que lanza el proceso hijo que se convierte en el principal.

Una vez este todo configurado debemos recargar.

```
systemctl daemon-reload
```

Para lanzar el servicio simplemente hacemos:

```
systemctl enable scripti.service
systemctl start scripti.service
```

Para habilitarlo al inicio y que se lance automáticamente:

`systemctl enable scripti.service`

Y finalmente para ver el estado podemos hacer:

`systemctl status scripti.service`

j)Identifique las conexiones de red abiertas a y desde su equipo.

```
netstat -putona
netstat -l #solo son puertos escuchando.
netstat -neta
```

Para todas los comandos sale una lista de información.

• **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.**

Para ver los recursos consumidos por los procesos del sistema usamos:

```
top
```

Este comando muestra datos con el número de usuarios conectados a la maquina, uso cpu, procesos....

→Otro comando sobre el uso de cpu, mem, entrada,salida... es *systemd-cgtop.*

Para monitorizar en tiempo real las conexiones de sus sistema, como bien sabemos tenemos que usar el comando **netsats** que es el orientado a conexiones:

```
netstat -c
netstat --continuous
```

> **l) Un primer nivel de filtrado de servicios los constituyen los tcp-wrappers. Configure el tcpwrapper 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.**

Los ***TCP Wrappers*** son listas de control de acceso (ACL – access control list) basadas en hosts, y utilizadas para filtrar accesos de red a los servicios locales.

→**Configuración:**

Acceder a `/etc/hosts.allow` y añadir:

(para permitir mi acceso por ssh, el de mi compi y a través de VPN)

Este archivo contiene los nombres de hosts que tienen permitido utilizar servicios de red y el spawn.

```
#Mi IP
sshd: 10.11.48.45 :spawn /bin/echo "($(date)|MIIP) from %a service %d >>/var/log/conexion_ssh.txt
sshd: 10.11.50.45 :spawn /bin/echo "($(date)|MIIP34) from %a service %d >>/var/log/conexion_ssh.txt
#IP compi
sshd: 10.11.48.48 :spawn /bin/echo "($(date)|COMPIIP) from %a service %d >>/var/log/conexion_ssh.txt
sshd: 10.11.50.48  :spawn /bin/echo "($(date)|COMPIIP34) from %a service %d >>/var/log/conexion_ssh.txt
#VPN
sshd: 10.30.:spawn /bin/echo "($(date)|VPN) from %a service %d >>/var/log/conexion_ssh.txt
```

En `/etc/hosts.deny` poner una linea con:

(Bloqueando asi a todas las ips que no estén en hosts allow.)

Este archivo contiene los nombres de hosts que no pueden utilizar servicios de red y el twist.

```
ALL:ALL : twist /bin/echo "%h has been banned from this server!"
```

¿Qué política general de filtrado ha aplicado?

Una política muy restrictiva de bloqueo total y que deja el paso solo a las conexiones que nosotros indiquemos en el `hosts.allow`

¿Es lo mismo el tcp-wrapper que un firewall?.

Los TCP Wrappers suelen utilizarse para filtrar direcciones ip y hostnames. Y un firewall es un dispositivo de hardware o un software que nos permite gestionar y filtrar la totalidad de trafico entrante y saliente que hay entre 2 redes u ordenadores de una misma red. Los wrappers tambien funcionan a nivel de app y los firewalls no.

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

```
logger -p mail.err "holi"
```

Este comando creará un log con el mensaje holi, de tipo mail. Que podremos comprobar si está o no en el fichero `cat var/log/mail.err`

→ `cat /var/log/syslog:` si accedemos aqui muestra todos los mensajes.(desde root) "

→ `cat /var/log/mail.err`: muestra los mensajes del recurso MAIL.

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?

→Añadir al fichero interfaces:

```
auto lo eth0 eth1 tun6to4 (añadir tun6to4)
iface tun6to4 inet6 v4tunnel
address 2002:a0b:3030::1
netmask 16
endpoint any
local 10.11.48.48
```

▼ Entrar en `/etc/hosts.allow`

▼ Añadir a ese fichero las lineas:

#MIIP

sshd: [2002:a0b:3030::1] :spawn /bin/echo "($(date)|MIPV6) from %a service %d >>/var/log/conexion_ssh.txt

#MI COMPI

sshd: [2002:a0b:302d::1] :spawn /bin/echo "($(date)|MIPV6) from %a service %d >>/var/log/conexion_ssh.txt

¿Qué hace su tcp-wrapper en las conexiones ssh en IPv6?

Mi tcp-wrapper para poder funcionar en IPv6 tenemos que añadirle las lineas escritas anteriormente en el hosts.allow, para así permitir las conexiones. De la otra manera denegará todo tipo de conexiones ipv6.

¿Necesita IPv6?. ¿Cómo se deshabilita IPv6 en su equipo?

No, actualmente se sigue utilizando IPv4 por lo que podríamos deshabilitar IPv6 sin ningún tipo de riesgo para ello.

En el fichero `/etc/sysctl.conf` poner:

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

→Para comprobar que funciona hacemos ping6 y ssh a mi compi:

```
ifup tun6to4
ping6 2002:a0b:302d::1
ssh lsi@2002:a0b:302d::1 o ssh -6 2002:a0b:302d::1
```

PARTE CONJUNTA:

> ### a)En colaboración con otro alumno de prácticas, configure un servidor y un cliente NTP. (yo soy servidor)

ANTES DE NADA:

→Instalar paquete NTP y asegurarnos de nuestra zona horaria(Europe/Madrid):

```
apt-get update
apt-get install ntp ntpdate
timedatectl
```

Entrar en el fichero /etc/ntp.conf → allí ya nos encontramos con las instrucciones tanto para ser cliente como para ser servidor.

#CLIENTE:

→ IMPORTANTE: Comentar los pools.

```
server 10.11.48.45 prefer
```

Y en la parte de restricciones añadiremos:
`restrtict 10.11.48.45`

#SERVIDOR:

1. Comentamos los pools.
2. Añadimos en la parte de servidor:
    a. `#server 127.127.1.1 minpoll 4  prefer`
    `#fudge 127.127.1.1 stratum 10`
3. Añadimos en la parte de restricciones de servidor:
    a. `#restrict 10.11.48.45 mask 255.255.255.255 noquery nopeer`
    `#restrict 127.127.1.1 mask 255.255.255.255 noserve nomodify`

CONFIGURACIONES Y COMPROBACIONES:

1. Reiniciar servicio con: `systemctl restart ntp`
2. Comprobar que funciona con: `ntpq -p`

FUNCIONAMIENTO:

→Servidor tira abajo el servicio:

```
systemctl stop system-tymesyncd.service
```

→Guardamos cambios:

```
systemctl restart ntp.
```

→El servidor cambia la fecha con:

```
date -- set “2017-08-02 12:45”
```

→El cliente solicita la fecha y hora via NTP con:

```
ntpdate -u 10.11.48.45
```

Debería ser la fecha y hora puesta por el servidor anteriormente, tambien lo podemos comprobar con `date o timedatectl.`

La fecha del cliente se cambiaria y podemos vovler a ponerla bien con `date --set`

> ## b) Cruzando los dos equipos anteriores, configure con rsyslog un servidor y un cliente de logs.(yo soy cliente)

Entrar en el fichero de `/etc/rsyslog.conf`

EL SERVIDOR:

→Descomenta las lineas de TCP para dejarnos recibir mensajes de otras ips

y añadimos en RULES la linea:

```
:fromhost-ip,isequal,"10.11.50.48" /var/log/messages
```

Que hace que si le llega un mensaje por la IP puesta, lo guarde en `messages`.

EL CLIENTE:

→ Ponemos esto debajo de #RULES#

```
$ActionQueueType LinkedList
$ActionQueueFileName Cliente_logs
$ActionResumeRetryCount -1
$ActionQueueSaveOnShutdown on
*.*@@10.11.48.48:514
```

INFO:

Información sobre mandar el mensaje a un solo Servidor, y por qué se uso lo de Queue:

- Estamos configurando la cola para que actúe como buffer y almacene los mensajes en caso de que el servidor se caiga
- `$ActionQueueType` permite que una Lista enlazada en la memoria de la cola.
- `$ActionFileName` define el almacenamiento de disco, en este caso se guardan los ficheros backup en el directorio `/var/lib/rsyslog/` con el prefijo Cliente_logs.
- `$ActionResumeRetryCount -1` setting prevents rsyslog from dropping messages when retrying to connect if server is not responding.
- enabled `$ActionQueueSaveOnShutdown` saves in-memory data if rsyslog shuts down,
- the last line forwards all received messages to the logging server, port specification is optional.

With the above configuration, rsyslog keeps messages in memory if the remote server is not reachable. A file on disk is created only if rsyslog runs out of the configured memory queue space or needs to shut down, which benefits the system performance.

Para comprobar el funcionamiento:

El cliente hace `logger "<mensaje>"`

y el servidor busca el mensaje en `cat /var/log/messages`

PROBAR FUNCIONAMIENTO DE LA COLA:

→El servidor tira abajo los siguientes servicios:

```
systemctl stop syslog.socket
systemctl stop rsyslog.service
systemctl stop syslog.service
```

→El cliente envia los mensajes otra vez con `logger`.

→ El servidor levanta los servicios:

```
systemctl start syslog.socket
systemctl start rsyslog.service
systemctl start syslog.service
```

y a continuación mira en `var/log/messages (cat ver/log/messages)` si se encuentran alli los mensajes enviados por el cliente durante el parón de conexión.

> ## 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?

Qué problemas de seguridad identifica en los dos apartados anteriores?.

Apartado A
En el apartado A algunos de los problemas que podríamos tener serían, los siguientes, las conexiones NTP, utilizan el protocolo **UDP que está siendo muy usado para los ataques de denegación de servicio,** es decir estos ataques se basan en la **falsificación del**

tráfico IP (spoofing). Otro posible problema es la posibilidad de que un equipo genere tráfico con una IP que no le corresponde y que éste llegue hasta internet. Recordemos también que con UDP podemos perder paquetes muy facilmente.
En resumen lo que puede llegar a pasar sería que en las versiones antiguas de NTP algunas redes tienen además un servicio de monitorización que permite a los administradores recopilar una lista de los 600 hosts que se han conectado al servidor. Los ciberatacantes aprovechan esta característica haciendo un ataque reflejo: envían un paquete con una dirección IP falsa mediante la que obtienen la lista. Después, lo amplifican realizando un ataque de denegación de servicio (DDoS) que puede dejar en fuera de juego temporalmente la conexión de todas las direcciones de hosts de la lista.

Apartado B

En el apartado B algunos de los problemas que tendríamos serían que si **hackean el servidor de logs, pueden ver todos los logs de la red.** También tenemos la posibilidad de un ataque DDoS, como se explicó en la sección anterior No hay autenticación, permitiendo que cualquiera envíe logs al servidor, no existe confidencialidad de los datos enviados, no hay integridad de los datos (alguien podría modificarlos en el camino – MITM)…
Aunque probablemente su problema más grave es la pobre seguridad que provee.

¿Cómo podría solucionar los problemas identificados?

Apartado A

Para mitigar un poco estos ataques se pueden aplicar medidas que **limiten el trafico NTP** en las conexiones pero se requiere que haya definido una jerarquía de servidores a nivel interno de la organización de forma el número de equipos que realizan consultas NTP al exterior sea reducido (restringir el acceso). Una buena manera para limitar el acceso es **utilizando un firewall rechazando así todo tipo de conexión que no venga de una IP autorizada.**
También tendriamos que tener en cuenta otros aspectos como son, una correcta configuración del servidor para no correr el riesgo de que nos cojan nuestra lista de hosts. Otra solución sería la actualización de las redes y sus protocolos, estos ataques a NTP se dan en versiones antiguas y no tanto en las recientes. Y finalmente además de todo lo anterior, es imprescindible que las empresas de cualquier tamaño cuenten con una solución de ciberseguridad avanzada, activa en todos los endpoints y que sea capaz de prevenir, detectar y neutralizar los ataques en todo momento

Aparatado B

Ya utilizamos TCP para no perder paquetes.
La IPsec (abreviatura de Internet Protocol security) es un conjunto de protocolos cuya función es asegurar las comunicaciones sobre el Protocolo de Internet (IP) autenticando y/o cifrando cada paquete IP en un flujo de datos. IPsec también incluye protocolos para

el establecimiento de claves de cifrado.
E de igual forma que en el apartado A podemos **solucionarlo con un firewall rechazando así todo tipo de conexión que no venga de una IP autorizada.**

> **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 limpiar el caché de APT: la app de apt guarda información en forma de caché sobre las actualizaciones de cada paquete que se encuentra instalado dentro del sistema.

Comando para ver cuanto espacio nos ocupa:

```
du -sh /var/cache/apt/archives
```

→Ahora nos deshacemos de información inservible con el comando :

```
apt clean
```

→ Actualizar el sistema con `apt-get upgrade` lo cual hará que se optimicen recursos de espacio y ocupar menos en el equipo.

→Eliminar ficheros de kernel que no empleemos: es la más extrema de todas pero, si estáis seguros de que no empleáis ningún otro kernel dentro del sistema, para qué almacenar sus ficheros. Podremos hacerlo con el
comando: `apt autoremove --purge`

→Borrar los man/helps:

`rm -rf /usr/share/man`

→Borrar idiomas-